Discourse Data Model

for use with Invantive SQL
7.1.1  Admin_Admin_badgesProtected_system_fieldsByBadges ...................................................... 45
7.1.2  Admin_Badge_grouping_idsByBadges .............................................................................. 46
7.1.3  Admin_Badge_groupingsByBadges .................................................................................... 46
7.1.4  Admin_Badge_idsByBadges .............................................................................................. 47
7.1.5  Admin_Badge_type_idsByBadges ...................................................................................... 48
7.1.6  Admin_Badge_typesByBadges .......................................................................................... 49
7.1.7  Admin_BadgesByBadges .................................................................................................... 50
7.1.8  BadgeSettings: Discourse Badge Settings ..................................................................... 51
7.1.9  deleteUserBadgesById ..................................................................................................... 52
7.1.10 postUserBadges ................................................................................................................ 53
7.1.11 postUserBadges_Badge_types .......................................................................................... 54
7.1.12 postUserBadges_Badges ................................................................................................... 55
7.1.13 postUserBadges_Granted_bies ....................................................................................... 56
7.1.14 postUserBadges_Topics .................................................................................................... 57
7.1.15 postUserBadges_User_badges .......................................................................................... 58
7.1.16 postUserBadges_Users ..................................................................................................... 59
7.1.17 UserBadges_Badge_typesByUsername ............................................................................ 60
7.1.18 UserBadges_BadgesByUsername .................................................................................... 61
7.1.19 UserBadges_Granted_biesByUsername .......................................................................... 62
7.1.20 UserBadges_TopicsByUsername ..................................................................................... 63
7.1.21 UserBadges_User_badgesByUsername ............................................................................ 64
7.1.22 UserBadges_UsersByUsername ....................................................................................... 65
7.1.23 UserBadgesByUsername .................................................................................................. 66

8  Schema: Categories 67

8.1  Tables .................................................................................................................................. 67
8.1.1  c_Topic_listTopicsById .................................................................................................. 67
8.1.2  c_Topic_listTopicsPostersById ...................................................................................... 68
8.1.3  c_UsersById ................................................................................................................... 70
8.1.4  Categories: Discourse Categories .................................................................................. 71
8.1.5  CategoryById .................................................................................................................. 73
8.1.6  CategorySettings: Discourse Category Settings ............................................................. 74
8.1.7  putCategories_CategoryAvailable_groupsById ............................................................. 75
8.1.8  putCategories_CategoryGroup_permissionsById .......................................................... 76
8.1.9  putCategoriesById .......................................................................................................... 78

9  Schema: Emails 79

9.1  Tables .................................................................................................................................. 79
9.1.1  AdminEmailsByAction .................................................................................................... 79

10  Schema: Flags 80

10.1  Tables ................................................................................................................................ 80
10.1.1  AdminFlags_PostsByType ............................................................................................. 80
10.1.2  AdminFlags_TopicsByType ............................................................................................ 81
10.1.3  AdminFlags_UsersByType ............................................................................................. 83
10.1.4  AdminFlagsById ........................................................................................................... 84

11  Schema: Groups 85

11.1  Tables ................................................................................................................................ 85
11.1.1  Groups: Discourse Groups ............................................................................................ 85
11.1.2  Groups_ExtrasVisible_group_namesByGroupName ......................................................... 86
11.1.3  GroupsByName .............................................................................................................. 87
11.1.4  putGroupsById ............................................................................................................ 88

12  Schema: Invites 89

12.1  Tables ................................................................................................................................ 89
12.1.1  postInvites ................................................................................................................... 89
III Discourse Data Model

12.1.2 postInvitesLink ................................................................. 90
12.1.3 postByIdInvite ............................................................... 91
13 Schema: Logs 92
13.1 Tables ............................................................................ 92
13.1.1 AdminLogsByStaffActionLogs ........................................... 92
14 Schema: Native 94
14.1 Tables ............................................................................ 94
14.1.1 NATIVEPLATFORMSCALARREQUESTS: Discourse Native Platform Scalar Requests ........................ 94
15 Schema: Notifications 96
15.1 Tables ............................................................................ 96
15.1.1 Notifications ............................................................... 96
15.1.2 Notifications_Notifications ............................................. 97
15.1.3 putNotificationsByMarkRead ........................................... 98
16 Schema: PasswordReset 99
16.1 Tables ............................................................................ 99
16.1.1 postSessionForgotPassword ............................................. 99
16.1.2 putUsersPasswordResetByIdToken .................................... 100
17 Schema: Plugins 101
17.1 Tables ............................................................................ 101
17.1.1 AdminPlugins: Discourse Plugins ..................................... 101
18 Schema: PollPlugin 102
18.1 Tables ............................................................................ 102
18.1.1 putPolls_VoteVote ........................................................ 102
18.1.2 putPollsVote ............................................................... 103
19 Schema: Posts 104
19.1 Tables ............................................................................ 104
19.1.1 deletePostActionsById ................................................... 104
19.1.2 postPostActions .......................................................... 105
19.1.3 Posts: Discourse Posts .................................................. 106
19.1.4 Posts_Actions_summaryById ......................................... 108
19.1.5 Posts_LatestPosts_Actions_summary: Discourse Posts Latest Post Actions Summary ........................... 109
19.1.6 PostsById ................................................................. 111
19.1.7 putPostsByIdLocked ...................................................... 113
20 Schema: PrivateMessages 114
20.1 Tables ............................................................................ 114
20.1.1 TopicsPrivateMessages_Topic_listTopicsByUsername ....................... 114
20.1.2 TopicsPrivateMessages_Topic_listTopicsParticipantsByUsername .............. 116
20.1.3 TopicsPrivateMessages_Topic_listTopicsPostsByUsername .................... 118
20.1.4 TopicsPrivateMessagesByUsername ................................... 119
20.1.5 TopicsPrivateMessagesSent_Topic_listTopicsByUsername ................... 121
20.1.6 TopicsPrivateMessagesSent_Topic_listTopicsParticipantsByUsername .......... 122
20.1.7 TopicsPrivateMessagesSent_Topic_listTopicsPostsByUsername ................ 124
20.1.8 TopicsPrivateMessagesSent_UsersByUsername ............................. 126
20.1.9 TopicsPrivateMessagesSentByUsername .................................. 127
21 Schema: Reports 128
21.1 Tables ............................................................................ 128

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
22 Schema: Search

22.1 Tables

22.1.1 Search_CategoriesQuery ................................................................. 132
22.1.2 Search_Grouped_search_resultCategory_idsQuery .............................. 133
22.1.3 Search_Grouped_search_resultPost_idsQuery .................................. 134
22.1.4 Search_Grouped_search_resultUser_idsQuery .................................. 135
22.1.5 Search_PostsQuery ....................................................................... 136
22.1.6 Search_TopicsPostersQuery ............................................................. 137
22.1.7 Search_TopicsQuery ...................................................................... 139
22.1.8 Search_UsersQuery ....................................................................... 140
22.1.9 SearchQuery ............................................................................... 141

23 Schema: SiteSettingsBackups

23.1 Tables

23.1.1 putAdminSiteSettingsAllowRestore .................................................. 143
23.1.2 putAdminSiteSettingsAutomaticBackupsEnabled ............................... 144
23.1.3 putAdminSiteSettingsBackupFrequency ........................................ 145
23.1.4 putAdminSiteSettingsBackupTimeOfDay ....................................... 146
23.1.5 putAdminSiteSettingsBackupWithUploads ..................................... 147
23.1.6 putAdminSiteSettingsEnableS3Backups ......................................... 148
23.1.7 putAdminSiteSettingsMaximumBackups ....................................... 149
23.1.8 putAdminSiteSettingsReadOnlyModeDuringBackup ......................... 150
23.1.9 putAdminSiteSettingsS3BackupBucket .......................................... 151
23.1.10 putAdminSiteSettingsS3DisableCleanup ....................................... 152

24 Schema: SiteSettingsBasicSetup

24.1 Tables

24.1.1 putAdminSiteSettingsAllowUserLocale ......................................... 153
24.1.2 putAdminSiteSettingsCategoriesTopics ......................................... 154
24.1.3 putAdminSiteSettingsCategoryColors .......................................... 155
24.1.4 putAdminSiteSettingsCategoryStyle ............................................ 156
24.1.5 putAdminSiteSettingsDefaultLocale ............................................ 157
24.1.6 putAdminSiteSettingsDesktopCategoryPageStyle ........................... 158
24.1.7 putAdminSiteSettingsEnableBadges ............................................. 159
24.1.8 putAdminSiteSettingsEnableMobileTheme .................................... 160
24.1.9 putAdminSiteSettingsEnableWhispers ........................................ 161
24.1.10 putAdminSiteSettingsFixedCategoryPositions ............................... 162
24.1.11 putAdminSiteSettingsFixedCategoryPositionsOnCreate ................ 163
24.1.12 putAdminSiteSettingsGaDomainName ....................................... 164
24.1.13 putAdminSiteSettingsGaTrackingCode ....................................... 165
24.1.14 putAdminSiteSettingsGaUniversalDomainName ............................ 166
24.1.15 putAdminSiteSettingsGaUniversalTrackingCode ......................... 167
24.1.16 putAdminSiteSettingsGtmContainerId ....................................... 168
24.1.17 putAdminSiteSettingsLimitSuggestedToCategory ......................... 169
24.1.18 putAdminSiteSettingsPostMenu .................................................. 170
24.1.19 putAdminSiteSettingsPostMenuHiddenItems ................................ 171
24.1.20 putAdminSiteSettingsRelativeDateDuration ................................ 172
24.1.21 putAdminSiteSettingsSetLocaleFromAcceptLanguageHeader ........... 173
24.1.22 putAdminSiteSettingsShowSubcategoryList .................................. 174
24.1.23 putAdminSiteSettingsSuggestedTopics ....................................... 175
24.1.24 putAdminSiteSettingsSuggestedTopicsMaxDaysOld ...................... 176
24.1.25 putAdminSiteSettingsTopMenu .................................................. 177
24.1.26 putAdminSiteSettingsTrackExternalRightClicks ............................ 178
<table>
<thead>
<tr>
<th>Schema: SiteSettingsDeveloper</th>
<th>179</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.1 Tables</td>
<td></td>
</tr>
<tr>
<td>25.1.1 putAdminSiteSettingsActiveUserRateLimitSecs</td>
<td>179</td>
</tr>
<tr>
<td>25.1.2 putAdminSiteSettingsAnonPollingInterval</td>
<td>180</td>
</tr>
<tr>
<td>25.1.3 putAdminSiteSettingsBackgroundPollingInterval</td>
<td>181</td>
</tr>
<tr>
<td>25.1.4 putAdminSiteSettingsEnableLongPolling</td>
<td>182</td>
</tr>
<tr>
<td>25.1.5 putAdminSiteSettingsFlushTimingsSecs</td>
<td>183</td>
</tr>
<tr>
<td>25.1.6 putAdminSiteSettingsForceHostname</td>
<td>185</td>
</tr>
<tr>
<td>25.1.7 putAdminSiteSettingsLongPollingBaseUrl</td>
<td>186</td>
</tr>
<tr>
<td>25.1.8 putAdminSiteSettingsLongPollingInterval</td>
<td>187</td>
</tr>
<tr>
<td>25.1.9 putAdminSiteSettingsPollingInterval</td>
<td>188</td>
</tr>
<tr>
<td>25.1.10 putAdminSiteSettingsPort</td>
<td>189</td>
</tr>
<tr>
<td>25.1.11 putAdminSiteSettingsQueueJobs</td>
<td>190</td>
</tr>
<tr>
<td>25.1.12 putAdminSiteSettingsRebakeOldPostsCount</td>
<td>191</td>
</tr>
<tr>
<td>25.1.13 putAdminSiteSettingsTopTopicsFormulaFirstPostLikesMultiplier</td>
<td>192</td>
</tr>
<tr>
<td>25.1.14 putAdminSiteSettingsTopTopicsFormulaLeasesLikesPerPostMultiplier</td>
<td>193</td>
</tr>
<tr>
<td>25.1.15 putAdminSiteSettingsTopTopicsFormulaLogViewMultiplier</td>
<td>194</td>
</tr>
<tr>
<td>25.1.16 putAdminSiteSettingsVerboseLocalization</td>
<td>195</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schema: SiteSettingsEmail</th>
<th>196</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.1 Tables</td>
<td></td>
</tr>
<tr>
<td>26.1.1 putAdminSiteSettingsAlternativeReplyByEmailAddress</td>
<td>196</td>
</tr>
<tr>
<td>26.1.2 putAdminSiteSettingsAlwaysShowTrimmerContent</td>
<td>197</td>
</tr>
<tr>
<td>26.1.3 putAdminSiteSettingsAttachmentContentTypeBlacklist</td>
<td>198</td>
</tr>
<tr>
<td>26.1.4 putAdminSiteSettingsAttachmentFilenameBlacklist</td>
<td>199</td>
</tr>
<tr>
<td>26.1.5 putAdminSiteSettingsAutoGeneratedWhitelist</td>
<td>200</td>
</tr>
<tr>
<td>26.1.6 putAdminSiteSettingsBlockAutoGeneratedEmails</td>
<td>201</td>
</tr>
<tr>
<td>26.1.7 putAdminSiteSettingsBounceScoreThreshold</td>
<td>202</td>
</tr>
<tr>
<td>26.1.8 putAdminSiteSettingsBounceScoreThresholdDeactivate</td>
<td>203</td>
</tr>
<tr>
<td>26.1.9 putAdminSiteSettingsDeleteDigestEmailAfterDays</td>
<td>204</td>
</tr>
<tr>
<td>26.1.10 putAdminSiteSettingsDeleteEmailLogsAfterDays</td>
<td>205</td>
</tr>
<tr>
<td>26.1.11 putAdminSiteSettingsDigestMinExcerptLength</td>
<td>206</td>
</tr>
<tr>
<td>26.1.12 putAdminSiteSettingsDigestOtherTopics</td>
<td>207</td>
</tr>
<tr>
<td>26.1.13 putAdminSiteSettingsDigestPosts</td>
<td>208</td>
</tr>
<tr>
<td>26.1.14 putAdminSiteSettingsDigestSuppressCategories</td>
<td>209</td>
</tr>
<tr>
<td>26.1.15 putAdminSiteSettingsDigestTopics</td>
<td>210</td>
</tr>
<tr>
<td>26.1.16 putAdminSiteSettingsDisableDigestEmails</td>
<td>211</td>
</tr>
<tr>
<td>26.1.17 putAdminSiteSettingsDisableEmails</td>
<td>212</td>
</tr>
<tr>
<td>26.1.18 putAdminSiteSettingsDisplayNameOnEmailFrom</td>
<td>213</td>
</tr>
<tr>
<td>26.1.19 putAdminSiteSettingsEmailAccentBgColor</td>
<td>214</td>
</tr>
<tr>
<td>26.1.20 putAdminSiteSettingsEmailAccentFgColor</td>
<td>215</td>
</tr>
<tr>
<td>26.1.21 putAdminSiteSettingsEmailCustomHeaders</td>
<td>216</td>
</tr>
<tr>
<td>26.1.22 putAdminSiteSettingsEmailIn</td>
<td>217</td>
</tr>
<tr>
<td>26.1.23 putAdminSiteSettingsEmailInMinTrust</td>
<td>218</td>
</tr>
<tr>
<td>26.1.24 putAdminSiteSettingsEmailLinkColor</td>
<td>219</td>
</tr>
<tr>
<td>26.1.25 putAdminSiteSettingsEmailPrefix</td>
<td>220</td>
</tr>
<tr>
<td>26.1.26 putAdminSiteSettingsEmailSiteTitle</td>
<td>221</td>
</tr>
<tr>
<td>26.1.27 putAdminSiteSettingsEmailSubject</td>
<td>222</td>
</tr>
<tr>
<td>26.1.28 putAdminSiteSettingsEmailTimeWindowMins</td>
<td>223</td>
</tr>
<tr>
<td>26.1.29 putAdminSiteSettingsEnableForwardedEmails</td>
<td>224</td>
</tr>
<tr>
<td>26.1.30 putAdminSiteSettingsEnableStagedUsers</td>
<td>225</td>
</tr>
<tr>
<td>26.1.31 putAdminSiteSettingsHardBounceScore</td>
<td>226</td>
</tr>
<tr>
<td>26.1.32 putAdminSiteSettingsIgnoreByTitle</td>
<td>227</td>
</tr>
<tr>
<td>26.1.33 putAdminSiteSettingsIncomingEmailPreference</td>
<td>228</td>
</tr>
<tr>
<td>26.1.34 putAdminSiteSettingsLogMailProcessingFailures</td>
<td>229</td>
</tr>
<tr>
<td>26.1.35 putAdminSiteSettingsMailgunApiKey</td>
<td>230</td>
</tr>
<tr>
<td>26.1.36 putAdminSiteSettingsManualPollingEnabled</td>
<td>231</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>26.1.37 putAdminSiteSettingsMaxEmailsPerDayPerUser</td>
<td>232</td>
</tr>
<tr>
<td>26.1.38 putAdminSiteSettingsMaxStagedUsersPerEmail</td>
<td>233</td>
</tr>
<tr>
<td>26.1.39 putAdminSiteSettingsPop3PollingEnabled</td>
<td>234</td>
</tr>
<tr>
<td>26.1.40 putAdminSiteSettingsPop3PollingHost</td>
<td>235</td>
</tr>
<tr>
<td>26.1.41 putAdminSiteSettingsPop3PollingOpensslVerify</td>
<td>236</td>
</tr>
<tr>
<td>26.1.42 putAdminSiteSettingsPop3PollingPassword</td>
<td>237</td>
</tr>
<tr>
<td>26.1.43 putAdminSiteSettingsPop3PollingPeriodMins</td>
<td>238</td>
</tr>
<tr>
<td>26.1.44 putAdminSiteSettingsPop3PollingPort</td>
<td>239</td>
</tr>
<tr>
<td>26.1.45 putAdminSiteSettingsPop3PollingSsl</td>
<td>240</td>
</tr>
<tr>
<td>26.1.46 putAdminSiteSettingsPop3PollingUsername</td>
<td>241</td>
</tr>
<tr>
<td>26.1.47 putAdminSiteSettingsPrivateEmailTimeWindowSeconds</td>
<td>242</td>
</tr>
<tr>
<td>26.1.48 putAdminSiteSettingsReplyByEmailAddress</td>
<td>243</td>
</tr>
<tr>
<td>26.1.49 putAdminSiteSettingsReplyByEmailEnabled</td>
<td>244</td>
</tr>
<tr>
<td>26.1.50 putAdminSiteSettingsResetBounceScoreAfterDays</td>
<td>245</td>
</tr>
<tr>
<td>26.1.51 putAdminSiteSettingsShortEmailLength</td>
<td>246</td>
</tr>
<tr>
<td>26.1.52 putAdminSiteSettingsShowTopicFeaturedLinkDigest</td>
<td>247</td>
</tr>
<tr>
<td>26.1.53 putAdminSiteSettingsSoftBounceScore</td>
<td>248</td>
</tr>
<tr>
<td>26.1.54 putAdminSiteSettingsStripImagesFromShortEmails</td>
<td>249</td>
</tr>
<tr>
<td>26.1.55 putAdminSiteSettingsUnsubscribeViaEmail</td>
<td>250</td>
</tr>
<tr>
<td>26.1.56 putAdminSiteSettingsUnsubscribeViaEmailFooter</td>
<td>251</td>
</tr>
</tbody>
</table>

### 27 Schema: SiteSettingsFiles

**Tables**

- putAdminSiteSettingsMaxEmailsPerDayPerUser ........................................... 232
- putAdminSiteSettingsMaxStagedUsersPerEmail ....................................... 233
- putAdminSiteSettingsPop3PollingEnabled ............................................. 234
- putAdminSiteSettingsPop3PollingHost .................................................. 235
- putAdminSiteSettingsPop3PollingOpensslVerify .................................... 236
- putAdminSiteSettingsPop3PollingPassword ........................................... 237
- putAdminSiteSettingsPop3PollingPeriodMins ......................................... 238
- putAdminSiteSettingsPop3PollingPort .................................................. 239
- putAdminSiteSettingsPop3PollingSsl .................................................... 240
- putAdminSiteSettingsPop3PollingUsername ........................................... 241
- putAdminSiteSettingsPrivateEmailTimeWindowSeconds ................................ 242
- putAdminSiteSettingsReplyByEmailAddress ........................................... 243
- putAdminSiteSettingsReplyByEmailEnabled ........................................... 244
- putAdminSiteSettingsResetBounceScoreAfterDays .................................... 245
- putAdminSiteSettingsShortEmailLength ................................................. 246
- putAdminSiteSettingsShowTopicFeaturedLinkDigest .................................. 247
- putAdminSiteSettingsSoftBounceScore .................................................. 248
- putAdminSiteSettingsStripImagesFromShortEmails ................................... 249
- putAdminSiteSettingsUnsubscribeViaEmail ........................................... 250
- putAdminSiteSettingsUnsubscribeViaEmailFooter ................................... 251

---

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
## Schema: SiteSettingsLegal

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>putAdminSiteSettingsFaqUrl</td>
<td>288</td>
</tr>
<tr>
<td>putAdminSiteSettingsPrivacyPolicyUrl</td>
<td>289</td>
</tr>
<tr>
<td>putAdminSiteSettingsTosUrl</td>
<td>290</td>
</tr>
</tbody>
</table>

## Schema: SiteSettingsLogin

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>putAdminSiteSettingsAllowNewRegistrations</td>
<td>291</td>
</tr>
<tr>
<td>putAdminSiteSettingsEmailDomainsBlacklist</td>
<td>292</td>
</tr>
<tr>
<td>putAdminSiteSettingsEmailDomainsWhitelist</td>
<td>293</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableFacebookLogins</td>
<td>294</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableGitHubLogins</td>
<td>295</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableGoogleOauth2Logins</td>
<td>296</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableInstagramLogins</td>
<td>297</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableLocalLogins</td>
<td>298</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableSignupCta</td>
<td>299</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableSso</td>
<td>300</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableSsoProvider</td>
<td>301</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableTwitterLogins</td>
<td>302</td>
</tr>
<tr>
<td>putAdminSiteSettingsEnableYahooLogins</td>
<td>303</td>
</tr>
<tr>
<td>putAdminSiteSettingsFacebookAppId</td>
<td>304</td>
</tr>
<tr>
<td>putAdminSiteSettingsFacebookAppSecret</td>
<td>305</td>
</tr>
<tr>
<td>putAdminSiteSettingsFacebookRequestExtraProfileDetails</td>
<td>306</td>
</tr>
<tr>
<td>putAdminSiteSettingsForgotPasswordStrict</td>
<td>307</td>
</tr>
<tr>
<td>putAdminSiteSettingsGithubClientId</td>
<td>308</td>
</tr>
<tr>
<td>putAdminSiteSettingsGithubClientSecret</td>
<td>309</td>
</tr>
<tr>
<td>putAdminSiteSettingsGoogleOauth2ClientId</td>
<td>310</td>
</tr>
<tr>
<td>putAdminSiteSettingsGoogleOauth2ClientSecret</td>
<td>311</td>
</tr>
<tr>
<td>putAdminSiteSettingsInstagramConsumerKey</td>
<td>312</td>
</tr>
<tr>
<td>putAdminSiteSettingsInstagramConsumerSecret</td>
<td>313</td>
</tr>
<tr>
<td>putAdminSiteSettingsInviteOnly</td>
<td>314</td>
</tr>
<tr>
<td>putAdminSiteSettingsLoginRequired</td>
<td>315</td>
</tr>
<tr>
<td>putAdminSiteSettingsLogOutStrict</td>
<td>316</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaximumSessionAge</td>
<td>317</td>
</tr>
<tr>
<td>putAdminSiteSettingsMustApproveUsers</td>
<td>318</td>
</tr>
<tr>
<td>putAdminSiteSettingsPendingUsersReminderDelay</td>
<td>319</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoAllowAllReturnPaths</td>
<td>320</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoNotApprovedUrl</td>
<td>321</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoOverridesAvatar</td>
<td>322</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoOverridesBio</td>
<td>323</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoOverridesEmail</td>
<td>324</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoOverridesName</td>
<td>325</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoOverridesUsername</td>
<td>326</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoSecret</td>
<td>327</td>
</tr>
<tr>
<td>putAdminSiteSettingsSsoUrl</td>
<td>328</td>
</tr>
<tr>
<td>putAdminSiteSettingsTwitterConsumerKey</td>
<td>329</td>
</tr>
<tr>
<td>putAdminSiteSettingsTwitterConsumerSecret</td>
<td>330</td>
</tr>
<tr>
<td>putAdminSiteSettingsVerboseSsoLogging</td>
<td>331</td>
</tr>
</tbody>
</table>

## Schema: SiteSettingsNarrativeBot

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>putAdminSiteSettingsDisableDiscourseNarrativeBotWelcomePost</td>
<td>332</td>
</tr>
<tr>
<td>putAdminSiteSettingsDiscourseNarrativeBotDisablePublicReplies</td>
<td>333</td>
</tr>
<tr>
<td>putAdminSiteSettingsDiscourseNarrativeBotEnabled</td>
<td>334</td>
</tr>
<tr>
<td>putAdminSiteSettingsDiscourseNarrativeBotIgnoredUsernames</td>
<td>335</td>
</tr>
<tr>
<td>putAdminSiteSettingsDiscourseNarrativeBotWelcomePostDelay</td>
<td>337</td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
31 Schema: SiteSettingsOnebox 339

31.1 Tables ................................................................. 339
31.1.1 putAdminSiteSettingsEnableFlashVideoOnebox ...................................................... 339
31.1.2 putAdminSiteSettingsMaxOneboxesPerPost ............................................................ 340
31.1.3 putAdminSiteSettingsOneboxDomainsBlacklist ....................................................... 341
31.1.4 putAdminSiteSettingsPostOneboxMaxLength .......................................................... 342

32 Schema: SiteSettingsOther 343

32.1 Tables ................................................................. 343
32.1.1 putAdminSiteSettingsAutomaticallyUnpinTopics ..................................................... 343
32.1.2 putAdminSiteSettingsBootstrapModeMinUsers .......................................................... 344
32.1.3 putAdminSiteSettingsColdAgeDaysHigh ................................................................. 345
32.1.4 putAdminSiteSettingsColdAgeDaysLow ................................................................. 346
32.1.5 putAdminSiteSettingsColdAgeDaysMedium ............................................................ 347
32.1.6 putAdminSiteSettingsDeleteDraftsOlderThanNDays ................................................. 348
32.1.7 putAdminSiteSettingsDisableAvatarEducationMessage ......................................... 349
32.1.8 putAdminSiteSettingsDisableEditNotifications ....................................................... 350
32.1.9 putAdminSiteSettingsDominatingTopicMinimumPercent ....................................... 351
32.1.10 putAdminSiteSettingsEducateUntilPosts ............................................................... 352
32.1.11 putAdminSiteSettingsGlobalNotice ......................................................................... 353
32.1.12 putAdminSiteSettingsHistoryHoursHigh ............................................................... 354
32.1.13 putAdminSiteSettingsHistoryHoursLow ................................................................. 355
32.1.14 putAdminSiteSettingsHistoryHoursMedium .......................................................... 356
32.1.15 putAdminSiteSettingsMaxSimilarResults .............................................................. 357
32.1.16 putAdminSiteSettingsMinimumTopicsSimilar ....................................................... 358
32.1.17 putAdminSiteSettingsNativeAppInstallBanner ....................................................... 359
32.1.18 putAdminSiteSettingsNewVersionEmails ............................................................... 360
32.1.19 putAdminSiteSettingsNotifyAboutFlagsAfter ....................................................... 361
32.1.20 putAdminSiteSettingsPermalinkNormalization ...................................................... 362
32.1.21 putAdminSiteSettingsPreviousVisitTimeoutHours ............................................... 363
32.1.22 putAdminSiteSettingsReadTimeWordCount ......................................................... 364
32.1.23 putAdminSiteSettingsSendWelcomeMessage .......................................................... 365
32.1.24 putAdminSiteSettingsSequentialRepliesThreshold .............................................. 366
32.1.25 putAdminSiteSettingsShowCreateTopicsNotice .................................................... 367
32.1.26 putAdminSiteSettingsSlugGenerationMethod ....................................................... 368
32.1.27 putAdminSiteSettingsStaffLikeWeight ................................................................. 369
32.1.28 putAdminSiteSettingsSummaryLikesRequired ...................................................... 370
32.1.29 putAdminSiteSettingsSummaryMaxResults .......................................................... 371
32.1.30 putAdminSiteSettingsSummaryPercentFilter ....................................................... 372
32.1.31 putAdminSiteSettingsSummaryPostsRequired ...................................................... 373
32.1.32 putAdminSiteSettingsSummaryScoreThreshold ..................................................... 374
32.1.33 putAdminSiteSettingsSuppressUncategorizedBadge ............................................. 375
32.1.34 putAdminSiteSettingsTopicPageTitleIncludesCategory ....................................... 376
32.1.35 putAdminSiteSettingsTopicPostLikeHeatHigh ....................................................... 377
32.1.36 putAdminSiteSettingsTopicPostLikeHeatLow ........................................................ 378
32.1.37 putAdminSiteSettingsTopicPostLikeHeatMedium .................................................. 379
32.1.38 putAdminSiteSettingsTopicViewDurationHours ..................................................... 380
32.1.39 putAdminSiteSettingsTopicViewHeatHigh ............................................................. 381
32.1.40 putAdminSiteSettingsTopicViewHeatLow ............................................................. 382
32.1.41 putAdminSiteSettingsTopicViewHeatMedium ........................................................ 383
32.1.42 putAdminSiteSettingsUserProfileViewDurationHours ......................................... 384
32.1.43 putAdminSiteSettingsVacuumDbDays ............................................................... 385
32.1.44 putAdminSiteSettingsVersionChecks ................................................................. 386

33 Schema: SiteSettingsPlugins 387

33.1 Tables ................................................................. 387
34.1 Tables

34.1.1 putAdminSiteSettingsDailyPerformanceReport ................................................................. 387
34.1.2 putAdminSiteSettingsDetailsEnabled ................................................................. 388
34.1.3 putAdminSiteSettingsPollEditWindowMins ................................................................. 389
34.1.4 putAdminSiteSettingsPollEnabled ................................................................. 390
34.1.5 putAdminSiteSettingsPollMaximumOptions ................................................................. 391

34 Schema: SiteSettingsPosting 392

34.1.6 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.7 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.8 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.9 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.10 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.11 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.12 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.13 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.14 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.15 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.16 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.17 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.18 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.19 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.20 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.21 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.22 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.23 putAdminSiteSettingsAllowDuplicateTopicTitles ................................................................. 392
34.1.24 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.25 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.26 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.27 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.28 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.29 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.30 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.31 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.32 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.33 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.34 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.35 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.36 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.37 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.38 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.39 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.40 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.41 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.42 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.43 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.44 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.45 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.46 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.47 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.48 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.49 putAdminSiteSettingsEnableEmoji ................................................................. 408
34.1.50 putAdminSiteSettingsEnableEmoji ................................................................. 408

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
36.1.18

36.1.17

36.1.16

36.1.15

36.1.14

36.1.13

36.1.12

36.1.11

36.1.10

36.1.9

36.1.8

36.1.7

36.1.6

36.1.5

36.1.4

36.1.3

36.1.2

36.1.1

35.1.20

35.1.16

35.1.13

35.1.10

35.1.9

35.1.8

35.1.7

35.1.6

35.1.5

35.1.4

35.1.3

35.1.2

35.1.1

34.1.58

34.1.55

34.1.52

34.1.51

34.1.50

34.1.49

34.1.48

34.1.47

34.1.46

34.1.45

34.1.44

34.1.43

34.1.42

34.1.41

34.1.40

34.1.39

34.1.38

34.1.37

34.1.36

34.1.35

34.1.34

34.1.33

34.1.32

34.1.31

34.1.30

34.1.29

34.1.28

34.1.27

34.1.26

34.1.25

34.1.24

34.1.23

34.1.22

34.1.21

34.1.20

34.1.19

34.1.18

34.1.17

34.1.16

34.1.15

34.1.14

34.1.13

34.1.12

34.1.11

34.1.10

34.1.9

34.1.8

34.1.7

34.1.6

34.1.5

34.1.4

34.1.3

34.1.2

34.1.1

33.1.58

33.1.55

33.1.52

33.1.51

33.1.50

33.1.49

33.1.48

33.1.47

33.1.46

33.1.45

33.1.44

33.1.43

33.1.42

33.1.41

33.1.40

33.1.39

33.1.38

33.1.37

33.1.36

33.1.35

33.1.34

33.1.33

33.1.32

33.1.31

33.1.30

33.1.29

33.1.28

33.1.27

33.1.26

33.1.25

33.1.24

33.1.23

33.1.22

33.1.21

33.1.20

33.1.19

33.1.18

33.1.17
37 Schema: SiteSettingsSearch 496

37.1 Tables ................................................................................................................. 496
37.1.1 putAdminSiteSettingsMinSearchTermLength ................................................. 496
37.1.2 putAdminSiteSettingsSearchPreferRecentPosts ............................................ 497
37.1.3 putAdminSiteSettingsSearchRecentPostsSize ................................................. 498
37.1.4 putAdminSiteSettingsSearchTokenizeChineseJapaneseKorean ....................... 499

38 Schema: SiteSettingsSecurity 500

38.1 Tables .................................................................................................................. 500
38.1.1 putAdminSiteSettingsAllowIndexInRobotsTxt ............................................... 500
38.1.2 putAdminSiteSettingsAllowModeratorsToCreateCategories ......................... 501
38.1.3 putAdminSiteSettingsCorsOrigins .................................................................... 502
38.1.4 putAdminSiteSettingsEnableEscapedFragments ............................................. 503
38.1.5 putAdminSiteSettingsEnableNoScriptSupport .............................................. 504
38.1.6 putAdminSiteSettingsForceHttps .................................................................... 505
38.1.7 putAdminSiteSettingsUseAdminIpWhitelist .................................................. 506

39 Schema: SiteSettingsSpam 507

39.1 Tables .................................................................................................................. 507
39.1.1 putAdminSiteSettingsAddRelNoFollowToUserContent ................................ 507
39.1.2 putAdminSiteSettingsAutoBlockFastTypesMaxTrustLevel .............................. 508
39.1.3 putAdminSiteSettingsAutoBlockFastTypesOnFirstPost .................................. 509
39.1.4 putAdminSiteSettingsAutoBlockFirstPostRegexp ......................................... 510
39.1.5 putAdminSiteSettingsAutoRespondToFlagActions ...................................... 511
39.1.6 putAdminSiteSettingsCoolDownMinutesAfterHidingPosts ............................ 512
39.1.7 putAdminSiteSettingsFlagSockpuppets .......................................................... 514
39.1.8 putAdminSiteSettingsFlagsRequiredToHidePost .......................................... 515
39.1.9 putAdminSiteSettingsLevenshteinDistanceSpammerEmails ......................... 516
39.1.10 putAdminSiteSettingsMaxAgeUnmatchedEmails .......................................... 517
39.1.11 putAdminSiteSettingsMaxAgeUnmatchedIps ................................................ 518
39.1.12 putAdminSiteSettingsMaxNewAccountsPerRegistrationIp ............................ 519
39.1.13 putAdminSiteSettingsMinBanEntriesForRollUp .......................................... 520
39.1.14 putAdminSiteSettingsMinFirstPostTypingTime ............................................ 521
39.1.15 putAdminSiteSettingsNewUserSpamHostThreshold .................................... 522
39.1.16 putAdminSiteSettingsNotifyModsWhenUserBlocked ................................... 523
39.1.17 putAdminSiteSettingsNumFlaggersToCloseTopic ........................................ 524
39.1.18 putAdminSiteSettingsNumFlagsToCloseTopic ............................................. 525
39.1.19 putAdminSiteSettingsNumSpamFlagsToBlockNewUser ............................... 526
39.1.20 putAdminSiteSettingsNumT3FlagsToBlockNewUser .................................... 527
39.1.21 putAdminSiteSettingsNumT3UsersToBlockNewUser .................................... 528
39.1.22 putAdminSiteSettingsNumUsersToBlockNewUser ....................................... 529
39.1.23 putAdminSiteSettingsWhiteListedSpamHostDomains .................................. 530

40 Schema: SiteSettingsTags 532

40.1 Tables .................................................................................................................. 532
40.1.1 putAdminSiteSettingsMaxTagLength ............................................................. 532
40.1.2 putAdminSiteSettingsMaxTagSearchResults ............................................... 533
40.1.3 putAdminSiteSettingsMaxTagsInFilterList .................................................... 534
40.1.4 putAdminSiteSettingsMaxTagsPerTopic ....................................................... 535
40.1.5 putAdminSiteSettingsMinTrustLevelToTagTopics ....................................... 536
40.1.6 putAdminSiteSettingsMinTrustToCreateTag ............................................... 537
40.1.7 putAdminSiteSettingsRemoveMutedTagsFromLatest ................................. 538
40.1.8 putAdminSiteSettingsShowFilterByTag ....................................................... 539
40.1.9 putAdminSiteSettingsStaffTags .................................................................... 540
40.1.10 putAdminSiteSettingsSuppressOverlappingTagsInList ............................... 541
40.1.11 putAdminSiteSettingsTaggingEnabled ....................................................... 542
41 Schema: SiteSettingsTrustLevels 545

41.1 Tables ................................................................. 545
  41.1.1 putAdminSiteSettingsDefaultInviteeTrustLevel ....................................................... 545
  41.1.2 putAdminSiteSettingsDefaultTrustLevel ................................................................. 546
  41.1.3 putAdminSiteSettingsMnTrustToAllowSelfWiki ....................................................... 547
  41.1.4 putAdminSiteSettingsMnTrustToCreateTopic ........................................................... 548
  41.1.5 putAdminSiteSettingsMnTrustToEditPost ................................................................. 549
  41.1.6 putAdminSiteSettingsMnTrustToEditWikiPost ......................................................... 550
  41.1.7 putAdminSiteSettingsMnTrustToSendMessages ......................................................... 551
  41.1.8 putAdminSiteSettingsT1RequiresReadPosts ............................................................... 552
  41.1.9 putAdminSiteSettingsT1RequiresTimeSpentMins ....................................................... 553
  41.1.10 putAdminSiteSettingsT1RequiresTopicsEntered ..................................................... 554
  41.1.11 putAdminSiteSettingsT2RequiresDaysVisited .......................................................... 555
  41.1.12 putAdminSiteSettingsT2RequiresLikesGiven ........................................................... 556
  41.1.13 putAdminSiteSettingsT2RequiresLikesReceived ...................................................... 557
  41.1.14 putAdminSiteSettingsT2RequiresReadPosts .............................................................. 558
  41.1.15 putAdminSiteSettingsT2RequiresTimeSpentMins .................................................... 559
  41.1.16 putAdminSiteSettingsT2RequiresTopicReplyCount ............................................... 560
  41.1.17 putAdminSiteSettingsT2RequiresTopicsEntered ..................................................... 561
  41.1.18 putAdminSiteSettingsT3LinksNoFollow ................................................................. 562
  41.1.19 putAdminSiteSettingsT3PromotionMinDuration ...................................................... 563
  41.1.20 putAdminSiteSettingsT3RequiresDaysVisited .......................................................... 564
  41.1.21 putAdminSiteSettingsT3RequiresLikesGiven ........................................................... 565
  41.1.22 putAdminSiteSettingsT3RequiresLikesReceived ...................................................... 566
  41.1.23 putAdminSiteSettingsT3RequiresMaxFlagged .......................................................... 567
  41.1.24 putAdminSiteSettingsT3RequiresPostsRead ............................................................ 568
  41.1.25 putAdminSiteSettingsT3RequiresPostsReadAllTime .............................................. 569
  41.1.26 putAdminSiteSettingsT3RequiresPostsReadCap ..................................................... 571
  41.1.27 putAdminSiteSettingsT3RequiresTopicsRepliedTo .................................................. 572
  41.1.28 putAdminSiteSettingsT3RequiresTopicsViewed ..................................................... 573
  41.1.29 putAdminSiteSettingsT3RequiresTopicsViewedAllTime ....................................... 574
  41.1.30 putAdminSiteSettingsT3RequiresTopicsViewedCap ............................................. 575
  41.1.31 putAdminSiteSettingsT3TimePeriod ................................................................. 576

42 Schema: SiteSettingsUserAPI 577

42.1 Tables ................................................................. 577
  42.1.1 putAdminSiteSettingsAllowedUserApiAuthRedirects ................................................ 577
  42.1.2 putAdminSiteSettingsAllowedUserApiPushUrls ..................................................... 578
  42.1.3 putAdminSiteSettingsAllowedUserApiKeys ............................................................... 579
  42.1.4 putAdminSiteSettingsAllowedUserApiKeyScopes ..................................................... 580
  42.1.5 putAdminSiteSettingsMaxUserApiReqsPerDay ......................................................... 581
  42.1.6 putAdminSiteSettingsMaxUserApiReqsPerMinute .................................................... 582
  42.1.7 putAdminSiteSettingsMinTrustLevelForUserApiKey .............................................. 583

43 Schema: SiteSettingsUserPreferences 584

43.1 Tables ................................................................. 584
  43.1.1 putAdminSiteSettingsDefaultCategoriesMuted ......................................................... 584
  43.1.2 putAdminSiteSettingsDefaultCategoriesTracking .................................................... 585
  43.1.3 putAdminSiteSettingsDefaultCategoriesWatching .................................................. 586
  43.1.4 putAdminSiteSettingsDefaultCategoriesWatchingFirstPost .................................... 587
  43.1.5 putAdminSiteSettingsDefaultEmailAlways .............................................................. 588
  43.1.6 putAdminSiteSettingsDefaultEmailDigestFrequency ............................................... 589
  43.1.7 putAdminSiteSettingsDefaultEmailDirect ............................................................... 590
  43.1.8 putAdminSiteSettingsDefaultEmailReplyTo ............................................................ 591
  43.1.9 putAdminSiteSettingsDefaultEmailMailingListMode .............................................. 592
44 Schema: SiteSettingsUsers 610

44.1 Tables .................................................................................................................................................. 610

44.1.1 putAdminSiteSettingsDefaultEmailMailingListModeFrequency ................................................. 610
44.1.2 putAdminSiteSettingsDefaultEmailPreviousReplies ................................................................. 610
44.1.3 putAdminSiteSettingsDefaultEmailPrivateMessages ............................................................ 610
44.1.4 putAdminSiteSettingsDefaultIncludeToInDigests ................................................................. 610
44.1.5 putAdminSiteSettingsDefaultOtherAutoTrackTopicAfterMsecs ........................................ 610
44.1.6 putAdminSiteSettingsDefaultOtherDisableJumpReply ...................................................... 610
44.1.7 putAdminSiteSettingsDefaultOtherEnableQuoting ................................................................ 610
44.1.8 putAdminSiteSettingsDefaultOtherExternalLinksInNewTab ............................................... 610
44.1.9 putAdminSiteSettingsDefaultOtherLikeNotificationFrequency ........................................... 610
44.1.10 putAdminSiteSettingsDefaultOtherNewTopicDurationMinutes ........................................... 610
44.1.11 putAdminSiteSettingsDefaultOtherNotificationLevelWhenReplying ....................................... 610
44.1.12 putAdminSiteSettingsDefaultOtherTopicsAutomaticUnpin .................................................... 610
44.1.13 putAdminSiteSettingsDefaultOtherEnableMailingListMode .................................................. 610
44.1.14 putAdminSiteSettingsMaxApiKeysPerUser .............................................................................. 610

45 Schema: Tags 638

45.1 Tables .................................................................................................................................................. 638

45.1.1 TagGroups: Discourse Tag Groups .......................................................................................... 638
45.1.2 TagGroups_Tag_groupParent_tag_nameById ........................................................................... 638
45.1.3 TagGroups_Tag_groupsTag_names: Discourse Tag Group Tags .............................................. 640
45.1.4 TagGroups_Tag_groupTag_namesById .................................................................................... 641
45.1.5 TagGroupsById .......................................................................................................................... 642
45.1.6 Tags: Discourse Tags .................................................................................................................. 643
45.1.7 TagsByTag .................................................................................................................................. 644

46 Schema: Topics 645

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
50 Schema: WebHooks

50.1 Tables

50.1.1 postAdminApi_Web_hookCategory_idsWebHooks ................................................................. 717
50.1.2 postAdminApi_Web_hookGroup_idsWebHooks ................................................................. 718
50.1.3 postAdminApi_Web_hookWeb_hook_event_typesWebHooks ............................................. 719
50.1.4 postAdminApiWebHooks .................................................................................................. 720

Index
1 SQL Driver for Discourse API

Invantive SQL is the fastest, easiest and most reliable way to exchange data with the Discourse API.

Use the "Search" option in the left menu to search for a specific term such as the table or column description. When you already know the term, please use the "Index" option. When you can't find the information needed, please click on the Chat button at the bottom or place your question in the user community. Other users or Invantive Support will try to help you to our best.

Discourse is an open source discussion platform for forums, chat rooms and community building.

Discourse API Clients

Invantive SQL is available on many user interfaces ("clients" in traditional server-client paradigm). All Invantive SQL statements can be exchanged with a close to 100% compatibility across all clients and operating systems (Windows, MacOS, Linux, iOS, Android).

The clients include Microsoft Excel, Microsoft Power BI, Microsoft Power Query, Microsoft Word and Microsoft Outlook. Web-based clients include Invantive Cloud, Invantive Bridge Online as OData proxy, Invantive App Online for interactive apps, Online SQL Editor for query execution and Invantive Data Access Point as extended proxy.

The Discourse Power BI connector is based on the Invantive SQL driver for Discourse, completed by a high-performance OData connector which works straight on Power BI without any add-on. The OData protocol is always version 4, independent whether the backing platform uses OData, SOAP or another protocol.

For technical users there are command-line editions of Invantive Data Hub running on iOS, Android, Windows, MacOS and Linux. Invantive Data Hub is also often used for enterprise server applications such as ETL. High-volume replication of data taken from the Discourse API into traditional databases such as SQL Server (on-premise and Azure), MySQL, PostgreSQL and Oracle is possible using Invantive Data Replicator. Invantive Data Replicator automatically creates and maintains Discourse datawarehouses, possibly in combination with data from over 70 other (cloud) platforms. Data Replicator supports data volumes up to over 1 TB and over 5,000 companies. The on-premise edition of Invantive Bridge offers an Discourse ADO.net provider.

Finally, online web apps can be build for Discourse using App Online of Invantive Cloud.

Monitor API Calls

When a query or DML-statement has been executed on Invantive SQL a developer can evaluate the actual calls made to the Discourse API using a query on sessionios@DataDictionary. As an alternative, extensive request and response logging can be enabled by setting log-native-calls-to-disk to true. In the %USERPROFILE%\Invantive\NativeLogs folder Invantive SQL will create log files per API request and response.

Specifications

The SQL driver for Discourse does not support partitioning. Define one data container in a database for each company in Discourse to enable parallel access for data from multiple companies.
An introduction into the concepts of Invantive SQL such as databases, data containers and partitioning can be found in the [Invantive SQL grammar](https://docs.invantive.com/sql/).

The configuration can be changed using various attributes during log on and use. A full list of configuration options is listed in the [driver attributes](https://docs.invantive.com/sql/attributes).

The catalog name is used to compose the full qualified name of an object like a table or view. The schema name is used to compose the full qualified name of an object like a table or view. On Discourse the comparison of two texts is case sensitive by default.

Changes and bug fixes on the Discourse SQL driver can be found in the [release notes](https://docs.invantive.com/software/). There is currently no specific section on the Invantive forums for Discourse. Please reach out to other users of Discourse by leaving a question or contact request.

Driver code for use in settings.xml:

```xml
<discourse>
  
  <alias>
    dce
  </alias>

  <recommendedalias>
    dce
  </recommendedalias>

  <connectionstring>
    connectionString
  </connectionstring>

  <setsql>
    set NAME VALUE
  </setsql>

  <logon>
    logon
  </logon>

</discourse>
```

More technical documentation as provided by the supplier of the Discourse API on the native API connection used can be found at [https://docs.discourse.org/](https://docs.discourse.org/).

General documentation on Discourse is available at [https://www.discourse.org/](https://www.discourse.org/).

Updated: 12-12-2020 18:47 using Invantive SQL version 20.1.301-BETA+3023.

# 2 SQL Driver Attributes for Discourse API

The SQL driver for Discourse has many attributes that can be finetuned to improve handling in scenarios with unreliable network connections to the API server of Discourse or high-volumes of data. Also, many drivers have driver-specific attributes to finetune actual behaviour or handle data not matching specifications.

The Discourse driver attributes are assigned a default value which seldom requires change. However, changes can be applied when needed on four levels, which are reflected in the table below by separate checkmarks:

- Connection string: the connection string from the settings*.xml file and applied during log on.
- Set SQL statement: a set SQL-statement to be executed once connection has been established.
- Drivers file: the providers.xml file (obsolete starting release 17.32).
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for Discourse can be found in the settings*.xml file used for the database. Settings*.xml files are typically located in the `%USERPROFILE%\invantive` folder in most deployment scenarios. The reference manuals contain instructions how to relocate the settings*.xml files. Each data container of a database in the connection string can have a `connectionString` element specifying the name and values of attributes. Both name and value must be properly escaped according to XML-semantics. Actual application of the value is solely done during log on. A new connection must be established to change the value of a driver attribute using a connection string.

The set SQL statement can be executed after log on. The syntax is: `set NAME VALUE`, or for a distributed database: `set NAME@ALIAS VALUE`. In some scenarios you may need to enclose the driver attribute name in square brackets to escape it from parsing, for instance...
when a reserved SQL keyword is part of the name. The new value takes effect straight after execution of the set-statement. The set-statement can be executed as often as needed during a session.

Driver attributes that can be interactively set to a value are typically presented in the log on window. Depending on the platform and design decisions of the user interface designer, some or all of the available driver attributes can have been made available.

The Discourse driver can be configured using the following attributes:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Origin</th>
<th>Default Value</th>
<th>Set from Connection String</th>
<th>Set from SQL-Statement</th>
<th>Set from Driver's File</th>
<th>Set from Log On</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>Name of the server running Discourse, such as 'meta.discourse.org'.</td>
<td>Discourse</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>swagger-specification-url</td>
<td>The Swagger URL such as <a href="https://example.org/rest/swagger.json">https://example.org/rest/swagger.json</a>.</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-values-unknown-path</td>
<td>Whether to ignore values outside of processed paths. An error will be generated when a value occurs outside a path otherwise.</td>
<td>True</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>maximum-discovered-column-count</td>
<td>Maximum number of discovered columns. An error will be generated when the column exceeds this value.</td>
<td>250</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>api-client-id</td>
<td>The client ID is a unique identifier of your application. It is generated by registering an application.</td>
<td>OData</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>page-size-rows</td>
<td>Number of rows to retrieve per page.</td>
<td>Discourse</td>
<td>100</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>api-client-secret</td>
<td>The client secret is to be kept confidential. Such as a password for a logon code, the client secret is the confidential part of an app identified by a client ID. It is needed during the OAuth2 Code Grant Flow together with the refresh token to get access.</td>
<td>OData</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>api-refresh-token</td>
<td>Refresh Token is a security token for the OAuth2 Code Grant Flow. With a Refresh Token and client secret you can retrieve a renewed access token to access protected resources. A Refresh Token and client secret must be stored securely since once compromised allows access to your protected resources.</td>
<td>OData</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>api-redirect-url</td>
<td>The redirect URI is the website a browser session is redirected to after the OAuth2 authentication process has been completed.</td>
<td>OData</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>analysis-enforce-row-uniqueness</td>
<td>Use for analysis only! Enforce rows to be unique.</td>
<td>Shared</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from Set SQL-Statement</td>
<td>Set from Driver File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>api-scope</td>
<td>The scope to request an OAuth token for.</td>
<td>OData</td>
<td></td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>api-token-url</td>
<td>The token URI is the OAuth2 endpoint to exchange tokens.</td>
<td>OData</td>
<td></td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>api-url</td>
<td>URL to access the API.</td>
<td>OData</td>
<td></td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>bulk-delete-page-size-row s</td>
<td>Number of rows to delete per batch when bulk deleting</td>
<td>Shared</td>
<td>10000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>bulk-insert-page-size-bytes</td>
<td>Approximate maximum size in bytes of batch when bulk inserting</td>
<td>Shared</td>
<td>10000000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>bulk-insert-page-size-row s</td>
<td>Number of rows to insert per batch when bulk inserting</td>
<td>Shared</td>
<td>250</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-400-bad-request-max-tries</td>
<td>Maximum number of tries when OData server reports bad format during retrieval of data.</td>
<td>OData</td>
<td>30</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-400-bad-request-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>5000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-400-bad-request-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-400-bad-request-sleep-multiplicator</td>
<td>Multiplication factor for sleep between retries OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-422-bad-request-max-tries</td>
<td>Maximum number of tries when OData server reports unprocessable entity during retrieval of data.</td>
<td>OData</td>
<td>30</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-422-bad-request-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when OData server reports unprocessable entity during retrieval of data.</td>
<td>OData</td>
<td>5000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-422-bad-request-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when OData server reports unprocessable entity during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-422-bad-request-sleep-multiplicator</td>
<td>Multiplication factor for sleep between retries OData server reports unprocessable entity during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-429-too-many-requests-max-tries</td>
<td>Maximum number of retries when the website reports that too many requests have been made during a timeslot of one minute or one day.</td>
<td>OData</td>
<td>10</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>download-error-429-too-many-requests-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.</td>
<td>OData</td>
<td>5000</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Conne ction String</td>
<td>Set from Set SQL-Statement</td>
<td>Set from Driver s File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>dow nload-error-429-too-many-requests-sleep-maxms</td>
<td>Maximum sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-429-too-many-requests-sleep-multic peaceator</td>
<td>Multiplication factor for sleep between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-502-server-unavailable-max trials</td>
<td>Maximum number of trials when OData server reports a bad gateway during retrieval of data.</td>
<td>OData</td>
<td>30</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-502-server-unavailable-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when OData server reports a bad gateway during retrieval of data.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-502-server-unavailable-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when OData server reports a bad gateway during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-502-server-unavailable-sleep-multic paceator</td>
<td>Multiplication factor for sleep between retries OData server reports a bad gateway during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-503-server-unavailable-max trials</td>
<td>Maximum number of trials when OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>30</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-503-server-unavailable-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>5000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-503-server-unavailable-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-503-server-unavailable-sleep-multic paceator</td>
<td>Multiplication factor for sleep between retries OData server reports that the API server is unavailable during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-504-gateway-timeout-max-tr ies</td>
<td>Maximum number of trials when the website reports a gateway timeout.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-504-gateway-timeout-sleep-initial- ms</td>
<td>Initial sleep in milliseconds between retries when the website reports a gateway timeout.</td>
<td>OData</td>
<td>5000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-504-gateway-timeout-max-tr ies</td>
<td>Maximum sleep in milliseconds between retries when the website reports a gateway timeout.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from Set SQL-Statement</td>
<td>Set from Drivers File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>---------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>timeout-sleep-max-ms</td>
<td>Multiplication factor for sleep between retries when the website reports a gateway timeout.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-504-gateway-ay-timeout-sleep-multiplicator</td>
<td>Maximum number of retries when an argument exception is returned when downloading a blob.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-argument-exception-max-tries</td>
<td>Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.</td>
<td>OData</td>
<td>1000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-argument-exception-sleep-initial-ms</td>
<td>Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-argument-exception-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-io-exception-max-tries</td>
<td>Maximum number of retries when a network I/O connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-io-exception-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-io-exception-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-io-exception-sleep-multiplicator</td>
<td>Multiplication factor for sleep between retries when a network I/O connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from SQL-Statement</td>
<td>Set from Driver Files</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
<td>-----------------------------</td>
<td>------------------------</td>
<td>-----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>dow nload-error-json-exception-max tries</td>
<td>Maximum number of tries when an invalid JSON body is returned.</td>
<td>OData</td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-json-exception-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when an invalid JSON body is returned.</td>
<td>OData</td>
<td>1000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-json-exception-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when an invalid JSON body is returned.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-json-exception-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when an invalid JSON body is returned.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-other-exception-max-tries</td>
<td>Maximum number of tries when an unqualified error occurs during retrieval of data.</td>
<td>OData</td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-other-exception-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.</td>
<td>OData</td>
<td>5000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-other-exception-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.</td>
<td>OData</td>
<td>30000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-other-exception-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when an unqualified error occurs during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-socket-exception-max-tries</td>
<td>Maximum number of tries when the network connection is forcible dropped during retrieval of data.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-socket-exception-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-socket-exception-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-socket-exception-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when the network connection is forcible dropped during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-exception-max-tries</td>
<td>Maximum number of tries when a web connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-exception-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-exception-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from Set SQL-Statement</td>
<td>Set from Driver File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>---------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
<td>---------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>dow nload-error-web-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-not-implemented-max-tries</td>
<td>Maximum number of tries when the connection reports not implemented.</td>
<td>OData</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-not-implemented-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the connection reports not implemented.</td>
<td>OData</td>
<td>5000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-not-implemented-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when the connection reports not implemented.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-not-implemented-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when the connection reports not implemented.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-timeout-max-tries</td>
<td>Maximum number of tries when the connection reports a timeout.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-timeout-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the connection reports a timeout.</td>
<td>OData</td>
<td>5000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-timeout-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when the connection reports a timeout.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-timeout-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when the connection reports a timeout.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-unauthorized-max-tries</td>
<td>Maximum number of tries when the connection reports an unauthorized error.</td>
<td>OData</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-unauthorized-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the connection reports an unauthorized error.</td>
<td>OData</td>
<td>5000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-unauthorized-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when the connection reports an unauthorized error.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>dow nload-error-web-unauthorized-sleep-multiplier</td>
<td>Multiplication factor for sleep between retries when the connection reports an unauthorized error.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>force-case-sensitive-identifiers</td>
<td>Consider identifiers as case-sensitive independent of the platform capabilities.</td>
<td>Shared</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>forced-casing-identifiers</td>
<td>Forced casing of identifiers. Choose from Unset, Lower, Upper and Shared</td>
<td>Shared</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Origin</th>
<th>Default Value</th>
<th>Set from Connection String</th>
<th>Set from Set SQL-Statement</th>
<th>Set from Driver File</th>
<th>Set from Log On</th>
</tr>
</thead>
<tbody>
<tr>
<td>http-disk-cache-compression-level</td>
<td>Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.</td>
<td>Shared</td>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-disk-cache-directory</td>
<td>Directory where HTTP cache is stored.</td>
<td>Shared</td>
<td>C: \Users\gle3.WS212\Invantive\Cache\http\gle3\shared</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-disk-cache-ignore-write-errors</td>
<td>Whether to ignore write errors to disk cache.</td>
<td>Shared</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-disk-cache-max-age-sec</td>
<td>Maximum acceptable age in seconds for use of data in the HTTP disk cache.</td>
<td>Shared</td>
<td>2592000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-get-timeout-ms</td>
<td>HTTP GET timeout (ms).</td>
<td>OData</td>
<td>300000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-memory-cache-compression-level</td>
<td>Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.</td>
<td>OData</td>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-memory-cache-max-age-sec</td>
<td>Maximum acceptable age in seconds for use of data in the HTTP memory cache.</td>
<td>OData</td>
<td>14400</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>http-post-timeout-ms</td>
<td>HTTP POST timeout (ms).</td>
<td>OData</td>
<td>300000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-400-errors</td>
<td>Ignore HTTP 400 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-401-errors</td>
<td>Ignore HTTP 401 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-402-errors</td>
<td>Ignore HTTP 402 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-403-errors</td>
<td>Ignore HTTP 403 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-404-errors</td>
<td>Ignore HTTP 404 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-422-errors</td>
<td>Ignore HTTP 422 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-429-errors</td>
<td>Ignore HTTP 429 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-500-errors</td>
<td>Ignore HTTP 500 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ignore-http-502-errors</td>
<td>Ignore HTTP 502 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from SQL-Statement</td>
<td>Set from Driver's File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>---------------</td>
<td>----------------------------</td>
<td>------------------------</td>
<td>------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>invalid-json-on-get-max-tries</td>
<td>Maximum number of tries when the JSON received on GET is invalid.</td>
<td>OData</td>
<td>10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-get-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the JSON received on GET is invalid.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-get-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-get-sleep-multiplicator</td>
<td>Multiplication factor for sleep between retries when the JSON received on GET is invalid.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-post-max-tries</td>
<td>Maximum number of tries when the JSON received on POST is invalid.</td>
<td>OData</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-post-sleep-initial-ms</td>
<td>Initial sleep in milliseconds between retries when the JSON received on POST is invalid.</td>
<td>OData</td>
<td>10000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-post-sleep-max-ms</td>
<td>Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.</td>
<td>OData</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invalid-json-on-post-sleep-multiplicator</td>
<td>Multiplication factor for sleep between retries when the JSON received on POST is invalid.</td>
<td>OData</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invantive-sql-correct-invalid-date</td>
<td>Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753.</td>
<td>SQL Engine</td>
<td>V1</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invantive-sql-forward-filters-to-data-containers</td>
<td>Whether to forward filters to data containers.</td>
<td>SQL Engine</td>
<td>V1</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invantive-sql-shuffle-fetch-results-data-containers</td>
<td>Whether to shuffle results fetched from data containers.</td>
<td>SQL Engine</td>
<td>V1</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>invantive-use-cache</td>
<td>Whether to cache the results of a query.</td>
<td>SQL Engine</td>
<td>V1</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>join-set-points-per-request</td>
<td>Maximum number of values in a request when executing a join set.</td>
<td>OData</td>
<td>60</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>limit-partition-calls-left</td>
<td>Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.</td>
<td>OData</td>
<td>500</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>log-native-calls-to-disk</td>
<td>Registers native calls to data container backend as disk files.</td>
<td>Shared</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>log-native-calls-to-trace</td>
<td>Log native calls to data container backend on the trace.</td>
<td>Shared</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>maximum-length-identifiers</td>
<td>Non-default maximum length in characters of identifier names.</td>
<td>Shared</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max-odata-filters</td>
<td>The maximum number of OData filter elements.</td>
<td>OData</td>
<td>100</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from Set SQL-Statement</td>
<td>Set from Driver File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>---------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td>---------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>max-url-length-accepted</td>
<td>The maximum accepted URL length before raising an error.</td>
<td>Shared</td>
<td>8000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>max-url-length-desired</td>
<td>The maximum desired URL length.</td>
<td>Shared</td>
<td>8000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>metadata-cache-max-age-sec</td>
<td>Maximum acceptable age in seconds for re-use of metadata.</td>
<td>OData</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>partition-slot-based-rate-limit-length-ms</td>
<td>Total length in ms across all slots of a partition-based rate limit.</td>
<td>Shared</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>partition-slot-based-rate-limit-slots</td>
<td>Number of slots per partition-based rate limit. Null means no slot-based rate limit</td>
<td>Shared</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pre-request-delay-ms</td>
<td>Pre-request delay in milliseconds per request.</td>
<td>Shared</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>requested-page-size</td>
<td>Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online</td>
<td>Shared</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>requests-parallel-max</td>
<td>Maximum number of parallel data requests from individual partitions on the data container.</td>
<td>Shared</td>
<td>32</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-400-errors</td>
<td>Simulate HTTP 400 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>simulate-http-400-errors-percentage</td>
<td>Percentage of simulated HTTP 400 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-401-errors</td>
<td>Simulate HTTP 401 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>simulate-http-401-errors-percentage</td>
<td>Percentage of simulated HTTP 401 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-403-errors</td>
<td>Simulate HTTP 403 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>simulate-http-403-errors-percentage</td>
<td>Percentage of simulated HTTP 403 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-429-errors</td>
<td>Simulate HTTP 429 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>simulate-http-429-errors-percentage</td>
<td>Percentage of simulated HTTP 429 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-500-errors</td>
<td>Simulate HTTP 500 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>simulate-http-500-errors-percentage</td>
<td>Percentage of simulated HTTP 500 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Origin</td>
<td>Default Value</td>
<td>Set from Connection String</td>
<td>Set from SQL-Statement</td>
<td>Set from Driver File</td>
<td>Set from Log On</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------------</td>
<td>-----------------------------</td>
<td>------------------------</td>
<td>----------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>simulate-http-502-errors</td>
<td>Simulate HTTP 502 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-502-errors-percentage</td>
<td>Percentage of simulated HTTP 502 errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-protocol-errors</td>
<td>Simulate HTTP protocol errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-protocol-errors-percentage</td>
<td>Percentage of simulated HTTP protocol errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-timeout-errors</td>
<td>Simulate HTTP timeout errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>False</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>simulate-http-timeout-errors-percentage</td>
<td>Percentage of simulated HTTP timeout errors when exchanging results with the OData endpoint.</td>
<td>OData</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>slot-based-rate-limit-length-ms</td>
<td>Total length in ms across all slots of a slot-based rate limit.</td>
<td>Shared</td>
<td>60000</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>slot-based-rate-limit-slots</td>
<td>Number of slots of a slot-based rate limit. Null means no slot-based rate limit.</td>
<td>Shared</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>standardize-identifiers</td>
<td>Rewite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.</td>
<td>Shared</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>standardize-identifiers-casing</td>
<td>Rewite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.</td>
<td>Shared</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>swagger-specification-http-disk-cache-max-age-sec</td>
<td>Maximum acceptable age in seconds for use of Swagger specification data in the HTTP disk cache.</td>
<td>Shared</td>
<td>86400</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>use-batch-insert</td>
<td>Whether to use batch insert.</td>
<td>OData</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>use-http-disk-cache-read</td>
<td>Whether to use HTTP responses from previous queries stored on disk to answer the current query.</td>
<td>Shared</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>use-http-disk-cache-write</td>
<td>Whether to memorize HTTP responses on disk.</td>
<td>Shared</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>use-http-memory-cache-read</td>
<td>Whether to use HTTP responses from previous queries stored in memory that can answer the current query.</td>
<td>OData</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>use-http-memory-cache-write</td>
<td>Whether to memorize HTTP responses from previous queries for use by future queries.</td>
<td>OData</td>
<td>True</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
3 Schema: About
3.1 Tables
3.1.1 Administrators: Discourse Administrators

fetch about page data
Catalog: Discourse
Schema: About
Label: Administrators

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /about.json
Insert Discourse API URL: /about.json
Update Discourse API URL: /about.json
Delete Discourse API URL: /about.json
Field Selection Method: NotRequired
Base Path: about.admins[*]
Select Discourse API Operation: get /about.json

Table Columns

The columns of the table Administrators are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>https</td>
<td>boolean</td>
<td>HTTPS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>locale</td>
<td>string</td>
<td>Locale</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title_1</td>
<td>string</td>
<td>Title 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>string</td>
<td>Version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.1.2 Moderators: Discourse Moderators

fetch about page data
Catalog: Discourse
Schema: About
Label: Moderators

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /about.json
Insert Discourse API URL: /about.json
Update Discourse API URL: /about.json
Delete Discourse API URL: /about.json

Field Selection Method: NotRequired
Base Path: about.moderators[*]
Select Discourse API Operation: get /about.json

**Table Columns**

The columns of the table Moderators are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>https</td>
<td>boolean</td>
<td>HTTPS</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>locale</td>
<td>string</td>
<td>Locale</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title_1</td>
<td>string</td>
<td>Title 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>string</td>
<td>Version</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**3.1.3 SystemInformation: Discourse System Information**

fetch about page data

Catalog: Discourse
Schema: About
Label: System Information

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /about.json
Insert Discourse API URL: /about.json
Update Discourse API URL: /about.json
Delete Discourse API URL: /about.json
Field Selection Method: NotRequired
Base Path: about
Select Discourse API Operation: get /about.json

Table Columns

The columns of the table SystemInformation are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>https</td>
<td>boolean</td>
<td>HTTPS</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>locale</td>
<td>string</td>
<td>Locale</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>string</td>
<td>Version</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

4 Schema: Admin
4.1 Tables
4.1.1 Admin_Site_settingsBySiteSettings: Discourse Admin Site Settings

Show site settings

Catalog: Discourse
Schema: Admin
Label: Admin Site Settings

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings.json
Insert Discourse API URL: /admin/site_settings.json
Update Discourse API URL: /admin/site_settings.json
Delete Discourse API URL: /admin/site_settings.json
Field Selection Method: NotRequired
Base Path: site_settings[*]
Select Discourse API Operation: get /admin/site_settings.json

Table Columns

The columns of the table Admin_Site_settingsBySiteSettings are shown below. Each column has an SQL data type.
### 4.1.2 Admin_Site_settingsChoicesBySiteSettings: Discourse Admin Site Setting Choices

Show site settings
Catalog: Discourse
Schema: Admin
Label: Admin Site Setting Choices

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings.json`
Insert Discourse API URL: `/admin/site_settings.json`
Update Discourse API URL: `/admin/site_settings.json`
Delete Discourse API URL: `/admin/site_settings.json`
Field Selection Method: NotRequired
Base Path: `site_settings[*].choices[*]`
Select Discourse API Operation: get `/admin/site_settings.json`

#### Table Columns

The columns of the table `Admin_Site_settingsChoicesBySiteSettings` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>string</td>
<td>Category</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>string</td>
<td>Default</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>preview</td>
<td>string</td>
<td>Preview</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>setting</td>
<td>string</td>
<td>Setting</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>translate_names</td>
<td>boolean</td>
<td>Translate Names</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>Type</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>Value</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
### 4.1.3 Admin_Site_settingsValid_valuesBySiteSettings: Discourse Admin Site Settings Valid Values

Show site settings  
Catalog: Discourse  
Schema: Admin  
Label: Admin Site Settings Valid Values  
This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.  
Select Discourse API URL: /admin/site_settings.json  
Insert Discourse API URL: /admin/site_settings.json  
Update Discourse API URL: /admin/site_settings.json  
Delete Discourse API URL: /admin/site_settings.json  
Field Selection Method: NotRequired  
Base Path: site_settings[*].valid_values[*]  
Select Discourse API Operation: get /admin/site_settings.json  

#### Table Columns

The columns of the table `Admin_Site_settingsValid_valuesBySiteSettings` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>string</td>
<td>Category</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>string</td>
<td>Default</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>preview</td>
<td>string</td>
<td>Preview</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>setting</td>
<td>string</td>
<td>Setting</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>translate_names</td>
<td>boolean</td>
<td>Translate Names</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>Value</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>value_1</td>
<td>string</td>
<td>Value 1</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.4 AdminUsers_GroupsById

Get a specific user by id. Returns more info than the public user route.  
Catalog: Discourse  
Schema: Admin
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/users/{id}.json`
Insert Discourse API URL: `/admin/users/{id}.json`
Update Discourse API URL: `/admin/users/{id}.json`
Delete Discourse API URL: `/admin/users/{id}.json`
Field Selection Method: NotRequired
Base Path: `groups[*]`
Select Discourse API Operation: `get /admin/users/{id}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `AdminUsers_GroupsById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `AdminUsers_GroupsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>allow_membership_requests</td>
<td>boolean</td>
<td>Allow Membership Requests</td>
<td></td>
<td></td>
</tr>
<tr>
<td>automatic</td>
<td>boolean</td>
<td>Automatic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>automatic_membership_retroactive</td>
<td>boolean</td>
<td>Automatic Membership Retroactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>badge_count</td>
<td>decimal</td>
<td>#Badges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>bounce_score</td>
<td>decimal</td>
<td>Bounce Score</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_activate</td>
<td>boolean</td>
<td>Can Activate</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_be_anonymized</td>
<td>boolean</td>
<td>Can Be Anonymized</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_be_deleted</td>
<td>boolean</td>
<td>Can Be Deleted</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_deactivate</td>
<td>boolean</td>
<td>Can Deactivate</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_delete_all_posts</td>
<td>boolean</td>
<td>Can Delete All Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_disable_second_factor</td>
<td>boolean</td>
<td>Can Disable Second Factor</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_grant_admin</td>
<td>boolean</td>
<td>Can Grant Admin</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_grant_moderation</td>
<td>boolean</td>
<td>Can Grant Moderation</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_impersonate</td>
<td>boolean</td>
<td>Can Impersonate</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_revoke_admin</td>
<td>boolean</td>
<td>Can Revoke Admin</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_revoke_moderation</td>
<td>boolean</td>
<td>Can Revoke Moderation</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_send_activation_email</td>
<td>boolean</td>
<td>Can Send Activation Email</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_view_action_logs</td>
<td>boolean</td>
<td>Can View Action Logs</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at_age</td>
<td>decimal</td>
<td>Created at Age</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>days_visited</td>
<td>decimal</td>
<td>Days Visited</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>default_notification_level</td>
<td>decimal</td>
<td>Default Notification Level</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>display_name</td>
<td>string</td>
<td>Display Name</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>flag_level</td>
<td>decimal</td>
<td>Flag Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>flags_given_count</td>
<td>decimal</td>
<td>Flags Given Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>flags_received_count</td>
<td>decimal</td>
<td>Flags Received Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>full_suspend_reason</td>
<td>string</td>
<td>Full Suspend Reason</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>has_messages</td>
<td>boolean</td>
<td>Has Messages</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>ip_address</td>
<td>string</td>
<td>IPAddress</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_emailed_age</td>
<td>decimal</td>
<td>Last Emailed Age</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_emailed_at</td>
<td>string</td>
<td>Last Emailed at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_seen_age</td>
<td>decimal</td>
<td>Last Seen Age</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_seen_at</td>
<td>string</td>
<td>Last Seen at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>like_given_count</td>
<td>decimal</td>
<td>Like Given Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>mentionable_level</td>
<td>decimal</td>
<td>Mentionable Level</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>messageable_level</td>
<td>decimal</td>
<td>Messageable Level</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>name_1</td>
<td>string</td>
<td>Name 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>------------------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>post_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>posts_read_count</td>
<td>decimal</td>
<td>Posts Read Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>primary_group</td>
<td>boolean</td>
<td>Primary Group</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>private_topics_count</td>
<td>decimal</td>
<td>Private Topics Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>public_admission</td>
<td>boolean</td>
<td>Public Admission</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>public_exit</td>
<td>boolean</td>
<td>Public Exit</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>registration_ip_address</td>
<td>string</td>
<td>Registration IP Address</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>staged</td>
<td>boolean</td>
<td>Staged</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>suspended</td>
<td>boolean</td>
<td>Suspended</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>suspended_at</td>
<td>string</td>
<td>Suspended at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>suspended_by_avatar_template</td>
<td>string</td>
<td>Suspended by Avatar Template</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>suspended_by_id</td>
<td>decimal</td>
<td>Suspended by ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>suspended_by_username</td>
<td>string</td>
<td>Suspended by Username</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>suspended_till</td>
<td>string</td>
<td>Suspended till</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>time_read</td>
<td>decimal</td>
<td>Time Read</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_count</td>
<td>decimal</td>
<td>#Topics</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topics_entered</td>
<td>decimal</td>
<td>Topics Entered</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_count</td>
<td>decimal</td>
<td>#Users</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>username_lower</td>
<td>string</td>
<td>Username Lowercase</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>visibility_level</td>
<td>decimal</td>
<td>Visibility Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>warnings_received_count</td>
<td>decimal</td>
<td>Warnings Received Count</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

4.1.5 AdminUsersById

Get a specific user by id. Returns more info than the public user route.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/users/{id}.json

Insert Discourse API URL: /admin/users/{id}.json

Update Discourse API URL: /admin/users/{id}.json

Delete Discourse API URL: /admin/users/{id}.json

Field Selection Method: NotRequired
Select Discourse API Operation: get /admin/users/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function AdminUsersById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function AdminUsersById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>badge_count</td>
<td>decimal</td>
<td>#Badges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bounce_score</td>
<td>decimal</td>
<td>Bounce Score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_activate</td>
<td>boolean</td>
<td>Can Activate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_be_anonymized</td>
<td>boolean</td>
<td>Can Be Anonymized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_be_deleted</td>
<td>boolean</td>
<td>Can Be Deleted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_deactivate</td>
<td>boolean</td>
<td>Can Deactivate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete_all_posts</td>
<td>boolean</td>
<td>Can Delete All Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_disable_second_factor</td>
<td>boolean</td>
<td>Can Disable Second Factor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_grant_admin</td>
<td>boolean</td>
<td>Can Grant Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_grant_moderation</td>
<td>boolean</td>
<td>Can Grant Moderation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_impersonate</td>
<td>boolean</td>
<td>Can Impersonate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_revoke_admin</td>
<td>boolean</td>
<td>Can Revoke Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_revoke_moderation</td>
<td>boolean</td>
<td>Can Revoke Moderation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_send_activation_email</td>
<td>boolean</td>
<td>Can Send Activation Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
<td>----------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>can_view_action_logs</td>
<td>boolean</td>
<td>Can View Action Logs</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>created_at_age</td>
<td>decimal</td>
<td>Created at Age</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>days_visited</td>
<td>decimal</td>
<td>Days Visited</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>flag_level</td>
<td>decimal</td>
<td>Flag Level</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>flags_given_count</td>
<td>decimal</td>
<td>Flags Given Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>flags_received_count</td>
<td>decimal</td>
<td>Flags Received Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>full_suspend_reason</td>
<td>string</td>
<td>Full Suspend Reason</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>ip_address</td>
<td>string</td>
<td>IP Address</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>last_emailed_age</td>
<td>decimal</td>
<td>Last Emailed Age</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>last_emailed_at</td>
<td>string</td>
<td>Last Emailed at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>last_seen_age</td>
<td>decimal</td>
<td>Last Seen Age</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>last_seen_at</td>
<td>string</td>
<td>Last Seen at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>like_given_count</td>
<td>decimal</td>
<td>Like Given Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>posts_read_count</td>
<td>decimal</td>
<td>Posts Read Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>private_topics_count</td>
<td>decimal</td>
<td>Private Topics Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>registration_ip_address</td>
<td>string</td>
<td>Registration IP Address</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>staged</td>
<td>boolean</td>
<td>Staged</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>suspended</td>
<td>boolean</td>
<td>Suspended</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>suspended_at</td>
<td>string</td>
<td>Suspended at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>suspended_by_avatar_template</td>
<td>string</td>
<td>Suspended by Avatar Template</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>suspended_by_id</td>
<td>decimal</td>
<td>Suspended by ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>suspended_by_username</td>
<td>string</td>
<td>Suspended by Username</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>suspended_till</td>
<td>string</td>
<td>Suspended till</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>time_read</td>
<td>decimal</td>
<td>Time Read</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>topic_count</td>
<td>decimal</td>
<td>#Topics</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>topics_entered</td>
<td>decimal</td>
<td>Topics Entered</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>Username</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>username_lower</td>
<td>string</td>
<td>Username Lowercase</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>warnings_received_count</td>
<td>decimal</td>
<td>Warnings Received Count</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>
4.1.6 `deleteAdminGroupsByGroupId`  
Delete a group.  
Catalog: Discourse  
Schema: Admin  
This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.  
Select Discourse API URL: `/admin/groups/{group_id}.json`  
Insert Discourse API URL: `/admin/groups/{group_id}.json`  
Update Discourse API URL: `/admin/groups/{group_id}.json`  
Delete Discourse API URL: `/admin/groups/{group_id}.json`  
Field Selection Method: NotRequired  
Base Path: `success`  
Select Discourse API Operation: `delete /admin/groups/{group_id}.json`  

**Table Columns**  
The columns of the table `deleteAdminGroupsByGroupId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

4.1.7 `deleteAdminUsersByIdGroupsGroupId`  
Remove a group assigned to a user.  
Catalog: Discourse  
Schema: Admin  
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.  
Select Discourse API URL: `/admin/users/{id}/groups/{group_id}`  
Insert Discourse API URL: `/admin/users/{id}/groups/{group_id}`  
Update Discourse API URL: `/admin/users/{id}/groups/{group_id}`  
Delete Discourse API URL: `/admin/users/{id}/groups/{group_id}`  
Field Selection Method: NotRequired  
Select Discourse API Operation: `delete /admin/users/{id}/groups/{group_id}`  

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `deleteAdminUsersByIdGroupsGroupId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `deleteAdminUsersByIdGroupsGroupId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 4.1.8 `deleteGroupsByGroupIdMembers`

Remove user(s) from a group

You may use one of the following request body params to specify the user(s) to remove.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/groups/{group_id}/members.json`

Insert Discourse API URL: `/groups/{group_id}/members.json`

Update Discourse API URL: `/groups/{group_id}/members.json`

Delete Discourse API URL: `/groups/{group_id}/members.json`

Field Selection Method: NotRequired

Base Path: success

Select Discourse API Operation: delete /groups/{group_id}/members.json
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `deleteGroupsByGroupIdMembers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `deleteGroupsByGroupIdMembers` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

4.1.9 GroupsByGroupName_MembersMembers

Get some group members.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/groups/{group_name}/members.json`

Insert Discourse API URL: `/groups/{group_name}/members.json`

Update Discourse API URL: `/groups/{group_name}/members.json`

Delete Discourse API URL: `/groups/{group_name}/members.json`

Field Selection Method: NotRequired

Base Path: members[*]

Select Discourse API Operation: get `/groups/{group_name}/members.json`
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `GroupsByGroupName_Members`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_name</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `GroupsByGroupName_Members` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_limit</td>
<td>decimal</td>
<td>Meta Limit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_offset</td>
<td>decimal</td>
<td>Meta Offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_total</td>
<td>decimal</td>
<td>Meta Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.10 GroupsByGroupName_OwnersMembers

Get some group members.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/groups/{group_name}/members.json`

Insert Discourse API URL: `/groups/{group_name}/members.json`

Update Discourse API URL: `/groups/{group_name}/members.json`

Delete Discourse API URL: `/groups/{group_name}/members.json`
Field Selection Method: NotRequired

Base Path: owners[*]

Select Discourse API Operation: get /groups/{group_name}/members.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function GroupsByGroupName_OwnersMembers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_name</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function GroupsByGroupName_OwnersMembers are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta_limit</td>
<td>decimal</td>
<td>Meta Limit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_offset</td>
<td>decimal</td>
<td>Meta Offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_total</td>
<td>decimal</td>
<td>Meta Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.11 GroupsByGroupNameMembers

Get some group members.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /groups/{group_name}/members.json

Insert Discourse API URL: /groups/{group_name}/members.json

Update Discourse API URL: /groups/{group_name}/members.json
Delete Discourse API URL: /groups/{group_name}/members.json

Field Selection Method: NotRequired

Select Discourse API Operation: get /groups/{group_name}/members.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function GroupsByGroupNameMembers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with ```select * from table(value1, value2, value3)``` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ```select * from table(name1 => value1, name3 => value3)``` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_name</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function GroupsByGroupNameMembers are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta_limit</td>
<td>decimal</td>
<td>Meta Limit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_offset</td>
<td>decimal</td>
<td>Meta Offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta_total</td>
<td>decimal</td>
<td>Meta Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.12 postAdminApiKeys

Generate an api_key for a user.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/api/keys

Insert Discourse API URL: /admin/api/keys

Update Discourse API URL: /admin/api/keys
Delete Discourse API URL: `/admin/api/keys`
Field Selection Method: NotRequired
Base Path: `api_key`
Select Discourse API Operation: `post /admin/api/keys`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postAdminApiKeys`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `postAdminApiKeys` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>string</td>
<td>Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.13 `postAdminGroups`

Create a group.
Catalog: Discourse
Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/groups`
Insert Discourse API URL: /admin/groups
Update Discourse API URL: /admin/groups
Delete Discourse API URL: /admin/groups
Field Selection Method: NotRequired
Base Path: basic_group
Select Discourse API Operation: post /admin/groups

Table Function Parameters

The following parameters can be used to control the behaviour of the table function postAdminGroups. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function postAdminGroups are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias_level</td>
<td>decimal</td>
<td>Alias Level</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>automatic</td>
<td>boolean</td>
<td>Automatic</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>automatic_membership_retroactive</td>
<td>boolean</td>
<td>Automatic Membership Retroactive</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>has_messages</td>
<td>boolean</td>
<td>Has Messages</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>is_member</td>
<td>boolean</td>
<td>Is Member</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>mentionable</td>
<td>boolean</td>
<td>Mentionable</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>primary_group</td>
<td>boolean</td>
<td>Primary Group</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>user_count</td>
<td>decimal</td>
<td>#Users</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>☑</td>
<td></td>
</tr>
</tbody>
</table>
4.1.14 postAdminUsersByIdGroups

Assign a group to a user.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/users/{id}/groups

Insert Discourse API URL: /admin/users/{id}/groups

Update Discourse API URL: /admin/users/{id}/groups

Delete Discourse API URL: /admin/users/{id}/groups

Field Selection Method: NotRequired

Select Discourse API Operation: post /admin/users/{id}/groups

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function postAdminUsersByIdGroups. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function postAdminUsersByIdGroups are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
4.1.15 putAdminUsersByIdActivate
Activate a user
Catalog: Discourse
Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/users/{id}/activate
Insert Discourse API URL: /admin/users/{id}/activate
Update Discourse API URL: /admin/users/{id}/activate
Delete Discourse API URL: /admin/users/{id}/activate
Field Selection Method: NotRequired
Base Path: success
Select Discourse API Operation: put /admin/users/{id}/activate

Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminUsersByIdActivate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminUsersByIdActivate are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
4.1.16 putAdminUsersByIdAnonymize

Anonymize a user's account by changing their username and email address to random values. This is non-reversible.

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/users/{id}/anonymize

Insert Discourse API URL: /admin/users/{id}/anonymize

Update Discourse API URL: /admin/users/{id}/anonymize

Delete Discourse API URL: /admin/users/{id}/anonymize

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/users/{id}/anonymize

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminUsersByIdAnonymize. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminUsersByIdAnonymize are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>success</td>
<td>string</td>
<td>Success</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.1.17 putAdminUsersByIdSilence

Silence a user

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/users/{id}/silence

Insert Discourse API URL: /admin/users/{id}/silence

Update Discourse API URL: /admin/users/{id}/silence

Delete Discourse API URL: /admin/users/{id}/silence

Field Selection Method: NotRequired

Base Path: silence

Select Discourse API Operation: put /admin/users/{id}/silence

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminUsersByIdSilence. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminUsersByIdSilence are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>silenced</td>
<td>boolean</td>
<td>Silenced</td>
<td></td>
<td></td>
</tr>
<tr>
<td>silenced_at</td>
<td>string</td>
<td>Silenced at</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Name | Data Type | Label | Required | Documentation
--- | --- | --- | --- | ---
silenced_by_avatar_template | string | Silenced by Avatar Template | ✔️ |  
silenced_by_id | int64 | Silenced by ID | ✔️ |  
silenced_by_username | string | Silenced by User Name | ✔️ |  
silenced_reason | string | Silenced Reason | ✔️ |  
silenced_till | string | Silenced till | ✔️ |  

**4.1.18 putAdminUsersByIdSuspend**

Suspend a user

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/admin/users/{id}/suspend`

**Insert Discourse API URL:** `/admin/users/{id}/suspend`

**Update Discourse API URL:** `/admin/users/{id}/suspend`

**Delete Discourse API URL:** `/admin/users/{id}/suspend`

Field Selection Method: NotRequired

**Select Discourse API Operation:** `put /admin/users/{id}/suspend`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminUsersByIdSuspend`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminUsersByIdSuspend` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑️</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

4.1.19 `putAdminUsersByIdUnsilence`

Unsilence a user

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/users/{id}/unsilence`

Insert Discourse API URL: `/admin/users/{id}/unsilence`

Update Discourse API URL: `/admin/users/{id}/unsilence`

Delete Discourse API URL: `/admin/users/{id}/unsilence`

Field Selection Method: NotRequired

Base Path: `unsilence`

Select Discourse API Operation: `put /admin/users/{id}/unsilence`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminUsersByIdUnsilence`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminUsersByIdUnsilence` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>silenced</td>
<td>boolean</td>
<td>Silenced</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>silenced_at</td>
<td>string</td>
<td>Silenced at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>silenced_reason</td>
<td>string</td>
<td>Silenced Reason</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>silenced_till</td>
<td>string</td>
<td>Silenced till</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

4.1.20 putAdminUsersByIdUnsuspend

Unsuspend a user

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/users/{id}/unsuspend`

Insert Discourse API URL: `/admin/users/{id}/unsuspend`

Update Discourse API URL: `/admin/users/{id}/unsuspend`

Delete Discourse API URL: `/admin/users/{id}/unsuspend`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/users/{id}/unsuspend`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminUsersByIdUnsuspend`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminUsersByIdUnsuspend` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

4.1.21 `putGroupsByGroupIId_UsernameMembers`

Add user(s) to a group

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/groups/{group_id}/members.json`

Insert Discourse API URL: `/groups/{group_id}/members.json`

Update Discourse API URL: `/groups/{group_id}/members.json`

Delete Discourse API URL: `/groups/{group_id}/members.json`

Field Selection Method: NotRequired

Base Path: `usernames[*]`

Select Discourse API Operation: `put /groups/{group_id}/members.json`

Table Function API Operation: `put /groups/{group_id}/members.json`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putGroupsByGroupIId_UsernameMembers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putGroupsByGroupId_UsernamesMembers` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>success</td>
<td>string</td>
<td>Success</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

4.1.22 putGroupsByGroupIdMembers

Add user(s) to a group
Add user(s) to a group (comma separated).

Catalog: Discourse

Schema: Admin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/groups/{group_id}/members.json`

Insert Discourse API URL: `/groups/{group_id}/members.json`

Update Discourse API URL: `/groups/{group_id}/members.json`

Delete Discourse API URL: `/groups/{group_id}/members.json`

Field Selection Method: NotRequired

Select Discourse API Operation: put /groups/{group_id}/members.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putGroupsByGroupIdMembers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The columns of the table function `putGroupsByGroupIdMembers` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>success</td>
<td>string</td>
<td>Success</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.1.23 SiteSettings: Discourse Site Settings

Show site settings  
Catalog: Discourse  
Schema: Admin  
Label: Site Settings  

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings.json`  
Insert Discourse API URL: `/admin/site_settings.json`  
Update Discourse API URL: `/admin/site_settings.json`  
Delete Discourse API URL: `/admin/site_settings.json`  

Field Selection Method: NotRequired  
Select Discourse API Operation: `get /admin/site_settings.json`

**Table Columns**

The columns of the table `SiteSettings` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

### 5 Schema: AdminEmails

#### 5.1 Tables

##### 5.1.1 Admin_SettingsByEmail: Discourse Admin Settings by Email

Show email settings  
Catalog: Discourse  
Schema: AdminEmails  
Label: Admin Settings by Email  

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API URL: /admin/email.json
Insert Discourse API URL: /admin/email.json
Update Discourse API URL: /admin/email.json
Delete Discourse API URL: /admin/email.json
Field Selection Method: NotRequired
Base Path: settings[*]
Select Discourse API Operation: get /admin/email.json

Table Columns

The columns of the table Admin_SettingsByEmail are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>delivery_method</td>
<td>string</td>
<td>Delivery Method</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>string</td>
<td>Value</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

5.1.2 AdminCustomizeByEmailTemplates: Discourse Admin Customized Templates by Email

Show list of email templates
Catalog: Discourse
Schema: AdminEmails
Label: Admin Customized Templates by Email

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/customize/email_templates.json
Insert Discourse API URL: /admin/customize/email_templates.json
Update Discourse API URL: /admin/customize/email_templates.json
Delete Discourse API URL: /admin/customize/email_templates.json
Field Selection Method: NotRequired
Base Path: email_templates[*]
Select Discourse API Operation: get /admin/customize/email_templates.json

Table Columns

The columns of the table AdminCustomizeByEmailTemplates are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>Body</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
### 5.1.3 EmailSettings: Discourse Email Settings

Show email settings  
Catalog: Discourse  
Schema: AdminEmails  
Label: Email Settings  

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

**Select Discourse API URL:** /admin/email.json  
**Insert Discourse API URL:** /admin/email.json  
**Update Discourse API URL:** /admin/email.json  
**Delete Discourse API URL:** /admin/email.json  
**Field Selection Method:** NotRequired  
**Select Discourse API Operation:** get /admin/email.json  

### Table Columns

The columns of the table EmailSettings are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>delivery_method</td>
<td>string</td>
<td>Delivery Method</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

---

### 6 Schema: Backups

#### 6.1 Tables

#### 6.1.1 Backups: Discourse Backups

get a list of backups  
Catalog: Discourse  
Schema: Backups  
Label: Backups  
Retrieve: true  
Insert: true
Table Columns

The columns of the table Backups are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
<td>string</td>
<td>Filename</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>link</td>
<td>string</td>
<td>Link</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>int64</td>
<td>Size</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

6.1.2 putAdminBackupsByFilename

Download a backup. Sends an email to the account specified by your api_username with a link and token to download the backup.

Catalog: Discourse

Schema: Backups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/backups/{filename}

Insert Discourse API URL: /admin/backups/{filename}

Update Discourse API URL: /admin/backups/{filename}

Delete Discourse API URL: /admin/backups/{filename}

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/backups/{filename}

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminBackupsByFilename. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminBackupsByFilename` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

6.1.3 putAdminBackupsReadonly

update readonly mode

Catalog: Discourse

Schema: Backups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/backups/readonly

Insert Discourse API URL: /admin/backups/readonly

Update Discourse API URL: /admin/backups/readonly

Delete Discourse API URL: /admin/backups/readonly

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/backups/readonly

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminBackupsReadonly`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminBackupsReadonly` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

7 Schema: Badges

7.1 Tables

7.1.1 Admin_Admin_badgesProtected_system_fieldsByBadges

get a list of badges

Catalog: Discourse

Schema: Badges

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/badges.json

Insert Discourse API URL: /admin/badges.json

Update Discourse API URL: /admin/badges.json

Delete Discourse API URL: /admin/badges.json

Field Selection Method: NotRequired

Base Path: `admin_badges.protected_system_fields[*]`

Select Discourse API Operation: get /admin/badges.json

**Table Columns**

The columns of the table `Admin_Admin_badgesProtected_system_fieldsByBadges` are shown below. Each column has an SQL data type.
### 7.1.2 Admin_Badge_grouping_idsByBadges

get a list of badges

Catalog: Discourse

Schema: Badges

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

**Select Discourse API URL:** /admin/badges.json

**Insert Discourse API URL:** /admin/badges.json

**Update Discourse API URL:** /admin/badges.json

**Delete Discourse API URL:** /admin/badges.json

Field Selection Method: NotRequired

**Base Path:** badge_grouping_ids[*]

**Select Discourse API Operation:** get /admin/badges.json

### Table Columns

The columns of the table Admin_Badge_grouping_idsByBadges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_post_action</td>
<td>decimal</td>
<td>Triggers Post Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_post_processed</td>
<td>decimal</td>
<td>Triggers Post Processed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_post_revision</td>
<td>decimal</td>
<td>Triggers Post Revision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_trust_level_change</td>
<td>decimal</td>
<td>Triggers Trust Level Change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_user_change</td>
<td>decimal</td>
<td>Triggers User Change</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 7.1.3 Admin_Badge_groupingsByBadges

get a list of badges

Catalog: Discourse

Schema: Badges
This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/badges.json
Insert Discourse API URL: /admin/badges.json
Update Discourse API URL: /admin/badges.json
Delete Discourse API URL: /admin/badges.json
Field Selection Method: NotRequired
Base Path: badge_groupings[*]
Select Discourse API Operation: get /admin/badges.json

**Table Columns**

The columns of the table Admin_Badge_groupingsByBadges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td>decimal</td>
<td>Position</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>boolean</td>
<td>System</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_action</td>
<td>decimal</td>
<td>Triggers Post Action</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_processed</td>
<td>decimal</td>
<td>Triggers Post Processed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_revision</td>
<td>decimal</td>
<td>Triggers Post Revision</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_trust_level_change</td>
<td>decimal</td>
<td>Triggers Trust Level Change</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_user_change</td>
<td>decimal</td>
<td>Triggers User Change</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**7.1.4 Admin_Badge_idsByBadges**

get a list of badges
Catalog: Discourse
Schema: Badges

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/badges.json
Insert Discourse API URL: /admin/badges.json
Update Discourse API URL: /admin/badges.json
Delete Discourse API URL: /admin/badges.json
Field Selection Method: NotRequired
Base Path: badge_ids[*]
Select Discourse API Operation: get /admin/badges.json

Table Columns
The columns of the table Admin_Badge_idsByBadges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_action</td>
<td>decimal</td>
<td>Triggers Post Action</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_processed</td>
<td>decimal</td>
<td>Triggers Post Processed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_revision</td>
<td>decimal</td>
<td>Triggers Post Revision</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_trust_level_change</td>
<td>decimal</td>
<td>Triggers Trust Level Change</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_user_change</td>
<td>decimal</td>
<td>Triggers User Change</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

7.1.5 Admin_Badge_type_idsByBadges
get a list of badges
Catalog: Discourse
Schema: Badges
This is a read-only table. The Discourse API may not support changing the data or the Invantine SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/badges.json
Insert Discourse API URL: /admin/badges.json
Update Discourse API URL: /admin/badges.json
Delete Discourse API URL: /admin/badges.json
Field Selection Method: NotRequired
Base Path: badge_type_ids[*]
Select Discourse API Operation: get /admin/badges.json

Table Columns
The columns of the table Admin_Badge_type_idsByBadges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
### Table Columns

The columns of the table `Admin_Badge_typesByBadges` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>sort_order</td>
<td>decimal</td>
<td>Sort Order</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_action</td>
<td>decimal</td>
<td>Triggers Post Action</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_processed</td>
<td>decimal</td>
<td>Triggers Post Processed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_post_revision</td>
<td>decimal</td>
<td>Triggers Post Revision</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_trust_level_change</td>
<td>decimal</td>
<td>Triggers Trust Level Change</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_user_change</td>
<td>decimal</td>
<td>Triggers User Change</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
7.1.7 Admin_BadgesByBadges

get a list of badges

Catalog: Discourse

Schema: Badges

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/badges.json

Insert Discourse API URL: /admin/badges.json

Update Discourse API URL: /admin/badges.json

Delete Discourse API URL: /admin/badges.json

Field Selection Method: NotRequired

Base Path: badges[*]

Select Discourse API Operation: get /admin/badges.json

Table Columns

The columns of the table Admin_BadgesByBadges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_title</td>
<td>boolean</td>
<td>Allow Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>auto_revoke</td>
<td>boolean</td>
<td>Auto Revoke</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>badge_grouping_id</td>
<td>decimal</td>
<td>Badge Grouping ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>badge_type_id</td>
<td>decimal</td>
<td>Badge Type ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>enabled</td>
<td>boolean</td>
<td>Enabled</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>grant_count</td>
<td>decimal</td>
<td>Grant Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>icon</td>
<td>string</td>
<td>Icon</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>image</td>
<td>string</td>
<td>Image</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>listable</td>
<td>boolean</td>
<td>Listable</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>long_description</td>
<td>string</td>
<td>Long Description</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>multiple_grant</td>
<td>boolean</td>
<td>Multiple Grant</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>query</td>
<td>string</td>
<td>Query</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>show_posts</td>
<td>boolean</td>
<td>Show Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td>boolean</td>
<td>System</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>target_posts</td>
<td>boolean</td>
<td>Target Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>trigger</td>
<td>decimal</td>
<td>Trigger</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
### 7.1.8 BadgeSettings: Discourse Badge Settings

get a list of badges

Catalog: Discourse

Schema: Badges

Label: Badge Settings

Retrieve: true

Insert: true

Update: false

Delete: false

Select Discourse API URL: /admin/badges.json

Insert Discourse API URL: /admin/badges.json

Update Discourse API URL: /admin/badges.json

Delete Discourse API URL: /admin/badges.json

Field Selection Method: NotRequired

Select Discourse API Operation: get /admin/badges.json

Insert Discourse API Operation: post /admin/badges.json

### Table Columns

The columns of the table BadgeSettings are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>triggers_none</td>
<td>decimal</td>
<td>Triggers None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_post_action</td>
<td>decimal</td>
<td>Triggers Post Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_post_processed</td>
<td>decimal</td>
<td>Triggers Post Processed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_post_revision</td>
<td>decimal</td>
<td>Triggers Post Revision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_trust_level_change</td>
<td>decimal</td>
<td>Triggers Trust Level Change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triggers_user_change</td>
<td>decimal</td>
<td>Triggers User Change</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 7.1.9 `deleteUserBadgesById`

revoke a badge assigned to a user

Catalog: Discourse  
Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/user_badges/{id}`

Insert Discourse API URL: `/user_badges/{id}`

Update Discourse API URL: `/user_badges/{id}`

Delete Discourse API URL: `/user_badges/{id}`

Field Selection Method: NotRequired  
Base Path: success

Select Discourse API Operation: `delete /user_badges/{id}`

#### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `deleteUserBadgesById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table Function Columns

The columns of the table function `deleteUserBadgesById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.10 `postUserBadges`

assign a badge to a user

Catalog: Discourse

Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/user_badges.json`

Insert Discourse API URL: `/user_badges.json`

Update Discourse API URL: `/user_badges.json`

Delete Discourse API URL: `/user_badges.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `post /user_badges.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `postUserBadges`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `postUserBadges` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.11 postUserBadges_Badge_types
assign a badge to a user
Catalog: Discourse
Schema: Badges
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API URL: /user_badges.json
Insert Discourse API URL: /user_badges.json
Update Discourse API URL: /user_badges.json
Delete Discourse API URL: /user_badges.json
Field Selection Method: NotRequired
Base Path: badge_types[*]
Select Discourse API Operation: post /user_badges.json

Table Function Parameters
The following parameters can be used to control the behaviour of the table function
postUserBadges_Badge_types. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.
Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function postUserBadges_Badge_types are shown below.
Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.1.12 postUserBadges_Badges

assign a badge to a user

Catalog: Discourse

Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user_badges.json

Insert Discourse API URL: /user_badges.json

Update Discourse API URL: /user_badges.json

Delete Discourse API URL: /user_badges.json

Field Selection Method: NotRequired

Base Path: badges[*]

Select Discourse API Operation: post /user_badges.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function postUserBadges_Badges. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function postUserBadges_Badges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default column added since the specification specifies that no data is returned.
7.1.13 postUserBadges_Granted_bies

assign a badge to a user

Catalog: Discourse

Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user_badges.json

Insert Discourse API URL: /user_badges.json

Update Discourse API URL: /user_badges.json

Delete Discourse API URL: /user_badges.json

Field Selection Method: NotRequired

Base Path: granted_bies[*]

Select Discourse API Operation: post /user_badges.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function postUserBadges_Granted_bies. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function postUserBadges_Granted_bies are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>✔️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.14 postUserBadges_Topics

assign a badge to a user

Catalog: Discourse
Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/user_badges.json`
Insert Discourse API URL: `/user_badges.json`
Update Discourse API URL: `/user_badges.json`
Delete Discourse API URL: `/user_badges.json`

Field Selection Method: NotRequired
Base Path: `topics[*]`
Select Discourse API Operation: `post /user_badges.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `postUserBadges_Topics`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `postUserBadges_Topics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.15 postUserBadges_User_badges
assign a badge to a user

Catalog: Discourse

Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user_badges.json
Insert Discourse API URL: /user_badges.json
Update Discourse API URL: /user_badges.json
Delete Discourse API URL: /user_badges.json

Field Selection Method: NotRequired
Base Path: user_badges[*]
Select Discourse API Operation: post /user_badges.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function postUserBadges_User_badges. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function postUserBadges_User_badges are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.1.16 postUserBadges_Users
assign a badge to a user
Catalog: Discourse
Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user_badges.json
Insert Discourse API URL: /user_badges.json
Update Discourse API URL: /user_badges.json
Delete Discourse API URL: /user_badges.json
Field Selection Method: NotRequired
Base Path: users[*]
Select Discourse API Operation: post /user_badges.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
postUserBadges_Users. A value must be provided at all times for required parameters, but
optional parameters in general do not need to have a value and the execution will default to a
pre-defined behaviour. Values can be specified by position and by name. In both cases, all
parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function postUserBadges_Users are shown below. Each
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.17 UserBadges_Badge_typesByUsername

get a list of badges assigned to a user

Catalog: Discourse

Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user-badges/{username}.json

Insert Discourse API URL: /user-badges/{username}.json

Update Discourse API URL: /user-badges/{username}.json

Delete Discourse API URL: /user-badges/{username}.json

Field Selection Method: NotRequired

Base Path: badge_types[*]

Select Discourse API Operation: get /user-badges/{username}.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function UserBadges_Badge_typesByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function UserBadges_Badge_typesByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.1.18 UserBadges_BadgesByUsername
get a list of badges assigned to a user

Catalog: Discourse

Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user-badges/{username}.json
Insert Discourse API URL: /user-badges/{username}.json
Update Discourse API URL: /user-badges/{username}.json
Delete Discourse API URL: /user-badges/{username}.json

Field Selection Method: NotRequired
Base Path: badges[*]
Select Discourse API Operation: get /user-badges/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function UserBadges_BadgesByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function UserBadges_BadgesByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.19 **UserBadges_Granted_biesByUsername**

get a list of badges assigned to a user

**Catalog:** Discourse

**Schema:** Badges

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/user-badges/{username}.json`

Insert Discourse API URL: `/user-badges/{username}.json`

Update Discourse API URL: `/user-badges/{username}.json`

Delete Discourse API URL: `/user-badges/{username}.json`

**Field Selection Method:** NotRequired

**Base Path:** `granted_bies[*]`

**Select Discourse API Operation:** get `/user-badges/{username}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `UserBadges_Granted_biesByUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `UserBadges_Granted_biesByUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.20 UserBadges_TopicsByUsername
get a list of badges assigned to a user
Catalog: Discourse
Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user-badges/{username}.json
Insert Discourse API URL: /user-badges/{username}.json
Update Discourse API URL: /user-badges/{username}.json
Delete Discourse API URL: /user-badges/{username}.json
Field Selection Method: NotRequired
Base Path: topics[*]
Select Discourse API Operation: get /user-badges/{username}.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function
UserBadges_TopicsByUsername. A value must be provided at all times for required
data types, but optional parameters in general do not need to have a value and the
operation will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function UserBadges_TopicsByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
7.1.21 UserBadges_User_badgesByUsername
get a list of badges assigned to a user

Catalog: Discourse
Schema: Badges

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user-badges/{username}.json
Insert Discourse API URL: /user-badges/{username}.json
Update Discourse API URL: /user-badges/{username}.json
Delete Discourse API URL: /user-badges/{username}.json

Field Selection Method: NotRequired
Base Path: user_badges[*]
Select Discourse API Operation: get /user-badges/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
UserBadges_User_badgesByUsername. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✅</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function UserBadges_User_badgesByUsername are shown
below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
</table>
| DUMMY     | string(1) | Dummy     | ✅        | Default column added since the
 specification specifies that no
data is returned. |
7.1.22 UserBadges_UsersByUsername
get a list of badges assigned to a user
Catalog: Discourse
Schema: Badges
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /user-badges/{username}.json
Insert Discourse API URL: /user-badges/{username}.json
Update Discourse API URL: /user-badges/{username}.json
Delete Discourse API URL: /user-badges/{username}.json
Field Selection Method: NotRequired
Base Path: users[*]
Select Discourse API Operation: get /user-badges/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function UserBadges_UsersByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function UserBadges_UsersByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default column added since the specification specifies that no data is returned.
7.1.23 UserBadgesByUsername
get a list of badges assigned to a user
Catalog: Discourse
Schema: Badges
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API URL: /user-badges/{username}.json
Insert Discourse API URL: /user-badges/{username}.json
Update Discourse API URL: /user-badges/{username}.json
Delete Discourse API URL: /user-badges/{username}.json
Field Selection Method: NotRequired
Select Discourse API Operation: get /user-badges/{username}.json

Table Function Parameters
The following parameters can be used to control the behaviour of the table function
UserBadgesByUsername. A value must be provided at all times for required parameters, but
optional parameters in general do not need to have a value and the execution will default to a
pre-defined behaviour. Values can be specified by position and by name. In both cases, all
parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.
Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function UserBadgesByUsername are shown below. Each
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>✔️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
8 Schema: Categories
8.1 Tables
8.1.1 c_Topic_listTopicsById

Get a list of topics in the specified category

Catalog: Discourse
Schema: Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /c/{id}.json
Insert Discourse API URL: /c/{id}.json
Update Discourse API URL: /c/{id}.json
Delete Discourse API URL: /c/{id}.json

Field Selection Method: NotRequired
Base Path: topic_list.topics[*]
Select Discourse API Operation: get /c/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function c_Topic_listTopicsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function c_Topic_listTopicsById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td>------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft</td>
<td>boolean</td>
<td>Draft</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>int64</td>
<td>Draft Sequence</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>int64</td>
<td>Highest Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>image_url</td>
<td>string</td>
<td>Image URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>int64</td>
<td>#Likes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>int64</td>
<td>Per Page</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>int64</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>int64</td>
<td>#Replies</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>unpinned</td>
<td>boolean</td>
<td>Unpinned</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>int64</td>
<td>Views</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

### 8.1.2 c_Topic_listTopicsPostersById

Get a list of topics in the specified category

**Catalog:** Discourse  
**Schema:** Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /c/{id}.json  
**Insert Discourse API URL:** /c/{id}.json
Table Function Parameters

The following parameters can be used to control the behaviour of the table function c_Topic_listTopicsPostersById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function c_Topic_listTopicsPostersById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft</td>
<td>boolean</td>
<td>Draft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>int64</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extras</td>
<td>string</td>
<td>Extras</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table Function Parameters

8.1.3 c_UsersById

Get a list of topics in the specified category

Catalog: Discourse

Schema: Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/c/{id}.json`

Insert Discourse API URL: `/c/{id}.json`

Update Discourse API URL: `/c/{id}.json`

Delete Discourse API URL: `/c/{id}.json`

Field Selection Method: NotRequired

Base Path: `users[*]`

Select Discourse API Operation: get `/c/{id}.json`
The following parameters can be used to control the behaviour of the table function c_UsersById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td><img src="" alt=" " /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function c_UsersById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft</td>
<td>boolean</td>
<td>Topic List Draft</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_sequence</td>
<td>int64</td>
<td>Topic List Draft Sequence</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>int64</td>
<td>Topic List Per Page</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td><img src="" alt=" " /></td>
<td></td>
</tr>
</tbody>
</table>

**8.1.4 Categories: Discourse Categories**

Get a list of categories. The Categories endpoint returns a list of all the categories for your discourse site that you have access to.

Catalog: Discourse

Schema: Categories

Label: Categories

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /categories.json
Insert Discourse API URL: /categories.json
Update Discourse API URL: /categories.json
Delete Discourse API URL: /categories.json
Field Selection Method: NotRequired
Base Path: category_list.categories[*]
Select Discourse API Operation: get /categories.json

Table Columns

The columns of the table Categories are shown below. Each column has an SQL data

type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>background_url</td>
<td>string</td>
<td>Background URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_create_category</td>
<td>boolean</td>
<td>Can Create Category</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>color</td>
<td>string</td>
<td>Colour</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description_excerpt</td>
<td>string</td>
<td>Description Excerpt</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description_text</td>
<td>string</td>
<td>Description Text</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft</td>
<td>boolean</td>
<td>Draft</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>int64</td>
<td>Draft Sequence</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>has_children</td>
<td>boolean</td>
<td>Has Children</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>logo_url</td>
<td>string</td>
<td>Logo URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>string</td>
<td>Notification Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>permission</td>
<td>int64</td>
<td>Permission</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td>int64</td>
<td>Position</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>int64</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>read_restricted</td>
<td>boolean</td>
<td>Read Restricted</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>text_color</td>
<td>string</td>
<td>Text Color</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_count</td>
<td>int64</td>
<td>#Topics</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_template</td>
<td>string</td>
<td>Topic Template</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_url</td>
<td>string</td>
<td>Topic URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topics_all_time</td>
<td>int64</td>
<td>Topics All Time</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topics_day</td>
<td>int64</td>
<td>Topics Day</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topics_month</td>
<td>int64</td>
<td>Topics Month</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topics_week</td>
<td>int64</td>
<td>Topics Week</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
8.1.5 CategoryById

Get a list of topics in the specified category

Catalog: Discourse

Schema: Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/c/{id}.json`

Insert Discourse API URL: `/c/{id}.json`

Update Discourse API URL: `/c/{id}.json`

Delete Discourse API URL: `/c/{id}.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `get /c/{id}.json`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `CategoryById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `CategoryById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.1.6 CategorySettings: Discourse Category Settings

Get a list of categories

The Categories endpoint returns a list of all the categories for your discourse site that you have access to.

Catalog: Discourse

Schema: Categories

Label: Category Settings

Retrieve: true

Insert: true

Update: false

Delete: false

Select Discourse API URL: /categories.json

Insert Discourse API URL: /categories.json

Update Discourse API URL: /categories.json

Delete Discourse API URL: /categories.json

Field Selection Method: NotRequired

Base Path: category_list

Select Discourse API Operation: get /categories.json

Insert Discourse API Operation: post /categories.json

Table Columns

The columns of the table CategorySettings are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>can_create_category</td>
<td>boolean</td>
<td>Can Create Category</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft</td>
<td>boolean</td>
<td>Draft</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>int64</td>
<td>Draft Sequence</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
8.1.7 putCategories_CategoryAvailable_groupsById

Update a single category

Catalog: Discourse

Schema: Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /categories/{id}

Insert Discourse API URL: /categories/{id}

Update Discourse API URL: /categories/{id}

Delete Discourse API URL: /categories/{id}

Field Selection Method: NotRequired

Base Path: category.available_groups[*]

Select Discourse API Operation: put /categories/{id}

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putCategories_CategoryAvailable_groupsById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putCategories_CategoryAvailable_groupsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_badges</td>
<td>boolean</td>
<td>Allow Badges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Auto Close Based On Last Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>background_url</td>
<td>string</td>
<td>Background URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>------------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>color</td>
<td>string</td>
<td>Colour</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>email_in_allow_strangers</td>
<td>boolean</td>
<td>Email in Allow Strangers</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>logo_url</td>
<td>string</td>
<td>Logo URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td>decimal</td>
<td>Position</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>read_restricted</td>
<td>boolean</td>
<td>Read Restricted</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>success</td>
<td>string</td>
<td>Success</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>suppress_from_homepage</td>
<td>boolean</td>
<td>Suppress From Homepage</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>text_color</td>
<td>string</td>
<td>Text Color</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_count</td>
<td>decimal</td>
<td>#Topics</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_template</td>
<td>string</td>
<td>Topic Template</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_url</td>
<td>string</td>
<td>Topic URL</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

8.1.8 putCategories_CategoryGroup_permissionsById

Update a single category
Catalog: Discourse
Schema: Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /categories/{id}
Insert Discourse API URL: /categories/{id}
Update Discourse API URL: /categories/{id}
Delete Discourse API URL: /categories/{id}
Field Selection Method: NotRequired
Base Path: category.group_permissions[*]
Select Discourse API Operation: put /categories/{id}

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putCategories_CategoryGroup_permissionsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putCategories_CategoryGroup_permissionsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_badges</td>
<td>boolean</td>
<td>Allow Badges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Auto Close Based On Last Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>background_url</td>
<td>string</td>
<td>Background URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>color</td>
<td>string</td>
<td>Colour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>email_in_allow_strangers</td>
<td>boolean</td>
<td>Email in Allow Strangers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>group_name</td>
<td>string</td>
<td>Group Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>logo_url</td>
<td>string</td>
<td>Logo URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>permission_type</td>
<td>decimal</td>
<td>Permission Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>position</td>
<td>decimal</td>
<td>Position</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read_restricted</td>
<td>boolean</td>
<td>Read Restricted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>success</td>
<td>string</td>
<td>Success</td>
<td></td>
<td></td>
</tr>
<tr>
<td>suppress_from_homepage</td>
<td>boolean</td>
<td>Suppress From Homepage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>text_color</td>
<td>string</td>
<td>Text Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_count</td>
<td>decimal</td>
<td>#Topics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_template</td>
<td>string</td>
<td>Topic Template</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.1.9 putCategoriesById

Update a single category

Catalog: Discourse

Schema: Categories

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /categories/{id}

Insert Discourse API URL: /categories/{id}

Update Discourse API URL: /categories/{id}

Delete Discourse API URL: /categories/{id}

Field Selection Method: NotRequired

Select Discourse API Operation: put /categories/{id}

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putCategoriesById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function putCategoriesById are shown below. Each column has an SQL data type.
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category_auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Category Auto Close Based On Last Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_background_url</td>
<td>string</td>
<td>Category Background URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_can_delete</td>
<td>boolean</td>
<td>Category Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_can_edit</td>
<td>boolean</td>
<td>Category Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_color</td>
<td>string</td>
<td>Category Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_email_in_allow_strangers</td>
<td>boolean</td>
<td>Category Email in Allow Strangers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_logo_url</td>
<td>string</td>
<td>Category Logo URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_name</td>
<td>string</td>
<td>Category Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_position</td>
<td>decimal</td>
<td>Category Position</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_post_count</td>
<td>decimal</td>
<td>Category Post Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_read_restricted</td>
<td>boolean</td>
<td>Category Read Restricted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_slug</td>
<td>string</td>
<td>Category Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_suppress_from_homepage</td>
<td>boolean</td>
<td>Category Suppress From Homepage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_text_color</td>
<td>string</td>
<td>Category Text Color</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_topic_count</td>
<td>decimal</td>
<td>Category Topic Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_topic_template</td>
<td>string</td>
<td>Category Topic Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_topic_url</td>
<td>string</td>
<td>Category Topic URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>success</td>
<td>string</td>
<td>Success</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9 Schema: Emails
9.1 Tables
9.1.1 AdminEmailsByAction

get a list of emails
Catalog: Discourse
Schema: Emails

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/emails/{action}.json`
Insert Discourse API URL: `/admin/emails/{action}.json`
Update Discourse API URL: `/admin/emails/{action}.json`
Delete Discourse API URL: `/admin/emails/{action}.json`
Field Selection Method: NotRequired

Select Discourse API Operation: get /admin/emails/{action}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function AdminEmailsByAction. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>string</td>
<td></td>
<td></td>
<td>Values: sent, skipped, bounced, received, rejected.</td>
</tr>
<tr>
<td>offset</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function AdminEmailsByAction are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounced</td>
<td>boolean</td>
<td>Bounced</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>email_type</td>
<td>string</td>
<td>Email Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_key</td>
<td>string</td>
<td>Reply Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>skipped</td>
<td>boolean</td>
<td>Skipped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>to_address</td>
<td>string</td>
<td>To Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>int64</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10 Schema: Flags
10.1 Tables
10.1.1 AdminFlags_PostsByType

get a list of flags

Catalog: Discourse

Schema: Flags
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/flags/{type}.json
Insert Discourse API URL: /admin/flags/{type}.json
Update Discourse API URL: /admin/flags/{type}.json
Delete Discourse API URL: /admin/flags/{type}.json

Field Selection Method: NotRequired
Base Path: posts[*]

Select Discourse API Operation: get /admin/flags/{type}.json

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function AdminFlags_PostsByType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td>Values: active, old.</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function AdminFlags_PostsByType are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>✔️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

### 10.1.2 AdminFlags_TopicsByType

get a list of flags

Catalog: Discourse
Schema: Flags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/flags/{type}.json

Insert Discourse API URL: /admin/flags/{type}.json

Update Discourse API URL: /admin/flags/{type}.json

Delete Discourse API URL: /admin/flags/{type}.json

Field Selection Method: NotRequired

Base Path: topics[*]

Select Discourse API Operation: get /admin/flags/{type}.json

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function AdminFlags_TopicsByType. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset</td>
<td>string</td>
<td></td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td></td>
<td>☑</td>
<td>Values: active, old.</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function AdminFlags_TopicsByType are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
10.1.3 AdminFlags_UsersByType

get a list of flags

Catalog: Discourse

Schema: Flags

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/flags/{type}.json

Insert Discourse API URL: /admin/flags/{type}.json

Update Discourse API URL: /admin/flags/{type}.json

Delete Discourse API URL: /admin/flags/{type}.json

Field Selection Method: NotRequired

Base Path: users[*]

Select Discourse API Operation: get /admin/flags/{type}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
AdminFlags_UsersByType. A value must be provided at all times for required parameters,
but optional parameters in general do not need to have a value and the execution will default
to a pre-defined behaviour. Values can be specified by position and by name. In both cases,
all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>☑</td>
<td></td>
<td>Values: active, old.</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function AdminFlags_UsersByType are shown below. Each
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
10.1.4 AdminFlagsByType

get a list of flags

Catalog: Discourse

Schema: Flags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/flags/{type}.json`

Insert Discourse API URL: `/admin/flags/{type}.json`

Update Discourse API URL: `/admin/flags/{type}.json`

Delete Discourse API URL: `/admin/flags/{type}.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `get /admin/flags/{type}.json`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `AdminFlagsByType`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td></td>
<td></td>
<td>Values: active, old.</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `AdminFlagsByType` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
11 Schema: Groups
11.1 Tables
11.1.1 Groups: Discourse Groups

Get some groups.

Catalog: Discourse

Schema: Groups

Label: Groups

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /groups.json

Insert Discourse API URL: /groups.json

Update Discourse API URL: /groups.json

Delete Discourse API URL: /groups.json

Field Selection Method: NotRequired

Base Path: groups[*]

Select Discourse API Operation: get /groups.json

Table Columns

The columns of the table Groups are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_membership_requests</td>
<td>boolean</td>
<td>Allow Membership Requests</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>automatic</td>
<td>boolean</td>
<td>Automatic</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>automatic_membership_retroactive</td>
<td>boolean</td>
<td>Automatic Membership Retroactive</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>default_notification_level</td>
<td>decimal</td>
<td>Default Notification Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>display_name</td>
<td>string</td>
<td>Display Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>full_name</td>
<td>string</td>
<td>Full Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>has_messages</td>
<td>boolean</td>
<td>Has Messages</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>is_group_user</td>
<td>boolean</td>
<td>Is Group User</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>mentionable_level</td>
<td>decimal</td>
<td>Mentionable Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>messageable_level</td>
<td>decimal</td>
<td>Messageable Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>primary_group</td>
<td>boolean</td>
<td>Primary Group</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>public_admission</td>
<td>boolean</td>
<td>Public Admission</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>public_exit</td>
<td>boolean</td>
<td>Public Exit</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_count</td>
<td>decimal</td>
<td>#Users</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
11.1.2 Groups_ExtrasVisible_group_namesByName

Get a specific group.

Catalog: Discourse

Schema: Groups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /groups/{name}.json

Insert Discourse API URL: /groups/{name}.json

Update Discourse API URL: /groups/{name}.json

Delete Discourse API URL: /groups/{name}.json

Field Selection Method: NotRequired

Base Path: extras.visible_group_names[*]

Select Discourse API Operation: get /groups/{name}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function Groups_ExtrasVisible_group_namesByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function Groups_ExtrasVisible_group_namesByName are shown below. Each column has an SQL data type.
11.1.3 GroupsByName

Get a specific group.

Catalog: Discourse

Schema: Groups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /groups/{name}.json

Insert Discourse API URL: /groups/{name}.json

Update Discourse API URL: /groups/{name}.json

Delete Discourse API URL: /groups/{name}.json

Field Selection Method: NotRequired

Select Discourse API Operation: get /groups/{name}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function GroupsByName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function GroupsByName are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_allow_membership_requests</td>
<td>boolean</td>
<td>Group Allow Membership Requests</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_automatic</td>
<td>boolean</td>
<td>Group Automatic</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>group_automatic_membership_reactive</td>
<td>boolean</td>
<td>Group Automatic Membership Retroactive</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_default_notification_level</td>
<td>decimal</td>
<td>Group Default Notification Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_full_name</td>
<td>string</td>
<td>Group Full Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_has_messages</td>
<td>boolean</td>
<td>Group Has Messages</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_id</td>
<td>decimal</td>
<td>Group ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>group_is_group_owner</td>
<td>boolean</td>
<td>Group is Group Owner</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_is_group_user</td>
<td>boolean</td>
<td>Group is Group User</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_mentionable</td>
<td>boolean</td>
<td>Group Mentionable</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_mentionable_level</td>
<td>decimal</td>
<td>Group Mentionable Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_messageable</td>
<td>boolean</td>
<td>Group Messageable</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_messageable_level</td>
<td>decimal</td>
<td>Group Messageable Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_name</td>
<td>string</td>
<td>Group Name</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>group_primary_group</td>
<td>boolean</td>
<td>Group Primary Group</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>group_public_admission</td>
<td>boolean</td>
<td>Group Public Admission</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_public_exit</td>
<td>boolean</td>
<td>Group Public Exit</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>group_user_count</td>
<td>decimal</td>
<td>Group User Count</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>group_visibility_level</td>
<td>decimal</td>
<td>Group Visibility Level</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

### 11.1.4 putGroupsById

Update a group.

**Catalog:** Discourse

**Schema:** Groups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /groups/{id}.json

**Insert Discourse API URL:** /groups/{id}.json

**Update Discourse API URL:** /groups/{id}.json

**Delete Discourse API URL:** /groups/{id}.json

**Field Selection Method:** NotRequired

**Base Path:** success
Select Discourse API Operation: put /groups/{id}.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putGroupsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putGroupsById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

**12 Schema: Invites**

**12.1 Tables**

**12.1.1 postInvites**

Invite a user by email to join your forum

Catalog: Discourse

Schema: Invites

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /invites

Insert Discourse API URL: /invites

Update Discourse API URL: /invites

Delete Discourse API URL: /invites
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postInvites`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `postInvites` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>✔️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

### 12.1.2 `postInvitesLink`

Generate an invite link but don't send an email
Catalog: Discourse
Schema: Invites

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/invites/link`
Insert Discourse API URL: `/invites/link`
Update Discourse API URL: `/invites/link`
Delete Discourse API URL: `/invites/link`
Field Selection Method: NotRequired
Select Discourse API Operation: post /invites/link

Table Function Parameters

The following parameters can be used to control the behaviour of the table function postInvitesLink. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function postInvitesLink are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>true</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

12.1.3 postByldInvite

Invite user to topic
Catalog: Discourse
Schema: Invites

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}/invite
Insert Discourse API URL: /t/{id}/invite
Update Discourse API URL: /t/{id}/invite
Delete Discourse API URL: /t/{id}/invite
Field Selection Method: NotRequired
Base Path: user
Select Discourse API Operation: post /t/{id}/invite

Table Function Parameters

The following parameters can be used to control the behaviour of the table function posttByIdInvite. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function posttByIdInvite are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13 Schema: Logs
13.1 Tables
13.1.1 AdminLogsByStaffActionLogs

get staff action logs
Catalog: Discourse
Schema: Logs

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/logs/staff_action_logs.json
Insert Discourse API URL: /admin/logs/staff_action_logs.json
Update Discourse API URL: /admin/logs/staff_action_logs.json
Delete Discourse API URL: /admin/logs/staff_action_logs.json

Field Selection Method: NotRequired
Select Discourse API Operation: get /admin/logs/staff_action_logs.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function AdminLogsByStaffActionLogs. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>action_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>action_name</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function AdminLogsByStaffActionLogs are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>staff_action_logs_acting_user_avatar_template</td>
<td>string</td>
<td>Staff Action Log Acting User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_acting_user_id</td>
<td>int64</td>
<td>Staff Action Log Acting User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_acting_user_username</td>
<td>string</td>
<td>Staff Action Log Acting User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_action</td>
<td>int64</td>
<td>Staff Action Log Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_action_name</td>
<td>string</td>
<td>Staff Action Log Action Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_category_id</td>
<td>int64</td>
<td>Staff Action Log Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_context</td>
<td>string</td>
<td>Staff Action Log Context</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff_action_logs_created_at</td>
<td>string</td>
<td>Staff Action Log Created at</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14 Schema: Native
14.1 Tables
14.1.1 NATIVEPLATFORMSCALARREQUESTS: Discourse Native Platform Scalar Requests

Direct access to native API.
Catalog: Discourse
Schema: Native
Alias: npt
Label: Native Platform Scalar Requests

Documentation:
The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the Discourse API server. It will contain a new row for every row inserted with a native API request in PAYLOAD_TEXT with the results of unaltered forwarding of the payload to the Discourse API server.

Retrieve: true
Insert: true
Update: false
Delete: false

View Columns
The columns of the view **NATIVEPLATFORMSCALARREQUESTS** are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOB_PREFERRED</td>
<td>boolean</td>
<td>BLOB Preferred</td>
<td>☑️</td>
<td>Indicator whether a BLOB result is preferred over text.</td>
</tr>
<tr>
<td>BLOB_RESPONSE_CACHE_MAX_AGE_SEC</td>
<td>int32</td>
<td>Response Cache Maximum Age (sec)</td>
<td>☐️</td>
<td>Maximum age in seconds of Bridge Online response cache entries to be used.</td>
</tr>
<tr>
<td>CONTENT_TYPE</td>
<td>string(240)</td>
<td>Content Type</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>DATE_ENDED</td>
<td>datetime</td>
<td>End Date</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>DATE_STARTED</td>
<td>datetime</td>
<td>Start Date</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>DRY_RUN</td>
<td>boolean</td>
<td>Run without Actions</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>ERROR_MESSAGE_CODE</td>
<td>string(30)</td>
<td>Error Message Code</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>ERROR_MESSAGE_TEXT</td>
<td>string(4000)</td>
<td>Error Message Text</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>HTTP_DISK_CACHE_MAX_AGE_SEC</td>
<td>int32</td>
<td>HTTP Disk Cache Maximum Age (sec)</td>
<td>☐️</td>
<td>Maximum age in seconds of HTTP disk cache entries to be used.</td>
</tr>
<tr>
<td>HTTP_DISK_CACHE_SAVE</td>
<td>boolean</td>
<td>Save HTTP Disk Cache</td>
<td>☐️</td>
<td>Whether results can be stored in HTTP disk cache.</td>
</tr>
<tr>
<td>HTTP_DISK_CACHE_USE</td>
<td>boolean</td>
<td>Use HTTP Disk Cache</td>
<td>☐️</td>
<td>Whether results can be fetched from HTTP disk cache.</td>
</tr>
<tr>
<td>HTTP_MEMORY_CACHE_MAX_AGE_SEC</td>
<td>int32</td>
<td>HTTP Memory Cache Maximum Age (sec)</td>
<td>☐️</td>
<td>Maximum age in seconds of HTTP memory cache entries to be used.</td>
</tr>
<tr>
<td>HTTP_MEMORY_CACHE_SAVE</td>
<td>boolean</td>
<td>Save HTTP Memory Cache</td>
<td>☐️</td>
<td>Whether results can be stored in HTTP memory cache.</td>
</tr>
<tr>
<td>HTTP_MEMORY_CACHE_USE</td>
<td>boolean</td>
<td>Use HTTP Memory Cache</td>
<td>☐️</td>
<td>Whether results can be fetched from HTTP memory cache.</td>
</tr>
<tr>
<td>HTTP_METHOD</td>
<td>string(30)</td>
<td>HTTP Method</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>ORIG_SYSTEM_GROUP</td>
<td>string(4000)</td>
<td>Original System Group</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>ORIG_SYSTEM_REFERENCE</td>
<td>string(4000)</td>
<td>Original System Reference</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>PAYLOAD_TEXT</td>
<td>string</td>
<td>Payload</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>RESULT_BLOB</td>
<td>byte[]</td>
<td>Result BLOB</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>RESULT_TEXT</td>
<td>string</td>
<td>Result Text</td>
<td>☐️</td>
<td></td>
</tr>
<tr>
<td>SUCCESSFUL</td>
<td>boolean</td>
<td>Successful</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>TIMEOUT_SEC</td>
<td>int32</td>
<td>Timeout (sec)</td>
<td>☐️</td>
<td>Timeout in seconds.</td>
</tr>
<tr>
<td>TRANSACTION_ID</td>
<td>int32</td>
<td>Transaction ID</td>
<td>☑️</td>
<td>Incrementing ID of the transaction.</td>
</tr>
<tr>
<td>URL</td>
<td>string(4000)</td>
<td>URL</td>
<td>☐️</td>
<td></td>
</tr>
</tbody>
</table>
15 Schema: Notifications

15.1 Tables

15.1.1 Notifications

Get notifications for the specified user

Catalog: Discourse

Schema: Notifications

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /notifications.json

Insert Discourse API URL: /notifications.json

Update Discourse API URL: /notifications.json

Delete Discourse API URL: /notifications.json

Field Selection Method: NotRequired

Select Discourse API Operation: get /notifications.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function Notifications. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function Notifications are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>load_more_notifications</td>
<td>string</td>
<td>Load More Notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seen_notification_id</td>
<td>decimal</td>
<td>Seen Notification ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
15.1.2 Notifications_Notifications

Get notifications for the specified user

Catalog: Discourse
Schema: Notifications

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /notifications.json
Insert Discourse API URL: /notifications.json
Update Discourse API URL: /notifications.json
Delete Discourse API URL: /notifications.json
Field Selection Method: NotRequired
Base Path: notifications[*]
Select Discourse API Operation: get /notifications.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function Notifications_Notifications. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function Notifications_Notifications are shown below. Each column has an SQL data type.
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_display_username</td>
<td>string</td>
<td>Data Display Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_original_post_id</td>
<td>decimal</td>
<td>Data Original Post ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_original_post_type</td>
<td>decimal</td>
<td>Data Original Post Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_original_username</td>
<td>string</td>
<td>Data Original Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data_topic_title</td>
<td>string</td>
<td>Data Topic Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>load_more_notifications</td>
<td>string</td>
<td>Load More Notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_type</td>
<td>decimal</td>
<td>Notification Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>boolean</td>
<td>Read</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seen_notification_id</td>
<td>decimal</td>
<td>Seen Notification ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>total_rows_notifications</td>
<td>decimal</td>
<td>Total Row s Notifications</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15.1.3 putNotificationsByMarkRead

Mark notifications as read

Catalog: Discourse

Schema: Notifications

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /notifications/mark-read.json

Insert Discourse API URL: /notifications/mark-read.json

Update Discourse API URL: /notifications/mark-read.json

Delete Discourse API URL: /notifications/mark-read.json

Field Selection Method: NotRequired

Base Path: success

Select Discourse API Operation: put /notifications/mark-read.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putNotificationsByMarkRead. A value must be provided at all times for required parameters,
but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putNotificationsByMarkRead` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

## 16 Schema: PasswordReset
### 16.1 Tables
#### 16.1.1 postSessionForgotPassword

Send password reset email

Catalog: Discourse

Schema: PasswordReset

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/session/forgot_password`

Insert Discourse API URL: `/session/forgot_password`

Update Discourse API URL: `/session/forgot_password`

Delete Discourse API URL: `/session/forgot_password`

Field Selection Method: NotRequired

Select Discourse API Operation: `post /session/forgot_password`

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function postSessionForgotPassword. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function postSessionForgotPassword are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 16.1.2 putUsersPasswordResetByToken

Create a new password

Catalog: Discourse

Schema: PasswordReset

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /users/password-reset/{token}

Insert Discourse API URL: /users/password-reset/{token}

Update Discourse API URL: /users/password-reset/{token}

Delete Discourse API URL: /users/password-reset/{token}

Field Selection Method: NotRequired

Select Discourse API Operation: put /users/password-reset/{token}

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putUsersPasswordResetByToken. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putUsersPasswordResetByToken` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>✔️</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**17 Schema: Plugins**

**17.1 Tables**

**17.1.1 AdminPlugins: Discourse Plugins**

get a list of installed plugins

Catalog: Discourse

Schema: Plugins

Label: Plugins

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/plugins`

Insert Discourse API URL: `/admin/plugins`

Update Discourse API URL: `/admin/plugins`

Delete Discourse API URL: `/admin/plugins`

Field Selection Method: NotRequired

Base Path: `plugins[*]`

Select Discourse API Operation: get `/admin/plugins`
Table Columns

The columns of the table AdminPlugins are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>boolean</td>
<td>Enabled</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>enabled_setting</td>
<td>string</td>
<td>Enabled Setting</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>string</td>
<td>Version</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

18 Schema: PollPlugin
18.1 Tables
18.1.1 putPolls_VoteVote

vote on a poll

Catalog: Discourse
Schema: PollPlugin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /polls/vote
Insert Discourse API URL: /polls/vote
Update Discourse API URL: /polls/vote
Delete Discourse API URL: /polls/vote

Field Selection Method: NotRequired
Base Path: vote[*]
Select Discourse API Operation: put /polls/vote

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putPolls_VoteVote. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putPolls_VoteVote` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>poll_name</td>
<td>string</td>
<td>Poll Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poll_options_html</td>
<td>string</td>
<td>Poll Option HTML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poll_options_id</td>
<td>string</td>
<td>Poll Option ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poll_options_votes</td>
<td>int64</td>
<td>Poll Option Votes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poll_status</td>
<td>string</td>
<td>Poll Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poll_type</td>
<td>string</td>
<td>Poll Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poll_voters</td>
<td>int64</td>
<td>Poll Voters</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

18.1.2 putPollsVote

vote on a poll

Catalog: Discourse

Schema: PollPlugin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /polls/vote

Insert Discourse API URL: /polls/vote

Update Discourse API URL: /polls/vote

Delete Discourse API URL: /polls/vote

Field Selection Method: NotRequired

Select Discourse API Operation: put /polls/vote

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putPollsVote`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putPollsVote` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>poll_name</td>
<td>string</td>
<td>Poll Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>poll_options_html</td>
<td>string</td>
<td>Poll Option HTML</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>poll_options_id</td>
<td>string</td>
<td>Poll Option ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>poll_options_votes</td>
<td>int64</td>
<td>Poll Option Votes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>poll_status</td>
<td>string</td>
<td>Poll Status</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>poll_type</td>
<td>string</td>
<td>Poll Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>poll_voters</td>
<td>int64</td>
<td>Poll Voters</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

19 Schema: Posts

19.1 Tables

19.1.1 deletePostActionsById

Un-like a post

Catalog: Discourse

Schema: Posts

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/post_actions/{id}`

Insert Discourse API URL: `/post_actions/{id}`

Update Discourse API URL: `/post_actions/{id}`

Delete Discourse API URL: `/post_actions/{id}`

Field Selection Method: NotRequired

Select Discourse API Operation: delete /post_actions/{id}
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `deletePostActionsById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `deletePostActionsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

19.1.2 postPostActions

Like a post and other actions

Catalog: Discourse

Schema: Posts

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/post_actions`

Insert Discourse API URL: `/post_actions`

Update Discourse API URL: `/post_actions`

Delete Discourse API URL: `/post_actions`

Field Selection Method: NotRequired

Select Discourse API Operation: `post /post_actions`
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postPostActions`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `postPostActions` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

19.1.3 Posts: Discourse Posts

Get a list of the latest posts across topics

Catalog: Discourse

Schema: Posts

Label: Posts

Retrieve: true

Insert: true

Update: false

Delete: false

Select Discourse API URL: /posts.json

Insert Discourse API URL: /posts.json

Update Discourse API URL: /posts.json

Delete Discourse API URL: /posts.json

Field Selection Method: NotRequired
Base Path: `latest_posts[*]`

Select Discourse API Operation: `get /posts.json`

Insert Discourse API Operation: `post /posts.json`

### Table Columns

The columns of the table `Posts` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wiki</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>raw</td>
<td>string</td>
<td>Raw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>static_doc</td>
<td>boolean</td>
<td>Static Doc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_html_title</td>
<td>string</td>
<td>Topic HTML Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_title</td>
<td>string</td>
<td>Topic Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
19.1.4 Posts_Actions_summaryById

Get a single post
Catalog: Discourse
Schema: Posts

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /posts/{id}.json
Insert Discourse API URL: /posts/{id}.json
Update Discourse API URL: /posts/{id}.json
Delete Discourse API URL: /posts/{id}.json
Field Selection Method: NotRequired
Base Path: actions_summary[*]
Select Discourse API Operation: get /posts/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function Posts_Actions_summaryById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `Posts_Actions_summaryById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_recover</td>
<td>boolean</td>
<td>Can Recover</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wikify</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>raw</td>
<td>string</td>
<td>Raw</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>decimal</td>
<td>Version</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>wiki</td>
<td>boolean</td>
<td>Wiki</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>yours</td>
<td>boolean</td>
<td>Yours</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

19.1.5 Posts_Latest_postsActions_summary: Discourse Posts Latest Post Actions Summary

Get a list of the latest posts across topics

Catalog: Discourse
Schema: Posts

Label: Posts Latest Post Actions Summary

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /posts.json
Insert Discourse API URL: /posts.json
Update Discourse API URL: /posts.json
Delete Discourse API URL: /posts.json

Field Selection Method: NotRequired
Base Path: latest_posts[*].actions_summary[*]
Select Discourse API Operation: get /posts.json

Table Columns

The columns of the table Posts_Latest_postsActions_summary are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_act</td>
<td>boolean</td>
<td>Can Act</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wikify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hidden_1</td>
<td>boolean</td>
<td>Hidden 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>raw</td>
<td>string</td>
<td>Raw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>static_doc</td>
<td>boolean</td>
<td>Static Doc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_html_title</td>
<td>string</td>
<td>Topic HTML Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_title</td>
<td>string</td>
<td>Topic Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>decimal</td>
<td>Version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wiki</td>
<td>boolean</td>
<td>Wiki</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yours</td>
<td>boolean</td>
<td>Yours</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19.1.6 PostsById
Get a single post
Catalog: Discourse
Schema: Posts
Retrieve: true
Insert: false
Update: true
Delete: false
Select Discourse API URL: /posts/{id}.json
Insert Discourse API URL: /posts/{id}.json
Update Discourse API URL: /posts/{id}.json
Delete Discourse API URL: /posts/{id}.json
Field Selection Method: NotRequired
Select Discourse API Operation: get /posts/{id}.json
Update Discourse API Operation: put /posts/{id}.json

Table Function Parameters
The following parameters can be used to control the behaviour of the table function PostsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-
defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `PostsById` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during update.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_recover</td>
<td>boolean</td>
<td>Can Recover</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wikify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>raw</td>
<td>string</td>
<td>Raw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>decimal</td>
<td>Version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wiki</td>
<td>boolean</td>
<td>Wiki</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yours</td>
<td>boolean</td>
<td>Yours</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 19.1.7 putPostsByIdLocked

Lock a post from being edited

Catalog: Discourse

Schema: Posts

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /posts/{id}/locked

Insert Discourse API URL: /posts/{id}/locked

Update Discourse API URL: /posts/{id}/locked

Delete Discourse API URL: /posts/{id}/locked

Field Selection Method: NotRequired

Select Discourse API Operation: put /posts/{id}/locked

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putPostsByIdLocked. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
### Table Function Columns

The columns of the table function `putPostsByIdLocked` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 20 Schema: PrivateMessages

#### 20.1 Tables

**20.1.1 TopicsPrivateMessages_Topic_listTopicsByUsername**

Get a list of private messages for a user

Catalog: Discourse  
Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/topics/private-messages/{username}.json`  
**Insert Discourse API URL:** `/topics/private-messages/{username}.json`  
**Update Discourse API URL:** `/topics/private-messages/{username}.json`  
**Delete Discourse API URL:** `/topics/private-messages/{username}.json`

Field Selection Method: NotRequired  
Base Path: `topic_list.topics[*]`

**Select Discourse API Operation:** `get /topics/private-messages/{username}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `TopicsPrivateMessages_Topic_listTopicsByUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `TopicsPrivateMessages_Topic_listTopicsByUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
20.1.2 TopicsPrivateMessages_Topic_listTopicsParticipantsByUsername

Get a list of private messages for a user

Catalog: Discourse

Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /topics/private-messages/{username}.json

Insert Discourse API URL: /topics/private-messages/{username}.json

Update Discourse API URL: /topics/private-messages/{username}.json

Delete Discourse API URL: /topics/private-messages/{username}.json

Field Selection Method: NotRequired

Base Path: topic_list.topics[*].participants[*]

Select Discourse API Operation: get /topics/private-messages/{username}.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function TopicsPrivateMessages_Topic_listTopicsParticipantsByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table Function Columns

The columns of the table function `TopicsPrivateMessages_Topic_listTopicsParticipantsByUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bookmarks</td>
<td>boolean</td>
<td>Bookmarked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unread</td>
<td>decimal</td>
<td>Unread</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
20.1.3 TopicsPrivateMessages_Topic_listTopicsPostersByUsername

Get a list of private messages for a user

Catalog: Discourse

Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /topics/private-messages/{username}.json
Insert Discourse API URL: /topics/private-messages/{username}.json
Update Discourse API URL: /topics/private-messages/{username}.json
Delete Discourse API URL: /topics/private-messages/{username}.json

Field Selection Method: NotRequired

Base Path: topic_list.topics[*].posters[*]

Select Discourse API Operation: get /topics/private-messages/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function TopicsPrivateMessages_Topic_listTopicsPostersByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function TopicsPrivateMessages_Topic_listTopicsPostersByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------</td>
<td>---------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>extras</td>
<td>string</td>
<td>Extras</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>unread</td>
<td>decimal</td>
<td>Unread</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

### 20.1.4 TopicsPrivateMessagesByUsername
Get a list of private messages for a user

Catalog: Discourse

Schema: PrivateMessages
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/topics/private-messages/{username}.json`

Insert Discourse API URL: `/topics/private-messages/{username}.json`

Update Discourse API URL: `/topics/private-messages/{username}.json`

Delete Discourse API URL: `/topics/private-messages/{username}.json`

Field Selection Method: NotRequired

Base Path: `topic_list`

Select Discourse API Operation: get `/topics/private-messages/{username}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `TopicsPrivateMessagesByUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `TopicsPrivateMessagesByUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
20.1.5 TopicsPrivateMessagesSent_Topic_listTopicsByUsername

Get a list of private messages sent

Catalog: Discourse

Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /topics/private-messages-sent/{username}.json

Insert Discourse API URL: /topics/private-messages-sent/{username}.json

Update Discourse API URL: /topics/private-messages-sent/{username}.json

Delete Discourse API URL: /topics/private-messages-sent/{username}.json

Field Selection Method: NotRequired

Base Path: topic_list.topics[*]

Select Discourse API Operation: get /topics/private-messages-sent/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function TopicsPrivateMessagesSent_Topic_listTopicsByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function TopicsPrivateMessagesSent_Topic_listTopicsByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unread</td>
<td>decimal</td>
<td>Unread</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20.1.6 TopicsPrivateMessagesSent_Topic_listTopicsParticipantsByUsername

Get a list of private messages sent

Catalog: Discourse

Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /topics/private-messages-sent/{username}.json
Insert Discourse API URL: /topics/private-messages-sent/{username}.json
Update Discourse API URL: /topics/private-messages-sent/{username}.json
Delete Discourse API URL: /topics/private-messages-sent/{username}.json
Field Selection Method: NotRequired
Base Path: topic_list.topics[*].participants[*]
Select Discourse API Operation: get /topics/private-messages-sent/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function TopicsPrivateMessagesSent_Topic_listTopicsParticipantsByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function TopicsPrivateMessagesSent_Topic_listTopicsParticipantsByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unread</td>
<td>decimal</td>
<td>Unread</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20.1.7 Topics/PrivateMessages/Sent_TopicsPostersByUsername

Get a list of private messages sent

Catalog: Discourse

Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /topics/private-messages-sent/{username}.json
Insert Discourse API URL: /topics/private-messages-sent/{username}.json
Update Discourse API URL: /topics/private-messages-sent/{username}.json
Delete Discourse API URL: /topics/private-messages-sent/{username}.json
Field Selection Method: NotRequired
Base Path: topic_list.topics['].posters[*]
Select Discourse API Operation: `get /topics/private-messages-sent/{username}.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `TopicsPrivateMessagesSent_Topic_listTopicsPostersByUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `TopicsPrivateMessagesSent_Topic_listTopicsPostersByUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extras</td>
<td>string</td>
<td>Extras</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Discourse Data Model

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>unread</td>
<td>decimal</td>
<td>Unread</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

#### 20.1.8 TopicsPrivateMessagesSent_UsersByUsername

Get a list of private messages sent

**Catalog:** Discourse

**Schema:** PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /topics/private-messages-sent/{username}.json

**Insert Discourse API URL:** /topics/private-messages-sent/{username}.json

**Update Discourse API URL:** /topics/private-messages-sent/{username}.json

**Delete Discourse API URL:** /topics/private-messages-sent/{username}.json

**Field Selection Method:** NotRequired

**Base Path:** users[*]

**Select Discourse API Operation:** get /topics/private-messages-sent/{username}.json

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function TopicsPrivateMessagesSent_UsersByUsername. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function TopicsPrivateMessagesSent_UsersByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_sequence</td>
<td>decimal</td>
<td>Topic List Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>decimal</td>
<td>Topic List Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20.1.9 TopicsPrivateMessagesSentByUsername

Get a list of private messages sent

Catalog: Discourse

Schema: PrivateMessages

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /topics/private-messages-sent/{username}.json
Insert Discourse API URL: /topics/private-messages-sent/{username}.json
Update Discourse API URL: /topics/private-messages-sent/{username}.json
Delete Discourse API URL: /topics/private-messages-sent/{username}.json

Field Selection Method: NotRequired
Select Discourse API Operation: get /topics/private-messages-sent/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function TopicsPrivateMessagesSentByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function TopicsPrivateMessagesSentByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_sequence</td>
<td>decimal</td>
<td>Topic List Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>decimal</td>
<td>Topic List Per Page</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

21 Schema: Reports

21.1 Tables

21.1.1 PageViewTotalReqs

return pageview stats, defaults to one month

Catalog: Discourse

Schema: Reports

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /page_view_total_reqs
Insert Discourse API URL: /page_view_total_reqs
Update Discourse API URL: /page_view_total_reqs
Delete Discourse API URL: /page_view_total_reqs
Field Selection Method: NotRequired
Base Path: report
Select Discourse API Operation: get /page_view_total_reqs

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function PageViewTotalReqs. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>end_date</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>group_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>start_date</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function PageViewTotalReqs are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>end_date</td>
<td>string</td>
<td>End Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>group_id</td>
<td>int64</td>
<td>Group ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prev30Days</td>
<td>int64</td>
<td>Previous 30 Days</td>
<td></td>
<td></td>
</tr>
<tr>
<td>start_date</td>
<td>string</td>
<td>Start Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>int64</td>
<td>Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xaxis</td>
<td>string</td>
<td>X-Axis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yaxis</td>
<td>string</td>
<td>Y-Axis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
21.1.2 PageViewTotalReqs_ReportData

return pageview stats, defaults to one month

Catalog: Discourse

Schema: Reports

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /page_view_total_reqs
Insert Discourse API URL: /page_view_total_reqs
Update Discourse API URL: /page_view_total_reqs
Delete Discourse API URL: /page_view_total_reqs

Field Selection Method: Not Required

Base Path: report.data[*]

Select Discourse API Operation: get /page_view_total_reqs

Table Function Parameters

The following parameters can be used to control the behaviour of the table function PageViewTotalReqs_ReportData. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>category_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>end_date</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>group_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>start_date</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function PageViewTotalReqs_ReportData are shown below. Each column has an SQL data type.
### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postExportCsvByExportEntity`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `postExportCsvByExportEntity` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

**22 Schema: Search**

**22.1 Tables**

**22.1.1 Search_CategoriesQuery**

Search for something

Catalog: Discourse

Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /search/query

**Insert Discourse API URL:** /search/query

**Update Discourse API URL:** /search/query

**Delete Discourse API URL:** /search/query

Field Selection Method: NotRequired

Base Path: categories[*]

**Select Discourse API Operation:** get /search/query

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `Search_CategoriesQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `Search_CategoriesQuery` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>✔️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

**22.1.2 Search_Grouped_search_resultCategory_idsQuery**

Search for something

Catalog: Discourse

Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/search/query`

Insert Discourse API URL: `/search/query`

Update Discourse API URL: `/search/query`

Delete Discourse API URL: `/search/query`

Field Selection Method: NotRequired

Base Path: `grouped_search_result.category_ids[*]`

Select Discourse API Operation: get `/search/query`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `Search_Grouped_search_resultCategory_idsQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function Search_Grouped_search_resultCategory_idsQuery are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

22.1.3 Search_Grouped_search_resultPost_idsQuery

Search for something

Catalog: Discourse

Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /search/query

Insert Discourse API URL: /search/query

Update Discourse API URL: /search/query

Delete Discourse API URL: /search/query

Field Selection Method: NotRequired

Base Path: grouped_search_result.post_ids[*]

Select Discourse API Operation: get /search/query

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `Search_Grouped_search_resultPost_idsQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td>☑</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `Search_Grouped_search_resultPost_idsQuery` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

**22.1.4 Search_Grouped_search_resultUser_idsQuery**

Search for something

Catalog: Discourse

Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/search/query`

Insert Discourse API URL: `/search/query`

Update Discourse API URL: `/search/query`

Delete Discourse API URL: `/search/query`

Field Selection Method: NotRequired

Base Path: `grouped_search_result.user_ids[*]`

Select Discourse API Operation: `get /search/query`
### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `Search_Grouped_search_resultUser_idsQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `Search_Grouped_search_resultUser_idsQuery` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

#### 22.1.5 Search_PostsQuery

Search for something
Catalog: Discourse
Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/search/query`
Insert Discourse API URL: `/search/query`
Update Discourse API URL: `/search/query`
Delete Discourse API URL: `/search/query`
Field Selection Method: NotRequired
Base Path: `posts[*]`
Select Discourse API Operation: get /search/query

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `Search_PostsQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `Search_PostsQuery` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>blurb</td>
<td>string</td>
<td>Blurb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**22.1.6 Search_TopicsPostersQuery**

Search for something
Catalog: Discourse
Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API URL: /search/query
Insert Discourse API URL: /search/query
Update Discourse API URL: /search/query
Delete Discourse API URL: /search/query
Field Selection Method: NotRequired
Base Path: topics[*].posters[*]
Select Discourse API Operation: get /search/query

Table Function Parameters

The following parameters can be used to control the behaviour of the table function Search_TopicsPostersQuery. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function Search_TopicsPostersQuery are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>--------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**22.1.7 Search_TopicsQuery**

Search for something

Catalog: Discourse

Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/search/query`

Insert Discourse API URL: `/search/query`

Update Discourse API URL: `/search/query`

Delete Discourse API URL: `/search/query`

Field Selection Method: NotRequired

Base Path: `topics[*]`

Select Discourse API Operation: `get /search/query`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `Search_TopicsQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `Search_TopicsQuery` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globaly</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**22.1.8 Search_UsersQuery**

Search for something
Catalog: Discourse
Schema: Search
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/search/query`
Insert Discourse API URL: `/search/query`
Update Discourse API URL: `/search/query`
Delete Discourse API URL: `/search/query`
Field Selection Method: NotRequired
Base Path: `users[*]`
Select Discourse API Operation: get `/search/query`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `Search_UsersQuery`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `Search_UsersQuery` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td>☑️</td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

**22.1.9 SearchQuery**

Search for something
Catalog: Discourse
Schema: Search

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /search/query
Insert Discourse API URL: /search/query
Update Discourse API URL: /search/query
Delete Discourse API URL: /search/query
Field Selection Method: NotRequired
Select Discourse API Operation: get /search/query

Table Function Parameters

The following parameters can be used to control the behaviour of the table function SearchQuery. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_blurbs</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function SearchQuery are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
23 Schema: SiteSettingsBackups
23.1 Tables
23.1.1 putAdminSiteSettingsAllowRestore

allow_restore

Catalog: Discourse
Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_restore
Insert Discourse API URL: /admin/site_settings/allow_restore
Update Discourse API URL: /admin/site_settings/allow_restore
Delete Discourse API URL: /admin/site_settings/allow_restore

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/allow_restore

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowRestore. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAllowRestore are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
23.1.2 putAdminSiteSettingsAutomaticBackupsEnabled

automatic backups enabled

Catalog: Discourse
Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/automatic_backups_enabled
Insert Discourse API URL: /admin/site_settings/automatic_backups_enabled
Update Discourse API URL: /admin/site_settings/automatic_backups_enabled
Delete Discourse API URL: /admin/site_settings/automatic_backups_enabled
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/automatic_backups_enabled

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutomaticBackupsEnabled. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAutomaticBackupsEnabled are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
23.1.3 putAdminSiteSettingsBackupFrequency

backup frequency
Catalog: Discourse
Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/backup_frequency
Insert Discourse API URL: /admin/site_settings/backup_frequency
Update Discourse API URL: /admin/site_settings/backup_frequency
Delete Discourse API URL: /admin/site_settings/backup_frequency
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/backup_frequency

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsBackupFrequency. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsBackupFrequency are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
23.1.4 putAdminSiteSettingsBackupTimeOfDay

backup time of day

Catalog: Discourse

Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/backup_time_of_day

Insert Discourse API URL: /admin/site_settings/backup_time_of_day

Update Discourse API URL: /admin/site_settings/backup_time_of_day

Delete Discourse API URL: /admin/site_settings/backup_time_of_day

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/backup_time_of_day

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsBackupTimeOfDay. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsBackupTimeOfDay are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
23.1.5 putAdminSiteSettingsBackupWithUploads

backup with uploads

Catalog: Discourse

Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/backup_with_uploads

Insert Discourse API URL: /admin/site_settings/backup_with_uploads

Update Discourse API URL: /admin/site_settings/backup_with_uploads

Delete Discourse API URL: /admin/site_settings/backup_with_uploads

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/backup_with_uploads

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsBackupWithUploads. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsBackupWithUploads are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
23.1.6 putAdminSiteSettingsEnableS3Backups

enable s3 backups

Catalog: Discourse

Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_s3_backups
Insert Discourse API URL: /admin/site_settings/enable_s3_backups
Update Discourse API URL: /admin/site_settings/enable_s3_backups
Delete Discourse API URL: /admin/site_settings/enable_s3_backups

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/enable_s3_backups

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEnableS3Backups. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsEnableS3Backups are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
23.1.7 putAdminSiteSettingsMaximumBackups
maximum backups
Catalog: Discourse
Schema: SiteSettingsBackups
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API URL: /admin/site_settings/maximum_backups
Insert Discourse API URL: /admin/site_settings/maximum_backups
Update Discourse API URL: /admin/site_settings/maximum_backups
Delete Discourse API URL: /admin/site_settings/maximum_backups
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/maximum_backups

Table Function Parameters
The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsMaximumBackups. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.
Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsMaximumBackups are shown
below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
23.1.8 putAdminSiteSettingsReadonlyModeDuringBackup

readonly_mode_during_backup

Catalog: Discourse

Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/readonly_mode_during_backup

Insert Discourse API
URL: /admin/site_settings/readonly_mode_during_backup

Update Discourse API
URL: /admin/site_settings/readonly_mode_during_backup

Delete Discourse API
URL: /admin/site_settings/readonly_mode_during_backup

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/readonly_mode_during_backup

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsReadonlyModeDuringBackup. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsReadonlyModeDuringBackup are shown below. Each column has an SQL data type.
### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsS3BackupBucket. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function putAdminSiteSettingsS3BackupBucket are shown below. Each column has an SQL data type.
23.1.10 putAdminSiteSettingsS3DisableCleanup

s3 disable cleanup
Catalog: Discourse
Schema: SiteSettingsBackups

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/s3_disable_cleanup
Insert Discourse API URL: /admin/site_settings/s3_disable_cleanup
Update Discourse API URL: /admin/site_settings/s3_disable_cleanup
Delete Discourse API URL: /admin/site_settings/s3_disable_cleanup

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/s3_disable_cleanup

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsS3DisableCleanup. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function putAdminSiteSettingsS3DisableCleanup are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
24 Schema: SiteSettingsBasicSetup
24.1 Tables
24.1.1 putAdminSiteSettingsAllowUserLocale

Update allow user locale
Catalog: Discourse
Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_user_locale
Insert Discourse API URL: /admin/site_settings/allow_user_locale
Update Discourse API URL: /admin/site_settings/allow_user_locale
Delete Discourse API URL: /admin/site_settings/allow_user_locale
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/allow_user_locale

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowUserLocale. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAllowUserLocale are shown below. Each column has an SQL data type.
### 24.1.2 putAdminSiteSettingsCategoriesTopics

Update the number of topics to show

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/categories_topics`

Insert Discourse API URL: `/admin/site_settings/categories_topics`

Update Discourse API URL: `/admin/site_settings/categories_topics`

Delete Discourse API URL: `/admin/site_settings/categories_topics`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/categories_topics`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsCategoriesTopics`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsCategoriesTopics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
24.1.3 putAdminSiteSettingsCategoryColors

Update a list of hexadecimal color values

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/category_colors

Insert Discourse API URL: /admin/site_settings/category_colors

Update Discourse API URL: /admin/site_settings/category_colors

Delete Discourse API URL: /admin/site_settings/category_colors

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/category_colors

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCategoryColors. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function putAdminSiteSettingsCategoryColors are shown below. Each column has an SQL data type.
24.1.4 putAdminSiteSettingsCategoryStyle

Update the category style

Catalog: Discourse
Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/category_style
Insert Discourse API URL: /admin/site_settings/category_style
Update Discourse API URL: /admin/site_settings/category_style
Delete Discourse API URL: /admin/site_settings/category_style

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/category_style

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsCategoryStyle. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsCategoryStyle are shown
below. Each column has an SQL data type.
24.1.5 putAdminSiteSettingsDefaultLocale

Update the default locale

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL:/admin/site_settings/default_locale
Insert Discourse API URL:/admin/site_settings/default_locale
Update Discourse API URL:/admin/site_settings/default_locale
Delete Discourse API URL:/admin/site_settings/default_locale

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/default_locale

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultLocale. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultLocale are shown below. Each column has an SQL data type.
24.1.6 putAdminSiteSettingsDesktopCategoryPageStyle

Update the visual style for the /categories page

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/desktop_category_page_style`

Insert Discourse API
URL: `/admin/site_settings/desktop_category_page_style`

Update Discourse API
URL: `/admin/site_settings/desktop_category_page_style`

Delete Discourse API
URL: `/admin/site_settings/desktop_category_page_style`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/desktop_category_page_style`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDesktopCategoryPageStyle`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The columns of the table function
putAdminSiteSettingsDesktopCategoryPageStyle are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.7 putAdminSiteSettingsEnableBadges

Update enable the badge system

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_badges

Insert Discourse API URL: /admin/site_settings/enable_badges

Update Discourse API URL: /admin/site_settings/enable_badges

Delete Discourse API URL: /admin/site_settings/enable_badges

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/enable_badges

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEnableBadges. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `putAdminSiteSettingsEnableBadges` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 24.1.8 `putAdminSiteSettingsEnableMobileTheme`

Update enable mobile theme

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_mobile_theme

Insert Discourse API URL: /admin/site_settings/enable_mobile_theme

Update Discourse API URL: /admin/site_settings/enable_mobile_theme

Delete Discourse API URL: /admin/site_settings/enable_mobile_theme

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_mobile_theme

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableMobileTheme`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns
The columns of the table function `putAdminSiteSettingsEnableMobileTheme` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 24.1.9 putAdminSiteSettingsEnableWhispers

Update enable whispers

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_whispers`

Insert Discourse API URL: `/admin/site_settings/enable_whispers`

Update Discourse API URL: `/admin/site_settings/enable_whispers`

Delete Discourse API URL: `/admin/site_settings/enable_whispers`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/enable_whispers`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableWhispers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableWhispers` are shown below. Each column has an SQL data type.
24.1.10 putAdminSiteSettingsFixedCategoryPositions

Update fixed category positions

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/fixed_category_positions

Insert Discourse API URL: /admin/site_settings/fixed_category_positions

Update Discourse API URL: /admin/site_settings/fixed_category_positions

Delete Discourse API URL: /admin/site_settings/fixed_category_positions

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/fixed_category_positions

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsFixedCategoryPositions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsFixedCategoryPositions are shown below. Each column has an SQL data type.
24.1.11 putAdminSiteSettingsFixedCategoryPositionsOnCreate

Update fixed category positions on create

Catalog: Discourse
Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/fixed_category_positions_on_create

Insert Discourse API
URL: /admin/site_settings/fixed_category_positions_on_create

Update Discourse API
URL: /admin/site_settings/fixed_category_positions_on_create

Delete Discourse API
URL: /admin/site_settings/fixed_category_positions_on_create

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/fixed_category_positions_on_create

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsFixedCategoryPositionsOnCreate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsFixedCategoryPositionsOnCreate` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.12 putAdminSiteSettingsGaDomainName

Update the OBSOLETE ga domain name

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/ga_domain_name`

Insert Discourse API URL: `/admin/site_settings/ga_domain_name`

Update Discourse API URL: `/admin/site_settings/ga_domain_name`

Delete Discourse API URL: `/admin/site_settings/ga_domain_name`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/ga_domain_name`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGaDomainName`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsGaDomainName are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.13 putAdminSiteSettingsGaTrackingCode

Update the OBSOLETE ga tracking code

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/ga_tracking_code

Insert Discourse API URL: /admin/site_settings/ga_tracking_code

Update Discourse API URL: /admin/site_settings/ga_tracking_code

Delete Discourse API URL: /admin/site_settings/ga_tracking_code

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/ga_tracking_code

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsGaTrackingCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `putAdminSiteSettingsGaTrackingCode` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 24.1.14 `putAdminSiteSettingsGaUniversalDomainName`

Update Google Universal Analytics domain name

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/ga_universal_domain_name`

Insert Discourse API URL: `/admin/site_settings/ga_universal_domain_name`

Update Discourse API URL: `/admin/site_settings/ga_universal_domain_name`

Delete Discourse API URL: `/admin/site_settings/ga_universal_domain_name`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/ga_universal_domain_name`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGaUniversalDomainName`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default value for the fourth parameter and the specified values for the first and third.

### Table Function Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The columns of the table function `putAdminSiteSettingsGaUniversalDomainName` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 24.1.15 putAdminSiteSettingsGaUniversalTrackingCode

Update Google Universal Analytics tracking code

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/ga_universal_tracking_code`

Insert Discourse API URL: `/admin/site_settings/ga_universal_tracking_code`

Update Discourse API

URL: `/admin/site_settings/ga_universal_tracking_code`

Delete Discourse API

URL: `/admin/site_settings/ga_universal_tracking_code`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/ga_universal_tracking_code`

#### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGaUniversalTrackingCode`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsGaUniversalTrackingCode` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.16 `putAdminSiteSettingsGtmContainerId`

Update Google Tag Manager container id

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/gtm_container_id`

Insert Discourse API URL: `/admin/site_settings/gtm_container_id`

Update Discourse API URL: `/admin/site_settings/gtm_container_id`

Delete Discourse API URL: `/admin/site_settings/gtm_container_id`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/gtm_container_id`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGtmContainerId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsGtmContainerId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.17 putAdminSiteSettingsLimitSuggestedToCategory

Update show topics from the current category

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/limit_suggested_to_category`

Insert Discourse API
URL: `/admin/site_settings/limit_suggested_to_category`

Update Discourse API
URL: `/admin/site_settings/limit_suggested_to_category`

Delete Discourse API
URL: `/admin/site_settings/limit_suggested_to_category`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/limit_suggested_to_category`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLimitSuggestedToCategory`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsLimitSuggestedToCategory` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.18 putAdminSiteSettingsPostMenu

Update the post menu items

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/post_menu`

Insert Discourse API URL: `/admin/site_settings/post_menu`

Update Discourse API URL: `/admin/site_settings/post_menu`

Delete Discourse API URL: `/admin/site_settings/post_menu`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/post_menu

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPostMenu`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
### Table Function Columns

The columns of the table function `putAdminSiteSettingsPostMenu` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 24.1.19 putAdminSiteSettingsPostMenuHiddenItems

Update the post menu hidden items

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/post_menu_hidden_items`

Insert Discourse API URL: `/admin/site_settings/post_menu_hidden_items`

Update Discourse API URL: `/admin/site_settings/post_menu_hidden_items`

Delete Discourse API URL: `/admin/site_settings/post_menu_hidden_items`

Field Selection Method: NotRequired

Select Discourse API Operation: `put` /`admin/site_settings/post_menu_hidden_items`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPostMenuHiddenItems`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
**Table Function Columns**

The columns of the table function `putAdminSiteSettingsPostMenuHiddenItems` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**24.1.20 putAdminSiteSettingsRelativeDateDuration**

Update relative date duration

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/relative_date_duration`

Insert Discourse API URL: `/admin/site_settings/relative_date_duration`

Update Discourse API URL: `/admin/site_settings/relative_date_duration`

Delete Discourse API URL: `/admin/site_settings/relative_date_duration`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/relative_date_duration`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsRelativeDateDuration`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
**Table Function Columns**

The columns of the table function `putAdminSiteSettingsRelativeDateDuration` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**24.1.21 putAdminSiteSettingsSetLocaleFromAcceptLanguageHeader**

Update set locale from accept language header

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/set_locale_from_accept_language_header

Insert Discourse API

URL: /admin/site_settings/set_locale_from_accept_language_header

Update Discourse API

URL: /admin/site_settings/set_locale_from_accept_language_header

Delete Discourse API

URL: /admin/site_settings/set_locale_from_accept_language_header

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/set_locale_from_accept_language_header

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSetLocaleFromAcceptLanguageHeader`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSetLocaleFromAcceptLanguageHeader` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 24.1.22 putAdminSiteSettingsShowSubcategoryList

Update show subcategory list

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/show_subcategory_list`

Insert Discourse API URL: `/admin/site_settings/show_subcategory_list`

Update Discourse API URL: `/admin/site_settings/show_subcategory_list`

Delete Discourse API URL: `/admin/site_settings/show_subcategory_list`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/show_subcategory_list

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShowSubcategoryList`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsShowSubcategoryList` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**24.1.23 putAdminSiteSettingsSuggestedTopics**

Update the number of suggested topics.

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/suggested_topics

Insert Discourse API URL: /admin/site_settings/suggested_topics

Update Discourse API URL: /admin/site_settings/suggested_topics

Delete Discourse API URL: /admin/site_settings/suggested_topics

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/suggested_topics

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSuggestedTopics`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSuggestedTopics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 24.1.24 putAdminSiteSettingsSuggestedTopicsMaxDaysOld

Update suggested topics max days old

Catalog: Discourse  
Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API  
URL: `/admin/site_settings/suggested_topics_max_days_old`

Insert Discourse API  
URL: `/admin/site_settings/suggested_topics_max_days_old`

Update Discourse API  
URL: `/admin/site_settings/suggested_topics_max_days_old`

Delete Discourse API  
URL: `/admin/site_settings/suggested_topics_max_days_old`

Field Selection Method: NotRequired

Select Discourse API Operation:  
put /admin/site_settings/suggested_topics_max_days_old

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSuggestedTopicsMaxDaysOld`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsSuggestedTopicsMaxDaysOld` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

24.1.25 putAdminSiteSettingsTopMenu

Update the homepage navigation

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/top_menu

Insert Discourse API URL: /admin/site_settings/top_menu

Update Discourse API URL: /admin/site_settings/top_menu

Delete Discourse API URL: /admin/site_settings/top_menu

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/top_menu

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTopMenu`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTopMenu` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 24.1.26 putAdminSiteSettingsTrackExternalRightClicks

Update track external right clicks

Catalog: Discourse

Schema: SiteSettingsBasicSetup

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/track_external_right_clicks`

Insert Discourse API

URL: `/admin/site_settings/track_external_right_clicks`

Update Discourse API

URL: `/admin/site_settings/track_external_right_clicks`

Delete Discourse API

URL: `/admin/site_settings/track_external_right_clicks`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/track_external_right_clicks

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTrackExternalRightClicks`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four...
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTrackExternalRightClicks` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 25 Schema: SiteSettingsDeveloper

#### 25.1 Tables

**25.1.1 putAdminSiteSettingsActiveUserRateLimitSecs**

active user rate limit secs

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/active_user_rate_limit_secs

Insert Discourse API
URL: /admin/site_settings/active_user_rate_limit_secs

Update Discourse API
URL: /admin/site_settings/active_user_rate_limit_secs

Delete Discourse API
URL: /admin/site_settings/active_user_rate_limit_secs

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/active_user_rate_limit_secs

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsActiveUserRateLimitSecs`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsActiveUserRateLimitSecs` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

25.1.2 `putAdminSiteSettingsAnonPollingInterval`

anon polling interval

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/anon_polling_interval`

Insert Discourse API URL: `/admin/site_settings/anon_polling_interval`

Update Discourse API URL: `/admin/site_settings/anon_polling_interval`

Delete Discourse API URL: `/admin/site_settings/anon_polling_interval`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/anon_polling_interval

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAnonPollingInterval`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAnonPollingInterval` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**25.1.3 putAdminSiteSettingsBackgroundPollingInterval**

background polling interval

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/background_polling_interval

Insert Discourse API

URL: /admin/site_settings/background_polling_interval

Update Discourse API

URL: /admin/site_settings/background_polling_interval

Delete Discourse API

URL: /admin/site_settings/background_polling_interval

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/background_polling_interval
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsBackgroundPollingInterval. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsBackgroundPollingInterval are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

25.1.4 putAdminSiteSettingsEnableLongPolling

enable long polling
Catalog: Discourse
Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_long_polling
Insert Discourse API URL: /admin/site_settings/enable_long_polling
Update Discourse API URL: /admin/site_settings/enable_long_polling
Delete Discourse API URL: /admin/site_settings/enable_long_polling
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/enable_long_polling
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableLongPolling`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableLongPolling` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

25.1.5 `putAdminSiteSettingsFlushTimingsSecs`

flush timings secs

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/flush_timings_secs
Insert Discourse API URL: /admin/site_settings/flush_timings_secs
Update Discourse API URL: /admin/site_settings/flush_timings_secs
Delete Discourse API URL: /admin/site_settings/flush_timings_secs

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/flush_timings_secs
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFlushTimingsSecs`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsFlushTimingsSecs` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

25.1.6 putAdminSiteSettingsForceHostname

force hostname

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/force_hostname

Insert Discourse API URL: /admin/site_settings/force_hostname

Update Discourse API URL: /admin/site_settings/force_hostname

Delete Discourse API URL: /admin/site_settings/force_hostname

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/force_hostname

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsForceHostname`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsForceHostname` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**25.1.7 putAdminSiteSettingsLongPollingBaseUrl**

long polling base url

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/long_polling_base_url`

Insert Discourse API URL: `/admin/site_settings/long_polling_base_url`

Update Discourse API URL: `/admin/site_settings/long_polling_base_url`

Delete Discourse API URL: `/admin/site_settings/long_polling_base_url`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/long_polling_base_url

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLongPollingBaseUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsLongPollingBaseUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 25.1.8 putAdminSiteSettingsLongPollingInterval

long polling interval

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/long_polling_interval`

Insert Discourse API URL: `/admin/site_settings/long_polling_interval`

Update Discourse API URL: `/admin/site_settings/long_polling_interval`

Delete Discourse API URL: `/admin/site_settings/long_polling_interval`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/long_polling_interval

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLongPollingInterval`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with ``select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsLongPollingInterval` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**25.1.9 putAdminSiteSettingsPollingInterval**

polling interval

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/polling_interval

Insert Discourse API URL: /admin/site_settings/polling_interval

Update Discourse API URL: /admin/site_settings/polling_interval

Delete Discourse API URL: /admin/site_settings/polling_interval

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/polling_interval

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPollingInterval`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsPollingInterval are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

25.1.10 putAdminSiteSettingsPort

port

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/port

Insert Discourse API URL: /admin/site_settings/port

Update Discourse API URL: /admin/site_settings/port

Delete Discourse API URL: /admin/site_settings/port

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/port

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsPort. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a
pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsPort` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**25.1.11 putAdminSiteSettingsQueueJobs**

queue jobs

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/queue_jobs

Insert Discourse API URL: /admin/site_settings/queue_jobs

Update Discourse API URL: /admin/site_settings/queue_jobs

Delete Discourse API URL: /admin/site_settings/queue_jobs

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/queue_jobs

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsQueueJobs`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsQueueJobs` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**25.1.12 putAdminSiteSettingsRebakeOldPostsCount**

rebake old posts count

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/rebake_old_posts_count`

Insert Discourse API URL: `/admin/site_settings/rebake_old_posts_count`

Update Discourse API URL: `/admin/site_settings/rebake_old_posts_count`

Delete Discourse API URL: `/admin/site_settings/rebake_old_posts_count`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/rebake_old_posts_count

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsRebakeOldPostsCount`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsRebakeOldPostsCount` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 25.1.13 putAdminSiteSettingsTopTopicsFormulaFirstPostLikesMultiplier

top topics formula first post likes multiplier

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/top_topics_formula_first_post_likes_multiplier

Insert Discourse API

URL: /admin/site_settings/top_topics_formula_first_post_likes_multiplier

Update Discourse API

URL: /admin/site_settings/top_topics_formula_first_post_likes_multiplier

Delete Discourse API

URL: /admin/site_settings/top_topics_formula_first_post_likes_multiplier

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/top_topics_formula_first_post_likes_multiplier
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTopTopicsFormulaFirstPostLikesMultiplier`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsTopTopicsFormulaFirstPostLikesMultiplier` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
Update Discourse API
URL: /admin/site_settings/top_topics_formula_least_likes_per_post_multiplier

Delete Discourse API
URL: /admin/site_settings/top_topics_formula_least_likes_per_post_multiplier

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/top_topics_formula_least_likes_per_post_multiplier

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopTopicsFormulaLeastLikesPerPostMultiplier. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTopTopicsFormulaLeastLikesPerPostMultiplier are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

25.1.15 putAdminSiteSettingsTopTopicsFormulaLogViewsMultiplier
top topics formula log views multiplier
Catalog: Discourse
Schema: SiteSettingsDeveloper
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/top_topics_formula_log_views_multiplier`

Insert Discourse API
URL: `/admin/site_settings/top_topics_formula_log_views_multiplier`

Update Discourse API
URL: `/admin/site_settings/top_topics_formula_log_views_multiplier`

Delete Discourse API
URL: `/admin/site_settings/top_topics_formula_log_views_multiplier`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/top_topics_formula_log_views_multiplier`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTopTopicsFormulaLogViewsMultiplier`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsTopTopicsFormulaLogViewsMultiplier` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
25.1.16 putAdminSiteSettingsVerboseLocalization

verbose localization

Catalog: Discourse

Schema: SiteSettingsDeveloper

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/verbose_localization

Insert Discourse API URL: /admin/site_settings/verbose_localization

Update Discourse API URL: /admin/site_settings/verbose_localization

Delete Discourse API URL: /admin/site_settings/verbose_localization

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/verbose_localization

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsVerboseLocalization. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with "select * from table(value1, value2, value3)" on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with "select * from table(name1 => value1, name3 => value3)" on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsVerboseLocalization are
shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
26 Schema: SiteSettingsEmail
26.1 Tables
26.1.1 putAdminSiteSettingsAlternativeReplyByEmailAddresses

alternative reply by email addresses

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/alternative_reply_by_email_addresses

Insert Discourse API
URL: /admin/site_settings/alternative_reply_by_email_addresses

Update Discourse API
URL: /admin/site_settings/alternative_reply_by_email_addresses

Delete Discourse API
URL: /admin/site_settings/alternative_reply_by_email_addresses

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/alternative_reply_by_email_addresses

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAlternativeReplyByEmailAddresses. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>false</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function 
`putAdminSiteSettingsAlternativeReplyByEmailAddress` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 26.1.2 putAdminSiteSettingsAlwaysShowTrimmedContent

always show trimmed content

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

**URL:** `/admin/site_settings/always_show_trimmed_content`

Insert Discourse API

**URL:** `/admin/site_settings/always_show_trimmed_content`

Update Discourse API

**URL:** `/admin/site_settings/always_show_trimmed_content`

Delete Discourse API

**URL:** `/admin/site_settings/always_show_trimmed_content`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/always_show_trimmed_content`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAlwaysShowTrimmedContent`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsAlwaysShowTrimmedContent are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.3 putAdminSiteSettingsAttachmentContentTypeBlacklist

attachment content type blacklist

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/attachment_content_type_blacklist

Insert Discourse API
URL: /admin/site_settings/attachment_content_type_blacklist

Update Discourse API
URL: /admin/site_settings/attachment_content_type_blacklist

Delete Discourse API
URL: /admin/site_settings/attachment_content_type_blacklist

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/attachment_content_type_blacklist

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAttachmentContentTypeBlacklist. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsAttachmentContentTypeBlacklist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

26.1.4 putAdminSiteSettingsAttachmentFilenameBlacklist

attachment filename blacklist

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/attachment_filename_blacklist`

Insert Discourse API
URL: `/admin/site_settings/attachment_filename_blacklist`

Update Discourse API
URL: `/admin/site_settings/attachment_filename_blacklist`

Delete Discourse API
URL: `/admin/site_settings/attachment_filename_blacklist`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/attachment_filename_blacklist

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAttachmentFilenameBlacklist`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAttachmentFilenameBlacklist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.5 putAdminSiteSettingsAutoGeneratedWhitelist**

auto generated whitelist

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/auto_generated_whitelist`

Insert Discourse API URL: `/admin/site_settings/auto_generated_whitelist`

Update Discourse API URL: `/admin/site_settings/auto_generated_whitelist`

Delete Discourse API URL: `/admin/site_settings/auto_generated_whitelist`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/auto_generated_whitelist

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAutoGeneratedWhitelist`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAutoGeneratedWhitelist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.6 putAdminSiteSettingsBlockAutoGeneratedEmails

block auto generated emails

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/block_auto_generated_emails`

Insert Discourse API

URL: `/admin/site_settings/block_auto_generated_emails`

Update Discourse API

URL: `/admin/site_settings/block_auto_generated_emails`

Delete Discourse API

URL: `/admin/site_settings/block_auto_generated_emails`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/block_auto_generated_emails

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsBlockAutoGeneratedEmails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsBlockAutoGeneratedEmails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.7 putAdminSiteSettingsBounceScoreThreshold

bounce score threshold

Catalog: Discourse  
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/bounce_score_threshold  
Insert Discourse API URL: /admin/site_settings/bounce_score_threshold  
Update Discourse API URL: /admin/site_settings/bounce_score_threshold  
Delete Discourse API URL: /admin/site_settings/bounce_score_threshold  
Field Selection Method: NotRequired  
Select Discourse API Operation:  
put /admin/site_settings/bounce_score_threshold

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsBounceScoreThreshold`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsBounceScoreThresholdDeactivate` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODY</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.8 putAdminSiteSettingsBounceScoreThresholdDeactivate

bounce score threshold deactivate

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/bounce_score_threshold_deactivate

Insert Discourse API
URL: /admin/site_settings/bounce_score_threshold_deactivate

Update Discourse API
URL: /admin/site_settings/bounce_score_threshold_deactivate

Delete Discourse API
URL: /admin/site_settings/bounce_score_threshold_deactivate

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/bounce_score_threshold_deactivate

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsBounceScoreThresholdDeactivate`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by
position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsBounceScoreThresholdDeactivate` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.9 `putAdminSiteSettingsDeleteDigestEmailAfterDays`

delete digest email after days

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/delete_digest_email_after_days

Insert Discourse API
URL: /admin/site_settings/delete_digest_email_after_days

Update Discourse API
URL: /admin/site_settings/delete_digest_email_after_days

Delete Discourse API
URL: /admin/site_settings/delete_digest_email_after_days

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/delete_digest_email_after_days

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDeleteDigestEmailAfterDays`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDeleteDigestEmailAfterDays` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 26.1.10 putAdminSiteSettingsDeleteEmailLogsAfterDays

delete email logs after days

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/delete_email_logs_after_days

Insert Discourse API

URL: /admin/site_settings/delete_email_logs_after_days

Update Discourse API

URL: /admin/site_settings/delete_email_logs_after_days

Delete Discourse API

URL: /admin/site_settings/delete_email_logs_after_days

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/delete_email_logs_after_days

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDeleteEmailLogsAfterDays. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDeleteEmailLogsAfterDays are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.11 putAdminSiteSettingsDigestMinExcerptLength
digest min excerpt length

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the tableNativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/digest_min_excerpt_length
Insert Discourse API URL: /admin/site_settings/digest_min_excerpt_length
Update Discourse API URL: /admin/site_settings/digest_min_excerpt_length
Delete Discourse API URL: /admin/site_settings/digest_min_excerpt_length
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/digest_min_excerpt_length

**Table Function Parameters**

The following parameters can be used to control the behavior of the table function `putAdminSiteSettingsDigestMinExcerptLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behavior. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsDigestMinExcerptLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.12 putAdminSiteSettingsDigestOtherTopics
digest other topics

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/digest_other_topics
Insert Discourse API URL: /admin/site_settings/digest_other_topics
Update Discourse API URL: /admin/site_settings/digest_other_topics
Delete Discourse API URL: /admin/site_settings/digest_other_topics
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/digest_other_topics

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDigestOtherTopics. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDigestOtherTopics are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.13 putAdminSiteSettingsDigestPosts

digest posts

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/digest_posts

Insert Discourse API URL: /admin/site_settings/digest_posts

Update Discourse API URL: /admin/site_settings/digest_posts

Delete Discourse API URL: /admin/site_settings/digest_posts

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/digest_posts
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDigestPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDigestPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.14 putAdminSiteSettingsDigestSuppressCategories

digest suppress categories

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/digestSuppressCategories

Insert Discourse API URL: /admin/site_settings/digestSuppressCategories

Update Discourse API URL: /admin/site_settings/digestSuppressCategories

Delete Discourse API URL: /admin/site_settings/digestSuppressCategories

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/digest_suppress_categories

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDigestSuppressCategories. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsDigestSuppressCategories are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.15 putAdminSiteSettingsDigestTopics**

digest topics
Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/digest_topics
Insert Discourse API URL: /admin/site_settings/digest_topics
Update Discourse API URL: /admin/site_settings/digest_topics
Delete Discourse API URL: /admin/site_settings/digest_topics
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/digest_topics

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDigestTopics`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDigestTopics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.16 putAdminSiteSettingsDisableDigestEmails

disable digest emails

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/disable_digest_emails

Insert Discourse API URL: /admin/site_settings/disable_digest_emails

Update Discourse API URL: /admin/site_settings/disable_digest_emails

Delete Discourse API URL: /admin/site_settings/disable_digest_emails

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/disable_digest_emails
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDisableDigestEmails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDisableDigestEmails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.17 putAdminSiteSettingsDisableEmails

disable emails

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/disable_emails

Insert Discourse API URL: /admin/site_settings/disable_emails

Update Discourse API URL: /admin/site_settings/disable_emails

Delete Discourse API URL: /admin/site_settings/disable_emails

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/disable_emails

Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDisableEmails. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsDisableEmails are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.18 putAdminSiteSettingsDisplayNameOnEmailFrom**

display name on email from

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/display_name_on_email_from

Insert Discourse API URL: /admin/site_settings/display_name_on_email_from

Update Discourse API
URL: /admin/site_settings/display_name_on_email_from

Delete Discourse API
URL: /admin/site_settings/display_name_on_email_from

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/display_name_on_email_from

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDisplayNameOnEmailFrom. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDisplayNameOnEmailFrom are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.19 putAdminSiteSettingsEmailAccentBgColor

email accent bg color

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_accent_bg_color

Insert Discourse API URL: /admin/site_settings/email_accent_bg_color

Update Discourse API URL: /admin/site_settings/email_accent_bg_color

Delete Discourse API URL: /admin/site_settings/email_accent_bg_color

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_accent_bg_color
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEmailAccentBgColor. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsEmailAccentBgColor are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.20 putAdminSiteSettingsEmailAccentFgColor

e-mail accent fg color

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_accent_fg_color

Insert Discourse API URL: /admin/site_settings/email_accent_fg_color

Update Discourse API URL: /admin/site_settings/email_accent_fg_color

Delete Discourse API URL: /admin/site_settings/email_accent_fg_color

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/email_accent_fg_color
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailAccentFgColor`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsEmailAccentFgColor` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.21 putAdminSiteSettingsEmailCustomHeaders

email custom headers

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_custom_headers

Insert Discourse API URL: /admin/site_settings/email_custom_headers

Update Discourse API URL: /admin/site_settings/email_custom_headers

Delete Discourse API URL: /admin/site_settings/email_custom_headers

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_custom_headers
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailCustomHeaders`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsEmailCustomHeaders` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.22 putAdminSiteSettingsEmailIn

e-mail in

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_in

Insert Discourse API URL: /admin/site_settings/email_in

Update Discourse API URL: /admin/site_settings/email_in

Delete Discourse API URL: /admin/site_settings/email_in

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_in

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailIn`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEmailIn` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.23 putAdminSiteSettingsEmailInMinTrust

email in min trust

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/email_in_min_trust`

Insert Discourse API URL: `/admin/site_settings/email_in_min_trust`

Update Discourse API URL: `/admin/site_settings/email_in_min_trust`

Delete Discourse API URL: `/admin/site_settings/email_in_min_trust`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/email_in_min_trust`

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailInMinTrust`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEmailInMinTrust` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 26.1.24 putAdminSiteSettingsEmailLinkColor

**email link color**

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/email_link_color`

Insert Discourse API URL: `/admin/site_settings/email_link_color`

Update Discourse API URL: `/admin/site_settings/email_link_color`

Delete Discourse API URL: `/admin/site_settings/email_link_color`

Field Selection Method: NotRequired

Select Discourse API Operation: `put` `/admin/site_settings/email_link_color`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailLinkColor`. A value must be provided at all times for required...
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsEmailLinkColor are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.25 putAdminSiteSettingsEmailPrefix

e-mail prefix

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_prefix

Insert Discourse API URL: /admin/site_settings/email_prefix

Update Discourse API URL: /admin/site_settings/email_prefix

Delete Discourse API URL: /admin/site_settings/email_prefix

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_prefix

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEmailPrefix. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsEmailPrefix` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.26 putAdminSiteSettingsEmailSiteTitle

email site title

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_site_title

Insert Discourse API URL: /admin/site_settings/email_site_title

Update Discourse API URL: /admin/site_settings/email_site_title

Delete Discourse API URL: /admin/site_settings/email_site_title

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_site_title

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailSiteTitle`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEmailSiteTitle` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.27 putAdminSiteSettingsEmailSubject**

email subject

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/email_subject`

Insert Discourse API URL: `/admin/site_settings/email_subject`

Update Discourse API URL: `/admin/site_settings/email_subject`

Delete Discourse API URL: `/admin/site_settings/email_subject`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/email_subject`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailSubject`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEmailSubject` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.28 putAdminSiteSettingsEmailTimeWindowMins**

email time window mins

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/email_time_window_mins`

Insert Discourse API URL: `/admin/site_settings/email_time_window_mins`

Update Discourse API URL: `/admin/site_settings/email_time_window_mins`

Delete Discourse API URL: `/admin/site_settings/email_time_window_mins`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_time_window_mins

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailTimeWindowMins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Table Function Columns

The columns of the table function putAdminSiteSettingsEnableForwardedEmails are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 26.1.29 putAdminSiteSettingsEnableForwardedEmails

enable forwarded emails

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_forwarded_emails

Insert Discourse API URL: /admin/site_settings/enable_forwarded_emails

Update Discourse API URL: /admin/site_settings/enable_forwarded_emails

Delete Discourse API URL: /admin/site_settings/enable_forwarded_emails

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_forwarded_emails

## Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEnableForwardedEmails. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableForwardedEmails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 26.1.30 putAdminSiteSettingsEnableStagedUsers

enable staged users

**Catalog:** Discourse

**Schema:** SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /admin/site_settings/enable_staged_users

**Insert Discourse API URL:** /admin/site_settings/enable_staged_users

**Update Discourse API URL:** /admin/site_settings/enable_staged_users

**Delete Discourse API URL:** /admin/site_settings/enable_staged_users

**Field Selection Method:** NotRequired

**Select Discourse API Operation:**

put /admin/site_settings/enable_staged_users

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableStagedUsers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableStagedUsers` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.31 `putAdminSiteSettingsHardBounceScore`

hard bounce score

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/hard_bounce_score`

Insert Discourse API URL: `/admin/site_settings/hard_bounce_score`

Update Discourse API URL: `/admin/site_settings/hard_bounce_score`

Delete Discourse API URL: `/admin/site_settings/hard_bounce_score`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/hard_bounce_score`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsHardBounceScore`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsHardBounceScore` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.32 `putAdminSiteSettingsIgnoreByTitle`

ignore by title

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/ignore_by_title`

Insert Discourse API URL: `/admin/site_settings/ignore_by_title`

Update Discourse API URL: `/admin/site_settings/ignore_by_title`

Delete Discourse API URL: `/admin/site_settings/ignore_by_title`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/ignore_by_title`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsIgnoreByTitle`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsIgnoreByTitle` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.33 putAdminSiteSettingsIncomingEmailPreferHtml

incoming email prefer html

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/incoming_email_prefer_html

Insert Discourse API URL: /admin/site_settings/incoming_email_prefer_html

Update Discourse API

URL: /admin/site_settings/incoming_email_prefer_html

Delete Discourse API

URL: /admin/site_settings/incoming_email_prefer_html

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/incoming_email_prefer_html

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsIncomingEmailPreferHtml`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `\`select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsIncomingEmailPreferHtml` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.34 `putAdminSiteSettingsLogMailProcessingFailures`

log mail processing failures

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/log_mail_processing_failures

Insert Discourse API
URL: /admin/site_settings/log_mail_processing_failures

Update Discourse API
URL: /admin/site_settings/log_mail_processing_failures

Delete Discourse API
URL: /admin/site_settings/log_mail_processing_failures

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/log_mail_processing_failures

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLogMailProcessingFailures`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsLogMailProcessingFailures` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.35 putAdminSiteSettingsMailgunApiKey

mailgun api key

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/mailgun_api_key

Insert Discourse API URL: /admin/site_settings/mailgun_api_key

Update Discourse API URL: /admin/site_settings/mailgun_api_key

Delete Discourse API URL: /admin/site_settings/mailgun_api_key

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/mailgun_api_key

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMailgunApiKey`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsMailgunApiKey` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.36 putAdminSiteSettingsManualPollingEnabled

manual polling enabled
Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/manual_polling_enabled`
Insert Discourse API URL: `/admin/site_settings/manual_polling_enabled`
Update Discourse API URL: `/admin/site_settings/manual_polling_enabled`
Delete Discourse API URL: `/admin/site_settings/manual_polling_enabled`

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/manual_polling_enabled

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsManualPollingEnabled`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsManualPollingEnabled` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.37 putAdminSiteSettingsMaxEmailsPerDayPerUser**

max emails per day per user

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/max_emails_per_day_per_user

Insert Discourse API
URL: /admin/site_settings/max_emails_per_day_per_user

Update Discourse API
URL: /admin/site_settings/max_emails_per_day_per_user

Delete Discourse API
URL: /admin/site_settings/max_emails_per_day_per_user

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_emails_per_day_per_user

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxEmailsPerDayPerUser`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxEmailsPerDayPerUser` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.38 putAdminSiteSettingsMaximumStagedUsersPerEmail**

maximum staged users per email

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/maximum_staged_users_per_email`

Insert Discourse API
URL: `/admin/site_settings/maximum_staged_users_per_email`

Update Discourse API
URL: `/admin/site_settings/maximum_staged_users_per_email`

Delete Discourse API
URL: `/admin/site_settings/maximum_staged_users_per_email`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/maximum_staged_users_per_email

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaximumStagedUsersPerEmail`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaximumStagedUsersPerEmail` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.39 putAdminSiteSettingsPop3PollingEnabled

pop3 polling enabled

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/pop3_polling_enabled`

Insert Discourse API URL: `/admin/site_settings/pop3_polling_enabled`

Update Discourse API URL: `/admin/site_settings/pop3_polling_enabled`

Delete Discourse API URL: `/admin/site_settings/pop3_polling_enabled`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/pop3_polling_enabled

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingEnabled`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingEnabled` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.40 putAdminSiteSettingsPop3PollingHost

pop3 polling host

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/pop3_polling_host`

Insert Discourse API URL: `/admin/site_settings/pop3_polling_host`

Update Discourse API URL: `/admin/site_settings/pop3_polling_host`

Delete Discourse API URL: `/admin/site_settings/pop3_polling_host`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/pop3_polling_host`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingHost`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingHost` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.41 putAdminSiteSettingsPop3PollingOpensslVerify

pop3 polling openssl verify

Catalog: Discourse  
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API  
URL: /admin/site_settings/pop3_polling_openssl_verify

Insert Discourse API  
URL: /admin/site_settings/pop3_polling_openssl_verify

Update Discourse API  
URL: /admin/site_settings/pop3_polling_openssl_verify

Delete Discourse API  
URL: /admin/site_settings/pop3_polling_openssl_verify

Field Selection Method: NotRequired

Select Discourse API Operation:  
put /admin/site_settings/pop3_polling_openssl_verify

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsPop3PollingOpensslVerify. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsPop3PollingOpensslVerify are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.42 putAdminSiteSettingsPop3PollingPassword

pop3 polling password

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/pop3_polling_password
Insert Discourse API URL: /admin/site_settings/pop3_polling_password
Update Discourse API URL: /admin/site_settings/pop3_polling_password
Delete Discourse API URL: /admin/site_settings/pop3_polling_password

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/pop3_polling_password

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingPassword`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingPassword` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 26.1.43 putAdminSiteSettingsPop3PollingPeriodMins

#### pop3 polling period mins

**Catalog:** Discourse  
**Schema:** SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /admin/site_settings/pop3_polling_period_mins  
**Insert Discourse API URL:** /admin/site_settings/pop3_polling_period_mins  
**Update Discourse API URL:** /admin/site_settings/pop3_polling_period_mins  
**Delete Discourse API URL:** /admin/site_settings/pop3_polling_period_mins

**Field Selection Method:** NotRequired  
**Select Discourse API Operation:**  
put /admin/site_settings/pop3_polling_period_mins

## Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingPeriodMins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingPeriodMins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**26.1.44 putAdminSiteSettingsPop3PollingPort**

pop3 polling port

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/pop3_polling_port

Insert Discourse API URL: /admin/site_settings/pop3_polling_port

Update Discourse API URL: /admin/site_settings/pop3_polling_port

Delete Discourse API URL: /admin/site_settings/pop3_polling_port

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/pop3_polling_port

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingPort`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingPort` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.45 putAdminSiteSettingsPop3PollingSsl

pop3 polling ssl

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/pop3_polling_ssl`

Insert Discourse API URL: `/admin/site_settings/pop3_polling_ssl`

Update Discourse API URL: `/admin/site_settings/pop3_polling_ssl`

Delete Discourse API URL: `/admin/site_settings/pop3_polling_ssl`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/pop3_polling_ssl`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingSsl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingSsl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.46 putAdminSiteSettingsPop3PollingUsername

pop3 polling username

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/pop3_polling_username`

Insert Discourse API URL: `/admin/site_settings/pop3_polling_username`

Update Discourse API URL: `/admin/site_settings/pop3_polling_username`

Delete Discourse API URL: `/admin/site_settings/pop3_polling_username`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/pop3_polling_username`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPop3PollingUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function `putAdminSiteSettingsPop3PollingUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Parameters

26.1.47 putAdminSiteSettingsPrivateEmailTimeWindowSeconds

private email time window seconds

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/private_email_time_window_seconds

Insert Discourse API
URL: /admin/site_settings/private_email_time_window_seconds

Update Discourse API
URL: /admin/site_settings/private_email_time_window_seconds

Delete Discourse API
URL: /admin/site_settings/private_email_time_window_seconds

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/private_email_time_window_seconds
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsPrivateEmailTimeWindowSeconds. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function putAdminSiteSettingsPrivateEmailTimeWindowSeconds are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 26.1.48 putAdminSiteSettingsReplyByEmailAddress

reply by email address

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/reply_by_email_address

Insert Discourse API URL: /admin/site_settings/reply_by_email_address

Update Discourse API URL: /admin/site_settings/reply_by_email_address

Delete Discourse API URL: /admin/site_settings/reply_by_email_address

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/reply_by_email_address

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsReplyByEmailAddress`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsReplyByEmailAddress` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 26.1.49 putAdminSiteSettingsReplyByEmailEnabled

reply by email enabled

Catalog: Discourse
Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/reply_by_email_enabled`
Insert Discourse API URL: `/admin/site_settings/reply_by_email_enabled`
Update Discourse API URL: `/admin/site_settings/reply_by_email_enabled`
Delete Discourse API URL: `/admin/site_settings/reply_by_email_enabled`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/reply_by_email_enabled

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsReplyByEmailEnabled`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsReplyByEmailEnabled` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.50 `putAdminSiteSettingsResetBounceScoreAfterDays`

reset bounce score after days

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/reset_bounce_score_after_days

Insert Discourse API

URL: /admin/site_settings/reset_bounce_score_after_days

Update Discourse API

URL: /admin/site_settings/reset_bounce_score_after_days

Delete Discourse API

URL: /admin/site_settings/reset_bounce_score_after_days

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/reset_bounce_score_after_days
**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsResetBounceScoreAfterDays`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsResetBounceScoreAfterDays` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.51 putAdminSiteSettingsShortEmailLength

short email length

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/short_email_length`

Insert Discourse API URL: `/admin/site_settings/short_email_length`

Update Discourse API URL: `/admin/site_settings/short_email_length`

Delete Discourse API URL: `/admin/site_settings/short_email_length`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/short_email_length
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShortEmailLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsShortEmailLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.52 putAdminSiteSettingsShowTopicFeaturedLinkInDigest

show topic featured link in digest

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/show_topic_featured_link_in_digest

Insert Discourse API
URL: /admin/site_settings/show_topic_featured_link_in_digest

Update Discourse API
URL: /admin/site_settings/show_topic_featured_link_in_digest

Delete Discourse API
URL: /admin/site_settings/show_topic_featured_link_in_digest

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/show_topic_featured_link_in_digest

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsShowTopicFeaturedLinkInDigest. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsShowTopicFeaturedLinkInDigest are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.53 putAdminSiteSettingsSoftBounceScore

soft bounce score

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/soft_bounce_score

Insert Discourse API URL: /admin/site_settings/soft_bounce_score

Update Discourse API URL: /admin/site_settings/soft_bounce_score

Delete Discourse API URL: /admin/site_settings/soft_bounce_score

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/soft_bounce_score

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSoftBounceScore. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsSoftBounceScore are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.54 putAdminSiteSettingsStripImagesFromShortEmails

strip images from short emails

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/strip_images_from_short_emails

Insert Discourse API
URL: /admin/site_settings/strip_images_from_short_emails

Update Discourse API
URL: /admin/site_settings/strip_images_from_short_emails

Delete Discourse API
URL: /admin/site_settings/strip_images_from_short_emails
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/strip_images_from_short_emails

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsStripImagesFromShortEmails. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsStripImagesFromShortEmails are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.55 putAdminSiteSettingsUnsubscribeViaEmail

unsubscribe via email

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/unsubscribe_via_email
Insert Discourse API URL: /admin/site_settings/unsubscribe_via_email
Update Discourse API URL: /admin/site_settings/unsubscribe_via_email
Delete Discourse API URL: /admin/site_settings/unsubscribe_via_email
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/unsubscribe_via_email

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsUnsubscribeViaEmail. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsUnsubscribeViaEmail are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

26.1.56 putAdminSiteSettingsUnsubscribeViaEmailFooter

unsubscribe via email footer

Catalog: Discourse

Schema: SiteSettingsEmail

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/unsubscribe_via_email_footer

Insert Discourse API
URL: /admin/site_settings/unsubscribe_via_email_footer

Update Discourse API
URL: /admin/site_settings/unsubscribe_via_email_footer
Delete Discourse API
URL: /admin/site_settings/unsubscribe_via_email_footer

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/unsubscribe_via_email_footer

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsUnsubscribeViaEmailFooter. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsUnsubscribeViaEmailFooter are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27 Schema: SiteSettingsFiles
27.1 Tables
27.1.1 putAdminSiteSettingsAllowAllAttachmentsForGroupMessages

allow all attachments for group messages

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/allow_all_attachments_for_group_messages

Insert Discourse API
URL: /admin/site_settings/allow_all_attachments_for_group_messages

Update Discourse API
URL: /admin/site_settings/allow_all_attachments_for_group_messages

Delete Discourse API
URL: /admin/site_settings/allow_all_attachments_for_group_messages

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/allow_all_attachments_for_group_messages

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowAllAttachmentsForGroupMessages. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAllowAllAttachmentsForGroupMessages are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.2 putAdminSiteSettingsAllowAnimatedAvatars

allow animated avatars

Catalog: Discourse
Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_animated_avatars
Insert Discourse API URL: /admin/site_settings/allow_animated_avatars
Update Discourse API URL: /admin/site_settings/allow_animated_avatars
Delete Discourse API URL: /admin/site_settings/allow_animated_avatars

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/allow_animated_avatars

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowAnimatedAvatars. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAllowAnimatedAvatars are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.3 putAdminSiteSettingsAllowAnimatedThumbnails

allow animated thumbnails
Catalog: Discourse
Schema: SiteSettingsFiles
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/allow_animated_thumbnails`
Insert Discourse API URL: `/admin/site_settings/allow_animated_thumbnails`
Update Discourse API URL: `/admin/site_settings/allow_animated_thumbnails`
Delete Discourse API URL: `/admin/site_settings/allow_animated_thumbnails`

Field Selection Method: NotRequired
Select Discourse API Operation: put `/admin/site_settings/allow_animated_thumbnails`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowAnimatedThumbnails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAllowAnimatedThumbnails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>0</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.4 putAdminSiteSettingsAllowProfileBackgrounds

allow profile backgrounds
Catalog: Discourse
Schema: SiteSettingsFiles

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_profile_backgrounds
Insert Discourse API URL: /admin/site_settings/allow_profile_backgrounds
Update Discourse API URL: /admin/site_settings/allow_profile_backgrounds
Delete Discourse API URL: /admin/site_settings/allow_profile_backgrounds

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/allow_profile_backgrounds

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsAllowProfileBackgrounds. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function
putAdminSiteSettingsAllowProfileBackgrounds are shown below. Each column
has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**27.1.5 putAdminSiteSettingsAllowUploadedAvatars**

allow uploaded avatars
Catalog: Discourse
Schema: SiteSettingsFiles
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/allow_uploaded_avatars`
Insert Discourse API URL: `/admin/site_settings/allow_uploaded_avatars`
Update Discourse API URL: `/admin/site_settings/allow_uploaded_avatars`
Delete Discourse API URL: `/admin/site_settings/allow_uploaded_avatars`

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/allow_uploaded_avatars

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowUploadedAvatars`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAllowUploadedAvatars` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.6 `putAdminSiteSettingsAuthorizedExtensions`

authorized extensions
Catalog: Discourse
Schema: SiteSettingsFiles
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/authorized_extensions
Insert Discourse API URL: /admin/site_settings/authorized_extensions
Update Discourse API URL: /admin/site_settings/authorized_extensions
Delete Discourse API URL: /admin/site_settings/authorized_extensions
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/authorized_extensions

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAuthorizedExtensions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsAuthorizedExtensions are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.7 putAdminSiteSettingsAutomaticallyDownloadGravatars
automatically download gravatar
Catalog: Discourse
Schema: SiteSettingsFiles
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/automatically_download_gravatars`

Insert Discourse API
URL: `/admin/site_settings/automatically_download_gravatars`

Update Discourse API
URL: `/admin/site_settings/automatically_download_gravatars`

Delete Discourse API
URL: `/admin/site_settings/automatically_download_gravatars`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/automatically_download_gravatars

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAutomaticallyDownloadGravatars`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsAutomaticallyDownloadGravatars` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
27.1.8 putAdminSiteSettingsAvatarSizes

avatar sizes

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/avatar_sizes

Insert Discourse API URL: /admin/site_settings/avatar_sizes

Update Discourse API URL: /admin/site_settings/avatar_sizes

Delete Discourse API URL: /admin/site_settings/avatar_sizes

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/avatar_sizes

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAvatarSizes. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAvatarSizes are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
27.1.9 putAdminSiteSettingsCleanOrphanUploadsGracePeriodHours

clean orphan uploads grace period hours

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/clean_orphan_uploads_grace_period_hours

Insert Discourse API
URL: /admin/site_settings/clean_orphan_uploads_grace_period_hours

Update Discourse API
URL: /admin/site_settings/clean_orphan_uploads_grace_period_hours

Delete Discourse API
URL: /admin/site_settings/clean_orphan_uploads_grace_period_hours

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/clean_orphan_uploads_grace_period_hours

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCleanOrphanUploadsGracePeriodHours. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsCleanOrphanUploadsGracePeriodHours are shown below. Each column has an SQL data type.
27.1.10 putAdminSiteSettingsCleanUpUploads

clean up uploads

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/clean_up_uploads

Insert Discourse API URL: /admin/site_settings/clean_up_uploads

Update Discourse API URL: /admin/site_settings/clean_up_uploads

Delete Discourse API URL: /admin/site_settings/clean_up_uploads

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/clean_up_uploads

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCleanUpUploads. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsCleanUpUploads are shown below. Each column has an SQL data type.
27.1.11 putAdminSiteSettingsConvertPastedImagesQuality
convert pasted images quality
Catalog: Discourse
Schema: SiteSettingsFiles
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/convert_pasted_images_quality`

Insert Discourse API
URL: `/admin/site_settings/convert_pasted_images_quality`

Update Discourse API
URL: `/admin/site_settings/convert_pasted_images_quality`

Delete Discourse API
URL: `/admin/site_settings/convert_pasted_images_quality`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/convert_pasted_images_quality

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsConvertPastedImagesQuality`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `\`select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `\`select * from table(value1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns
The columns of the table function `putAdminSiteSettingsConvertPastedImagesToHqJpg` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**27.1.12 putAdminSiteSettingsConvertPastedImagesToHqJpg**

convert pasted images to hq jpg

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/convert_pasted_images_to_hq_jpg`

Insert Discourse API
URL: `/admin/site_settings/convert_pasted_images_to_hq_jpg`

Update Discourse API
URL: `/admin/site_settings/convert_pasted_images_to_hq_jpg`

Delete Discourse API
URL: `/admin/site_settings/convert_pasted_images_to_hq_jpg`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/convert_pasted_images_to_hq_jpg`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsConvertPastedImagesToHqJpg`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsConvertPastedImagesToHqJpg` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.13 putAdminSiteSettingsCrawlImages

crawl images

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/crawl_images

Insert Discourse API URL: /admin/site_settings/crawl_images

Update Discourse API URL: /admin/site_settings/crawl_images

Delete Discourse API URL: /admin/site_settings/crawl_images

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/crawl_images

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsCrawlImages`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsCrawlImages are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.14 putAdminSiteSettingsCreateThumbnails

create thumbnails

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/create_thumbnails

Insert Discourse API URL: /admin/site_settings/create_thumbnails

Update Discourse API URL: /admin/site_settings/create_thumbnails

Delete Discourse API URL: /admin/site_settings/create_thumbnails

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/create_thumbnails

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCreateThumbnails. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `putAdminSiteSettingsCreateThumbnails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 27.1.15 putAdminSiteSettingsDefaultAvatars

Default avatars

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/default_avatars`

Insert Discourse API URL: `/admin/site_settings/default_avatars`

Update Discourse API URL: `/admin/site_settings/default_avatars`

Delete Discourse API URL: `/admin/site_settings/default_avatars`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/default_avatars`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultAvatars`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultAvatars` are shown below. Each column has an SQL data type.
27.1.16 putAdminSiteSettingsDisabledImageDownloadDomains

disabled image download domains

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/disabled_image_download_domains

Insert Discourse API
URL: /admin/site_settings/disabled_image_download_domains

Update Discourse API
URL: /admin/site_settings/disabled_image_download_domains

Delete Discourse API
URL: /admin/site_settings/disabled_image_download_domains

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/disabled_image_download_domains

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDisabledImageDownloadDomains. A value must be provided at all
times for required parameters, but optional parameters in general do not need to have a
value and the execution will default to a pre-defined behaviour. Values can be specified by
position and by name. In both cases, all parameters not specified will be treated using their
default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsDisabledImageDownloadDomains are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.17 putAdminSiteSettingsDownloadRemoteImagesMaxDaysOld
download remote images max days old

Catalog: Discourse
Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/download_remote_images_max_days_old

Insert Discourse API
URL: /admin/site_settings/download_remote_images_max_days_old

Update Discourse API
URL: /admin/site_settings/download_remote_images_max_days_old

Delete Discourse API
URL: /admin/site_settings/download_remote_images_max_days_old

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/download_remote_images_max_days_old

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDownloadRemoteImagesMaxDaysOld. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsDownloadRemoteImagesMaxDaysOld` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

27.1.18 `putAdminSiteSettingsDownloadRemoteImagesThreshold`  
download remote images threshold  
Catalog: Discourse  
Schema: SiteSettingsFiles  
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API  
URL: /admin/site_settings/download_remote_images_threshold  

Insert Discourse API  
URL: /admin/site_settings/download_remote_images_threshold  

Update Discourse API  
URL: /admin/site_settings/download_remote_images_threshold  

Delete Discourse API  
URL: /admin/site_settings/download_remote_images_threshold  
Field Selection Method: NotRequired  
Select Discourse API Operation:  
put /admin/site_settings/download_remote_images_threshold  

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDownloadRemoteImagesThreshold`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsDownloadRemoteImagesThreshold` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.19 `putAdminSiteSettingsDownloadRemoteImagesToLocal`

download remote images to local

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/download_remote_images_to_local`

Insert Discourse API
URL: `/admin/site_settings/download_remote_images_to_local`

Update Discourse API
URL: `/admin/site_settings/download_remote_images_to_local`

Delete Discourse API
URL: `/admin/site_settings/download_remote_images_to_local`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/download_remote_images_to_local

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDownloadRemoteImagesToLocal`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDownloadRemoteImagesToLocal` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 27.1.20 putAdminSiteSettingsEnableS3Uploads

enable s3 uploads

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_s3_uploads`

Insert Discourse API URL: `/admin/site_settings/enable_s3_uploads`

Update Discourse API URL: `/admin/site_settings/enable_s3_uploads`

Delete Discourse API URL: `/admin/site_settings/enable_s3_uploads`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/enable_s3_uploads

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableS3Uploads`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableS3Uploads` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 27.1.21 putAdminSiteSettingsExternalSystemAvatarsEnabled

external system avatars enabled

Catalog: Discourse
Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/external_system_avatars_enabled

Insert Discourse API
URL: /admin/site_settings/external_system_avatars_enabled

Update Discourse API
URL: /admin/site_settings/external_system_avatars_enabled

Delete Discourse API
URL: /admin/site_settings/external_system_avatars_enabled

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/external_system_avatars_enabled

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsExternalSystemAvatarsEnabled`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and
the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsExternalSystemAvatarsEnabled` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 27.1.22 putAdminSiteSettingsExternalSystemAvatarsUrl

external system avatars url

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/external_system_avatars_url

Insert Discourse API

URL: /admin/site_settings/external_system_avatars_url

Update Discourse API

URL: /admin/site_settings/external_system_avatars_url

Delete Discourse API

URL: /admin/site_settings/external_system_avatars_url

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/external_system_avatars_url

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsExternalSystemAvatarsUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsExternalSystemAvatarsUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.23 `putAdminSiteSettingsMaxAttachmentSizeKb`

max attachment size kb

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_attachment_size_kb

Insert Discourse API URL: /admin/site_settings/max_attachment_size_kb

Update Discourse API URL: /admin/site_settings/max_attachment_size_kb

Delete Discourse API URL: /admin/site_settings/max_attachment_size_kb

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_attachment_size_kb

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxAttachmentSizeKb. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with \`select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with \`select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsMaxAttachmentSizeKb are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.24 putAdminSiteSettingsMaxImageHeight

max image height

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_image_height

Insert Discourse API URL: /admin/site_settings/max_image_height

Update Discourse API URL: /admin/site_settings/max_image_height

Delete Discourse API URL: /admin/site_settings/max_image_height

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/max_image_height

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxImageHeight. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsMaxImageHeight are shown
below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.25 putAdminSiteSettingsMaxImageMegapixels

max image megapixels

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_image_megapixels

Insert Discourse API URL: /admin/site_settings/max_image_megapixels

Update Discourse API URL: /admin/site_settings/max_image_megapixels

Delete Discourse API URL: /admin/site_settings/max_image_megapixels

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_image_megapixels

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsMaxImageMegapixels. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑️</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxImageMegapixels` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑️</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**27.1.26 putAdminSiteSettingsMaxImageSizeKb**

max image size kb

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_image_size_kb`

Insert Discourse API URL: `/admin/site_settings/max_image_size_kb`

Update Discourse API URL: `/admin/site_settings/max_image_size_kb`

Delete Discourse API URL: `/admin/site_settings/max_image_size_kb`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/max_image_size_kb

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxImageSizeKb`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with \`select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with \`select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxImageSizeKb` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 27.1.27 putAdminSiteSettingsMaxImageWidth

max image width

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_image_width`

Insert Discourse API URL: `/admin/site_settings/max_image_width`

Update Discourse API URL: `/admin/site_settings/max_image_width`

Delete Discourse API URL: `/admin/site_settings/max_image_width`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/max_image_width`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxImageWidth`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxImageWidth` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.28 `putAdminSiteSettingsPreventAnonsFromDownloadingFiles`

prevent anons from downloading files

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/prevent_anons_from_downloading_files

Insert Discourse API

URL: /admin/site_settings/prevent_anons_from_downloading_files

Update Discourse API

URL: /admin/site_settings/prevent_anons_from_downloading_files

Delete Discourse API

URL: /admin/site_settings/prevent_anons_from_downloading_files

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/prevent_anons_from_downloading_files

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPreventAnonsFromDownloadingFiles`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by
position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPreventAnonsFromDownloadingFiles` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 27.1.29 putAdminSiteSettingsPurgeDeletedUploadsGracePeriodDays

purge deleted uploads grace period days

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/purge_deleted_uploads_grace_period_days

Insert Discourse API

URL: /admin/site_settings/purge_deleted_uploads_grace_period_days

Update Discourse API

URL: /admin/site_settings/purge_deleted_uploads_grace_period_days

Delete Discourse API

URL: /admin/site_settings/purge_deleted_uploads_grace_period_days

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/purge_deleted_uploads_grace_period_days

### Table Function Parameters

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPurgeDeletedUploadsGracePeriodDays`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsPurgeDeletedUploadsGracePeriodDays` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.30 `putAdminSiteSettingsS3AccessKeyId`

`s3 access key id`

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/s3_access_key_id`

Insert Discourse API URL: `/admin/site_settings/s3_access_key_id`

Update Discourse API URL: `/admin/site_settings/s3_access_key_id`

Delete Discourse API URL: `/admin/site_settings/s3_access_key_id`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/s3_access_key_id`

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsS3AccessKeyId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsS3AccessKeyId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**27.1.31 putAdminSiteSettingsS3CdnUrl**

s3 cdn url

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/s3_cdn_url`

Insert Discourse API URL: `/admin/site_settings/s3_cdn_url`

Update Discourse API URL: `/admin/site_settings/s3_cdn_url`

Delete Discourse API URL: `/admin/site_settings/s3_cdn_url`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/s3_cdn_url`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsS3CdnUrl`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsS3CdnUrl are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**27.1.32 putAdminSiteSettingsS3Region**

s3 region

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/s3_region

Insert Discourse API URL: /admin/site_settings/s3_region

Update Discourse API URL: /admin/site_settings/s3_region

Delete Discourse API URL: /admin/site_settings/s3_region

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/s3_region

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsS3Region. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsS3Region` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.33 `putAdminSiteSettingsS3SecretAccessKey`

s3 secret access key

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/s3_secret_access_key

Insert Discourse API URL: /admin/site_settings/s3_secret_access_key

Update Discourse API URL: /admin/site_settings/s3_secret_access_key

Delete Discourse API URL: /admin/site_settings/s3_secret_access_key

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/s3_secret_access_key

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsS3SecretAccessKey`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsS3SecretAccessKey` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.34 putAdminSiteSettingsS3UploadBucket

s3 upload bucket

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/s3_upload_bucket`

Insert Discourse API URL: `/admin/site_settings/s3_upload_bucket`

Update Discourse API URL: `/admin/site_settings/s3_upload_bucket`

Delete Discourse API URL: `/admin/site_settings/s3_upload_bucket`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/s3_upload_bucket

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsS3UploadBucket`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsS3UploadBucket` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

27.1.35 `putAdminSiteSettingsS3UseIamProfile`

s3 use iam profile

Catalog: Discourse

Schema: SiteSettingsFiles

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/s3_use_iam_profile

Insert Discourse API URL: /admin/site_settings/s3_use_iam_profile

Update Discourse API URL: /admin/site_settings/s3_use_iam_profile

Delete Discourse API URL: /admin/site_settings/s3_use_iam_profile

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/s3_use_iam_profile

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsS3UseIamProfile`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsFagUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**28 Schema: SiteSettingsLegal**

**28.1 Tables**

**28.1.1 putAdminSiteSettingsFagUrl**

faq url

Catalog: Discourse

Schema: SiteSettingsLegal

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/fag_url

Insert Discourse API URL: /admin/site_settings/fag_url

Update Discourse API URL: /admin/site_settings/fag_url

Delete Discourse API URL: /admin/site_settings/fag_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/fag_url

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFagUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPagUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 28.1.2 putAdminSiteSettingsPrivacyPolicyUrl

privacy policy url

Catalog: Discourse

Schema: SiteSettingsLegal

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/privacy_policy_url

Insert Discourse API URL: /admin/site_settings/privacy_policy_url

Update Discourse API URL: /admin/site_settings/privacy_policy_url

Delete Discourse API URL: /admin/site_settings/privacy_policy_url

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/privacy_policy_url

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPrivacyPolicyUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsPrivacyPolicyUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

28.1.3 putAdminSiteSettingsTosUrl

tos url

Catalog: Discourse

Schema: SiteSettingsLegal

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/tos_url

Insert Discourse API URL: /admin/site_settings/tos_url

Update Discourse API URL: /admin/site_settings/tos_url

Delete Discourse API URL: /admin/site_settings/tos_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/tos_url

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTosUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsTosUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29 Schema: SiteSettingsLogin

29.1 Tables

29.1.1 putAdminSiteSettingsAllowNewRegistrations

Update allow new registrations

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_new_registrations

Insert Discourse API URL: /admin/site_settings/allow_new_registrations

Update Discourse API URL: /admin/site_settings/allow_new_registrations

Delete Discourse API URL: /admin/site_settings/allow_new_registrations

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/allow_new_registrations

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowNewRegistrations`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAllowNewRegistrations` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>False</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.2 putAdminSiteSettingsEmailDomainsBlacklist

email domains blacklist

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/email_domains_blacklist`

Insert Discourse API URL: `/admin/site_settings/email_domains_blacklist`

Update Discourse API URL: `/admin/site_settings/email_domains_blacklist`

Delete Discourse API URL: `/admin/site_settings/email_domains_blacklist`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/email_domains_blacklist

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailDomainsBlacklist`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEmailDomainsBlacklist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.3 putAdminSiteSettingsEmailDomainsWhitelist**

email domains whitelist

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/email_domains_whitelist`

Insert Discourse API URL: `/admin/site_settings/email_domains_whitelist`

Update Discourse API URL: `/admin/site_settings/email_domains_whitelist`

Delete Discourse API URL: `/admin/site_settings/email_domains_whitelist`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/email_domains_whitelist

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailDomainsWhitelist`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEmailDomainsWhitelist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.4 putAdminSiteSettingsEnableFacebookLogins**

enable facebook logins

**Catalog:** Discourse  
**Schema:** SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/admin/site_settings/enable_facebook_logins`  
**Insert Discourse API URL:** `/admin/site_settings/enable_facebook_logins`  
**Update Discourse API URL:** `/admin/site_settings/enable_facebook_logins`  
**Delete Discourse API URL:** `/admin/site_settings/enable_facebook_logins`  
**Field Selection Method:** NotRequired

**Select Discourse API Operation:**
`put /admin/site_settings/enable_facebook_logins`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableFacebookLogins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableFacebookLogins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.5 putAdminSiteSettingsEnableGithubLogins

enable github logins

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_github_logins`

Insert Discourse API URL: `/admin/site_settings/enable_github_logins`

Update Discourse API URL: `/admin/site_settings/enable_github_logins`

Delete Discourse API URL: `/admin/site_settings/enable_github_logins`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_github_logins

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableGithubLogins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableGithubLogins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.6 putAdminSiteSettingsEnableGoogleOauth2Logins

Update enable Google Oauth2 authentication

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/enable_google_oauth2_logins`

Insert Discourse API

URL: `/admin/site_settings/enable_google_oauth2_logins`

Update Discourse API

URL: `/admin/site_settings/enable_google_oauth2_logins`

Delete Discourse API

URL: `/admin/site_settings/enable_google_oauth2_logins`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_google_oauth2_logins

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableGoogleOauth2Logins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableGoogleOauth2Logins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.7 `putAdminSiteSettingsEnableInstagramLogins`

enable instagram logins

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_instagram_logins`

Insert Discourse API URL: `/admin/site_settings/enable_instagram_logins`

Update Discourse API URL: `/admin/site_settings/enable_instagram_logins`

Delete Discourse API URL: `/admin/site_settings/enable_instagram_logins`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_instagram_logins

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableInstagramLogins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableInstagramLogins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.8 putAdminSiteSettingsEnableLocalLogins

Update `enable_local_logins`

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_local_logins`

Insert Discourse API URL: `/admin/site_settings/enable_local_logins`

Update Discourse API URL: `/admin/site_settings/enable_local_logins`

Delete Discourse API URL: `/admin/site_settings/enable_local_logins`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/enable_local_logins`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableLocalLogins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEnableLocalLogins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.9 putAdminSiteSettingsEnableSignupCta

Update enable signup cta

Catalog: Discourse  
Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_signup_cta`

Insert Discourse API URL: `/admin/site_settings/enable_signup_cta`

Update Discourse API URL: `/admin/site_settings/enable_signup_cta`

Delete Discourse API URL: `/admin/site_settings/enable_signup_cta`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/enable_signup_cta`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableSignupCta`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableSignupCta` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.10 putAdminSiteSettingsEnableSso

enable sso

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_sso

Insert Discourse API URL: /admin/site_settings/enable_sso

Update Discourse API URL: /admin/site_settings/enable_sso

Delete Discourse API URL: /admin/site_settings/enable_sso

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/enable_sso

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableSso`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEnableSso` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.11 putAdminSiteSettingsEnableSsoProvider

enable sso provider

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_sso_provider`

Insert Discourse API URL: `/admin/site_settings/enable_sso_provider`

Update Discourse API URL: `/admin/site_settings/enable_sso_provider`

Delete Discourse API URL: `/admin/site_settings/enable_sso_provider`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/enable_sso_provider`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableSsoProvider`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEnableSsoProvider` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.12 putAdminSiteSettingsEnableTwitterLogins**

Enable twitter logins

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_twitter_logins`

Insert Discourse API URL: `/admin/site_settings/enable_twitter_logins`

Update Discourse API URL: `/admin/site_settings/enable_twitter_logins`

Delete Discourse API URL: `/admin/site_settings/enable_twitter_logins`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_twitter_logins

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableTwitterLogins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEnableTwitterLogins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.13 `putAdminSiteSettingsEnableYahooLogins`

Update enable yahoo logins

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_yahoo_logins`

Insert Discourse API URL: `/admin/site_settings/enable_yahoo_logins`

Update Discourse API URL: `/admin/site_settings/enable_yahoo_logins`

Delete Discourse API URL: `/admin/site_settings/enable_yahoo_logins`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enable_yahoo_logins

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableYahooLogins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsEnableYahooLogins are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.14 putAdminSiteSettingsFacebookAppId

facebook app id

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/facebook_app_id`

Insert Discourse API URL: `/admin/site_settings/facebook_app_id`

Update Discourse API URL: `/admin/site_settings/facebook_app_id`

Delete Discourse API URL: `/admin/site_settings/facebook_app_id`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/facebook_app_id

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsFacebookAppId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsFacebookAppId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.15 putAdminSiteSettingsFacebookAppSecret

facebook app secret

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/facebook_app_secret`

Insert Discourse API URL: `/admin/site_settings/facebook_app_secret`

Update Discourse API URL: `/admin/site_settings/facebook_app_secret`

Delete Discourse API URL: `/admin/site_settings/facebook_app_secret`

Field Selection Method: NotRequired

Select Discourse API Operation:

```
put /admin/site_settings/facebook_app_secret
```

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFacebookAppSecret`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsFacebookAppSecret` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.16 putAdminSiteSettingsFacebookRequestExtraProfileDetails**

facebook request extra profile details

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/facebook_request_extra_profile_details`

Insert Discourse API
URL: `/admin/site_settings/facebook_request_extra_profile_details`

Update Discourse API
URL: `/admin/site_settings/facebook_request_extra_profile_details`

Delete Discourse API
URL: `/admin/site_settings/facebook_request_extra_profile_details`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/facebook_request_extra_profile_details`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFacebookRequestExtraProfileDetails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsFacebookRequestExtraProfileDetails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.17 putAdminSiteSettingsForgotPasswordStrict**

forgot password strict

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/forgot_password_strict

Insert Discourse API URL: /admin/site_settings/forgot_password_strict

Update Discourse API URL: /admin/site_settings/forgot_password_strict

Delete Discourse API URL: /admin/site_settings/forgot_password_strict

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/forgot_password_strict

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsForgotPasswordStrict`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsForgotPasswordStrict` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.18 putAdminSiteSettingsGithubClientId**

github client id

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/github_client_id

Insert Discourse API URL: /admin/site_settings/github_client_id

Update Discourse API URL: /admin/site_settings/github_client_id

Delete Discourse API URL: /admin/site_settings/github_client_id

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/github_client_id

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGithubClientId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsGithubClientId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.19 putAdminSiteSettingsGithubClientSecret**

github client secret

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/github_client_secret

Insert Discourse API URL: /admin/site_settings/github_client_secret

Update Discourse API URL: /admin/site_settings/github_client_secret

Delete Discourse API URL: /admin/site_settings/github_client_secret

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/github_client_secret

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGithubClientSecret`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsGithubClientSecret` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.20 `putAdminSiteSettingsGoogleOath2ClientId`

Update google oauth2 client id

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/google_oath2_client_id`

Insert Discourse API URL: `/admin/site_settings/google_oath2_client_id`

Update Discourse API URL: `/admin/site_settings/google_oath2_client_id`

Delete Discourse API URL: `/admin/site_settings/google_oath2_client_id`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/google_oath2_client_id

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGoogleOath2ClientId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsGoogleOath2ClientId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.21 putAdminSiteSettingsGoogleOath2ClientSecret

Update google oauth2 client secret

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/google_oath2_client_secret`

Insert Discourse API URL: `/admin/site_settings/google_oath2_client_secret`

Update Discourse API URL: `/admin/site_settings/google_oath2_client_secret`

Delete Discourse API URL: `/admin/site_settings/google_oath2_client_secret`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/google_oath2_client_secret

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsGoogleOath2ClientSecret`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsGoogleOath2ClientSecret` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.22 putAdminSiteSettingsInstagramConsumerKey**

instagram consumer key

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/instagram_consumer_key`

Insert Discourse API URL: `/admin/site_settings/instagram_consumer_key`

Update Discourse API URL: `/admin/site_settings/instagram_consumer_key`

Delete Discourse API URL: `/admin/site_settings/instagram_consumer_key`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/instagram_consumer_key

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsInstagramConsumerKey`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsInstagramConsumerKey` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.23 putAdminSiteSettingsInstagramConsumerSecret**

*instagram consumer secret*

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/instagram_consumer_secret`

Insert Discourse API URL: `/admin/site_settings/instagram_consumer_secret`

Update Discourse API URL: `/admin/site_settings/instagram_consumer_secret`

Delete Discourse API URL: `/admin/site_settings/instagram_consumer_secret`

Field Selection Method: NotRequired

Select Discourse API Operation:

d1  /admin/site_settings/instagram_consumer_secret

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsInstagramConsumerSecret`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsInstagramConsumerSecret` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.24 putAdminSiteSettingsInviteOnly**

Update invite only

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/invite_only`
Insert Discourse API URL: `/admin/site_settings/invite_only`
Update Discourse API URL: `/admin/site_settings/invite_only`
Delete Discourse API URL: `/admin/site_settings/invite_only`

Field Selection Method: NotRequired
Select Discourse API Operation: put `/admin/site_settings/invite_only`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsInviteOnly`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsInviteOnly` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 29.1.25 putAdminSiteSettingsLoginRequired

Update login required

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/login_required`

Insert Discourse API URL: `/admin/site_settings/login_required`

Update Discourse API URL: `/admin/site_settings/login_required`

Delete Discourse API URL: `/admin/site_settings/login_required`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/login_required`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLoginRequired`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function putAdminSiteSettingsLoginRequired are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.26 putAdminSiteSettingsLogOutStrict

log out strict

Catalog: Discourse  
Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/log_out_strict

Insert Discourse API URL: /admin/site_settings/log_out_strict

Update Discourse API URL: /admin/site_settings/log_out_strict

Delete Discourse API URL: /admin/site_settings/log_out_strict

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/log_out_strict

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsLogOutStrict. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsLogOutStrict` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.27 putAdminSiteSettingsMaximumSessionAge**

maximum session age

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/maximum_session_age`

Insert Discourse API URL: `/admin/site_settings/maximum_session_age`

Update Discourse API URL: `/admin/site_settings/maximum_session_age`

Delete Discourse API URL: `/admin/site_settings/maximum_session_age`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/maximum_session_age`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaximumSessionAge`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaximumSessionAge` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.28 putAdminSiteSettingsMustApproveUsers**

Update must approve users

**Catalog:** Discourse

**Schema:** SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

- **Select Discourse API URL:** `/admin/site_settings/must_approve_users`
- **Insert Discourse API URL:** `/admin/site_settings/must_approve_users`
- **Update Discourse API URL:** `/admin/site_settings/must_approve_users`
- **Delete Discourse API URL:** `/admin/site_settings/must_approve_users`

**Field Selection Method:** NotRequired

**Select Discourse API Operation:**
put `/admin/site_settings/must_approve_users`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMustApproveUsers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsPendingUsersReminderDelay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.29 `putAdminSiteSettingsPendingUsersReminderDelay`

pending users reminder delay

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/pending_users_reminder_delay

Insert Discourse API
URL: /admin/site_settings/pending_users_reminder_delay

Update Discourse API
URL: /admin/site_settings/pending_users_reminder_delay

Delete Discourse API
URL: /admin/site_settings/pending_users_reminder_delay

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/pending_users_reminder_delay

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPendingUsersReminderDelay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsPendingUsersReminderDelay are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.30 putAdminSiteSettingsSsoAllowsAllReturnPaths

sso allows all return paths
Catalog: Discourse
Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/sso_allows_all_return_paths

Insert Discourse API
URL: /admin/site_settings/sso_allows_all_return_paths

Update Discourse API
URL: /admin/site_settings/sso_allows_all_return_paths

Delete Discourse API
URL: /admin/site_settings/sso_allows_all_return_paths

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/sso_allows_all_return_paths

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSsoAllowsAllReturnPaths. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSsoNotApprovedUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 29.1.31 putAdminSiteSettingsSsoNotApprovedUrl

sso not approved url

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/sso_not_approved_url`

Insert Discourse API URL: `/admin/site_settings/sso_not_approved_url`

Update Discourse API URL: `/admin/site_settings/sso_not_approved_url`

Delete Discourse API URL: `/admin/site_settings/sso_not_approved_url`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/sso_not_approved_url

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoNotApprovedUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSsoNotApprovedUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>[false]</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.32 putAdminSiteSettingsSsoOverridesAvatar

sso overrides avatar

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/sso_overrides_avatar`

Insert Discourse API URL: `/admin/site_settings/sso_overrides_avatar`

Update Discourse API URL: `/admin/site_settings/sso_overrides_avatar`

Delete Discourse API URL: `/admin/site_settings/sso_overrides_avatar`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/sso_overrides_avatar

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoOverridesAvatar`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSsoOverridesAvatar` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.33 putAdminSiteSettingsSsoOverridesBio

sso overrides bio

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/sso_overrides_bio`

Insert Discourse API URL: `/admin/site_settings/sso_overrides_bio`

Update Discourse API URL: `/admin/site_settings/sso_overrides_bio`

Delete Discourse API URL: `/admin/site_settings/sso_overrides_bio`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/sso_overrides_bio`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoOverridesBio`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSsoOverridesEmail` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.34 putAdminSiteSettingsSsoOverridesEmail**

sso overrides email

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/sso_overrides_email`

Insert Discourse API URL: `/admin/site_settings/sso_overrides_email`

Update Discourse API URL: `/admin/site_settings/sso_overrides_email`

Delete Discourse API URL: `/admin/site_settings/sso_overrides_email`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/sso_overrides_email

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoOverridesEmail`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSsoOverridesEmail` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 29.1.35 putAdminSiteSettingsSsoOverridesName

sso overrides name

catalog: Discourse

schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/sso_overrides_name`

Insert Discourse API URL: `/admin/site_settings/sso_overrides_name`

Update Discourse API URL: `/admin/site_settings/sso_overrides_name`

Delete Discourse API URL: `/admin/site_settings/sso_overrides_name`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/sso_overrides_name

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoOverridesName`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSsoOverriderName` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.36 putAdminSiteSettingsSsoOverriderUsername**

sso overrides username

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/sso_overrides_username

Insert Discourse API URL: /admin/site_settings/sso_overrides_username

Update Discourse API URL: /admin/site_settings/sso_overrides_username

Delete Discourse API URL: /admin/site_settings/sso_overrides_username

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/sso_overrides_username

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoOverriderUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSsoOverridesUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.37 putAdminSiteSettingsSsoSecret**

sso secret
Catalog: Discourse
Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/sso_secret
Insert Discourse API URL: /admin/site_settings/sso_secret
Update Discourse API URL: /admin/site_settings/sso_secret
Delete Discourse API URL: /admin/site_settings/sso_secret
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/sso_secret

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoSecret`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSsoSecret` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.38 putAdminSiteSettingsSsoUrl

sso url

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/sso_url

Insert Discourse API URL: /admin/site_settings/sso_url

Update Discourse API URL: /admin/site_settings/sso_url

Delete Discourse API URL: /admin/site_settings/sso_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/sso_url

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSsoUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSsoUrl` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

29.1.39 putAdminSiteSettingsTwitterConsumerKey

twitter consumer key

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/twitter_consumer_key

Insert Discourse API URL: /admin/site_settings/twitter_consumer_key

Update Discourse API URL: /admin/site_settings/twitter_consumer_key

Delete Discourse API URL: /admin/site_settings/twitter_consumer_key

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/twitter_consumer_key

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTwitterConsumerKey`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTwitterConsumerKey` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**29.1.40 putAdminSiteSettingsTwitterConsumerSecret**

twitter consumer secret

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/twitter_consumer_secret

Insert Discourse API URL: /admin/site_settings/twitter_consumer_secret

Update Discourse API URL: /admin/site_settings/twitter_consumer_secret

Delete Discourse API URL: /admin/site_settings/twitter_consumer_secret

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/twitter_consumer_secret

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTwitterConsumerSecret`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTwitterConsumerSecret` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 29.1.41 putAdminSiteSettingsVerboseSsoLogging

verbose sso logging

Catalog: Discourse

Schema: SiteSettingsLogin

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/verbose_sso_logging`

Insert Discourse API URL: `/admin/site_settings/verbose_sso_logging`

Update Discourse API URL: `/admin/site_settings/verbose_sso_logging`

Delete Discourse API URL: `/admin/site_settings/verbose_sso_logging`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/verbose_sso_logging

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsVerboseSsoLogging`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsVerboseSsoLogging` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
</table>
| RESULT   | string    | Result| ☑️       | Outcome of operation as single
|          |           |       |          | plain text column.                                |

### 30 Schema: SiteSettingsNarrativeBot

#### 30.1 Tables

**30.1.1 putAdminSiteSettingsDisableDiscourseNarrativeBotWelcomePost**

disable discourse narrative bot welcome post

Catalog: Discourse

Schema: SiteSettingsNarrativeBot

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/disable_discourse_narrative_bot_welcome_post

Insert Discourse API

URL: /admin/site_settings/disable_discourse_narrative_bot_welcome_post

Update Discourse API

URL: /admin/site_settings/disable_discourse_narrative_bot_welcome_post

Delete Discourse API

URL: /admin/site_settings/disable_discourse_narrative_bot_welcome_post

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/disable_discourse_narrative_bot_welcome_post

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDisableDiscourseNarrativeBotWelcomePost. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDisableDiscourseNarrativeBotWelcomePost are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

30.1.2 putAdminSiteSettingsDiscourseNarrativeBotDisablePublicReplies

discourse narrative bot disable public replies

Catalog: Discourse

Schema: SiteSettingsNarrativeBot

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/discourse_narrative_bot_disable_public_replies

Insert Discourse API

URL: /admin/site_settings/discourse_narrative_bot_disable_public_replies

Update Discourse API

URL: /admin/site_settings/discourse_narrative_bot_disable_public_replies
Delete Discourse API
URL: /admin/site_settings/discourse_narrative_bot_disable_public_replies

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/discourse_narrative_bot_disable_public_replies

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDiscourseNarrativeBotDisablePublicReplies. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDiscourseNarrativeBotDisablePublicReplies are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

30.1.3 putAdminSiteSettingsDiscourseNarrativeBotEnabled

enable discourse narrative bot

Catalog: Discourse
Schema: SiteSettingsNarrativeBot

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/discourse_narrative_bot_enabled

Insert Discourse API
URL: /admin/site_settings/discourse_narrative_bot_enabled

Update Discourse API
URL: /admin/site_settings/discourse_narrative_bot_enabled

Delete Discourse API
URL: /admin/site_settings/discourse_narrative_bot_enabled

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/discourse_narrative_bot_enabled

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDiscourseNarrativeBotEnabled. A value must be provided at all times
for required parameters, but optional parameters in general do not need to have a value and
the execution will default to a pre-defined behaviour. Values can be specified by position and
by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function
putAdminSiteSettingsDiscourseNarrativeBotEnabled are shown below. Each
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

30.1.4 putAdminSiteSettingsDiscourseNarrativeBotIgnoredUsernames

discourse narrative bot ignored usernames

Catalog: Discourse

Schema: SiteSettingsNarrativeBot

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/discourse_narrative_bot_ignored_usernames`

Insert Discourse API
URL: `/admin/site_settings/discourse_narrative_bot_ignored_usernames`

Update Discourse API
URL: `/admin/site_settings/discourse_narrative_bot_ignored_usernames`

Delete Discourse API
URL: `/admin/site_settings/discourse_narrative_bot_ignored_usernames`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/discourse_narrative_bot_ignored_usernames`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDiscourseNarrativeBotIgnoredUsernames`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsDiscourseNarrativeBotIgnoredUsernames` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
30.1.5 putAdminSiteSettingsDiscourseNarrativeBotWelcomePostDelay

discourse narrative bot welcome post delay

Catalog: Discourse

Schema: SiteSettingsNarrativeBot

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/discourse_narrative_bot_welcome_post_delay

Insert Discourse API

URL: /admin/site_settings/discourse_narrative_bot_welcome_post_delay

Update Discourse API

URL: /admin/site_settings/discourse_narrative_bot_welcome_post_delay

Delete Discourse API

URL: /admin/site_settings/discourse_narrative_bot_welcome_post_delay

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/discourse_narrative_bot_welcome_post_delay

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDiscourseNarrativeBotWelcomePostDelay. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The columns of the table function `putAdminSiteSettingsDiscourseNarrativeBotWelcomePostDelay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**30.1.6 putAdminSiteSettingsDiscourseNarrativeBotWelcomePostType**

discourse narrative bot welcome post type

Catalog: Discourse

Schema: SiteSettingsNarrativeBot

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/discourse_narrative_bot_welcome_post_type`

Insert Discourse API

URL: `/admin/site_settings/discourse_narrative_bot_welcome_post_type`

Update Discourse API

URL: `/admin/site_settings/discourse_narrative_bot_welcome_post_type`

Delete Discourse API

URL: `/admin/site_settings/discourse_narrative_bot_welcome_post_type`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/discourse_narrative_bot_welcome_post_type

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDiscourseNarrativeBotWelcomePostType`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsDiscourseNarrativeBotWelcomePostType` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>❌</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

31 Schema: SiteSettingsOnebox

31.1 Tables

31.1.1 putAdminSiteSettingsEnableFlashVideoOnebox

enable flash video onebox

Catalog: Discourse

Schema: SiteSettingsOnebox

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_flash_video_onebox`

Insert Discourse API URL: `/admin/site_settings/enable_flash_video_onebox`

Update Discourse API URL: `/admin/site_settings/enable_flash_video_onebox`

Delete Discourse API URL: `/admin/site_settings/enable_flash_video_onebox`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/enable_flash_video_onebox`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableFlashVideoOnebox`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableFlashVideoOnebox` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

31.1.2 `putAdminSiteSettingsMaxOneboxesPerPost`

`max oneboxes per post`

Catalog: Discourse

Schema: SiteSettingsOnebox

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the `table NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_oneboxes_per_post`

Insert Discourse API URL: `/admin/site_settings/max_oneboxes_per_post`

Update Discourse API URL: `/admin/site_settings/max_oneboxes_per_post`

Delete Discourse API URL: `/admin/site_settings/max_oneboxes_per_post`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/max_oneboxes_per_post`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxOneboxesPerPost`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxOneboxesPerPost` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**31.1.3 putAdminSiteSettingsOneboxDomainsBlacklist**

onebox domains blacklist

Catalog: Discourse

Schema: SiteSettingsOnebox

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/onebox_domains_blacklist`

Insert Discourse API URL: `/admin/site_settings/onebox_domains_blacklist`

Update Discourse API URL: `/admin/site_settings/onebox_domains_blacklist`

Delete Discourse API URL: `/admin/site_settings/onebox_domains_blacklist`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/onebox_domains_blacklist`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsOneboxDomainsBlacklist`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
**Table Function Columns**

The columns of the table function `putAdminSiteSettingsOneboxDomainsBlacklist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**31.1.4 putAdminSiteSettingsPostOneboxMaxlength**

post onebox maxlength

Catalog: Discourse

Schema: SiteSettingsOnebox

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/post_onebox_maxlength`

Insert Discourse API URL: `/admin/site_settings/post_onebox_maxlength`

Update Discourse API URL: `/admin/site_settings/post_onebox_maxlength`

Delete Discourse API URL: `/admin/site_settings/post_onebox_maxlength`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/post_onebox_maxlength`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPostOneboxMaxlength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsPostOneboxMaxlength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

32 Schema: SiteSettingsOther

32.1 Tables

32.1.1 putAdminSiteSettingsAutomaticallyUnpinTopics

automatically unpin topics

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/automatically_unpin_topics

Insert Discourse API URL: /admin/site_settings/automatically_unpin_topics

Update Discourse API
URL: /admin/site_settings/automatically_unpin_topics

Delete Discourse API
URL: /admin/site_settings/automatically_unpin_topics

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/automatically_unpin_topics

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAutomaticallyUnpinTopics`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAutomaticallyUnpinTopics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.2 putAdminSiteSettingsBootstrapModeMinUsers

**bootstrap mode min users**

**Catalog:** Discourse  
**Schema:** SiteSettingsOther  

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/admin/site_settings/bootstrap_mode_min_users`  
**Insert Discourse API URL:** `/admin/site_settings/bootstrap_mode_min_users`  
**Update Discourse API URL:** `/admin/site_settings/bootstrap_mode_min_users`  
**Delete Discourse API URL:** `/admin/site_settings/bootstrap_mode_min_users`  

**Field Selection Method:** NotRequired  
**Select Discourse API Operation:** put `/admin/site_settings/bootstrap_mode_min_users`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsBootstrapModeMinUsers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsBootstrapModeMinUsers` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.3 `putAdminSiteSettingsColdAgeDaysHigh`

cold age days high

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/cold_age_days_high`

Insert Discourse API URL: `/admin/site_settings/cold_age_days_high`

Update Discourse API URL: `/admin/site_settings/cold_age_days_high`

Delete Discourse API URL: `/admin/site_settings/cold_age_days_high`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/cold_age_days_high`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsColdAgeDaysHigh`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsColdAgeDaysHigh are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.4 putAdminSiteSettingsColdAgeDaysLow

cold age days low

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/cold_age_days_low

Insert Discourse API URL: /admin/site_settings/cold_age_days_low

Update Discourse API URL: /admin/site_settings/cold_age_days_low

Delete Discourse API URL: /admin/site_settings/cold_age_days_low

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/cold_age_days_low

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsColdAgeDaysLow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsColdAgeDaysLow` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.5 putAdminSiteSettingsColdAgeDaysMedium

cold age days medium

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/cold_age_days_medium`

Insert Discourse API URL: `/admin/site_settings/cold_age_days_medium`

Update Discourse API URL: `/admin/site_settings/cold_age_days_medium`

Delete Discourse API URL: `/admin/site_settings/cold_age_days_medium`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/cold_age_days_medium`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsColdAgeDaysMedium`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsColdAgeDaysMedium` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.6 putAdminSiteSettingsDeleteDraftsOlderThanNDays**

delete drafts older than n days

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**

URL: `/admin/site_settings/delete_drafts_older_than_n_days`

**Insert Discourse API**

URL: `/admin/site_settings/delete_drafts_older_than_n_days`

**Update Discourse API**

URL: `/admin/site_settings/delete_drafts_older_than_n_days`

**Delete Discourse API**

URL: `/admin/site_settings/delete_drafts_older_than_n_days`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/delete_drafts_older_than_n_days

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDeleteDraftsOlderThanNDays`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDeleteDraftsOlderThanNDays` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.7 putAdminSiteSettingsDisableAvatarEducationMessage

disable avatar education message

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/disable_avatar_education_message

Insert Discourse API
URL: /admin/site_settings/disable_avatar_education_message

Update Discourse API
URL: /admin/site_settings/disable_avatar_education_message

Delete Discourse API
URL: /admin/site_settings/disable_avatar_education_message

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/disable_avatar_education_message

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDisableAvatarEducationMessage`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDisableAvatarEducationMessage` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.8 putAdminSiteSettingsDisableEditNotifications

disable edit notifications

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/disable_edit_notifications

Insert Discourse API URL: /admin/site_settings/disable_edit_notifications

Update Discourse API URL: /admin/site_settings/disable_edit_notifications

Delete Discourse API URL: /admin/site_settings/disable_edit_notifications

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/disable_edit_notifications

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDisableEditNotifications`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDisableEditNotifications` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.9 putAdminSiteSettingsDominatingTopicMinimumPercent

dominating topic minimum percent

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/dominating_topic_minimum_percent

Insert Discourse API
URL: /admin/site_settings/dominating_topic_minimum_percent

Update Discourse API
URL: /admin/site_settings/dominating_topic_minimum_percent

Delete Discourse API
URL: /admin/site_settings/dominating_topic_minimum_percent

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/dominating_topic_minimum_percent

### Table Function Parameters

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDominatingTopicMinimumPercent. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDominatingTopicMinimumPercent` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 32.1.10 putAdminSiteSettingsEducateUntilPosts

educate until posts

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/educate_until_posts

Insert Discourse API URL: /admin/site_settings/educate_until_posts

Update Discourse API URL: /admin/site_settings/educate_until_posts

Delete Discourse API URL: /admin/site_settings/educate_until_posts

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/educate_until_posts
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEducateUntilPosts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsEducateUntilPosts are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.11 putAdminSiteSettingsGlobalNotice

global notice

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/global_notice

Insert Discourse API URL: /admin/site_settings/global_notice

Update Discourse API URL: /admin/site_settings/global_notice

Delete Discourse API URL: /admin/site_settings/global_notice

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/global_notice

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsGlobalNotice. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function putAdminSiteSettingsGlobalNotice are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.12 putAdminSiteSettingsHistoryHoursHigh

history hours high

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/history_hours_high

Insert Discourse API URL: /admin/site_settings/history_hours_high

Update Discourse API URL: /admin/site_settings/history_hours_high

Delete Discourse API URL: /admin/site_settings/history_hours_high

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/history_hours_high

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsHistoryHoursHigh. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsHistoryHoursHigh` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
</tr>
</tbody>
</table>

**32.1.13 putAdminSiteSettingsHistoryHoursLow**

history hours low

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/history_hours_low

Insert Discourse API URL: /admin/site_settings/history_hours_low

Update Discourse API URL: /admin/site_settings/history_hours_low

Delete Discourse API URL: /admin/site_settings/history_hours_low

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/history_hours_low

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsHistoryHoursLow`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsHistoryHoursLow` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.14 putAdminSiteSettingsHistoryHoursMedium

history hours medium

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/history_hours_medium

Insert Discourse API URL: /admin/site_settings/history_hours_medium

Update Discourse API URL: /admin/site_settings/history_hours_medium

Delete Discourse API URL: /admin/site_settings/history_hours_medium

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/history_hours_medium

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsHistoryHoursMedium`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsHistoryHoursMedium` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.15 putAdminSiteSettingsMaxSimilarResults**

max similar results
Catalog: Discourse
Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_similar_results
Insert Discourse API URL: /admin/site_settings/max_similar_results
Update Discourse API URL: /admin/site_settings/max_similar_results
Delete Discourse API URL: /admin/site_settings/max_similar_results

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/max_similar_results

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxSimilarResults`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>false</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxSimilarResults` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>false</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.16 putAdminSiteSettingsMinimumTopicsSimilar

minimum topics similar

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/minimum_topics_similar`

Insert Discourse API URL: `/admin/site_settings/minimum_topics_similar`

Update Discourse API URL: `/admin/site_settings/minimum_topics_similar`

Delete Discourse API URL: `/admin/site_settings/minimum_topics_similar`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/minimum_topics_similar

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinimumTopicsSimilar`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsMinimumTopicsSimilar are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.17 putAdminSiteSettingsNativeAppInstallBanner

"native app install banner"

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/native_app_install_banner

Insert Discourse API URL: /admin/site_settings/native_app_install_banner

Update Discourse API URL: /admin/site_settings/native_app_install_banner

Delete Discourse API URL: /admin/site_settings/native_app_install_banner

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/native_app_install_banner

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsNativeAppInstallBanner. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsNativeAppInstallBanner` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.18 putAdminSiteSettingsNewVersionEmails

new version emails

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/new_version_emails`

Insert Discourse API URL: `/admin/site_settings/new_version_emails`

Update Discourse API URL: `/admin/site_settings/new_version_emails`

Delete Discourse API URL: `/admin/site_settings/new_version_emails`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/new_version_emails

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNewVersionEmails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsNewVersionEmails are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.19 putAdminSiteSettingsNotifyAboutFlagsAfter**

notify about flags after

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/notify_about_flags_after

Insert Discourse API URL: /admin/site_settings/notify_about_flags_after

Update Discourse API URL: /admin/site_settings/notify_about_flags_after

Delete Discourse API URL: /admin/site_settings/notify_about_flags_after

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/notify_about_flags_after

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsNotifyAboutFlagsAfter. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsNotifyAboutFlagsAfter are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.20 putAdminSiteSettingsPermalinkNormalizations

permalink normalizations

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/permalink_normalizations
Insert Discourse API URL: /admin/site_settings/permalink_normalizations
Update Discourse API URL: /admin/site_settings/permalink_normalizations
Delete Discourse API URL: /admin/site_settings/permalink_normalizations

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/permalink_normalizations

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsPermalinkNormalizations. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsPermalinkNormalizations` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.21 `putAdminSiteSettingsPreviousVisitTimeoutHours`

previous visit timeout hours

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/previous_visit_timeout_hours

Insert Discourse API
URL: /admin/site_settings/previous_visit_timeout_hours

Update Discourse API
URL: /admin/site_settings/previous_visit_timeout_hours

Delete Discourse API
URL: /admin/site_settings/previous_visit_timeout_hours

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/previous_visit_timeout_hours

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPreviousVisitTimeoutHours`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>false</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPreviousVisitTimeoutHours` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>false</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.22 `putAdminSiteSettingsReadTimeWordCount`

read time word count

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/read_time_word_count

Insert Discourse API URL: /admin/site_settings/read_time_word_count

Update Discourse API URL: /admin/site_settings/read_time_word_count

Delete Discourse API URL: /admin/site_settings/read_time_word_count

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/read_time_word_count

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsReadTimeWordCount`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsReadTimeWordCount` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.23 putAdminSiteSettingsSendWelcomeMessage**

send welcome message

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/send_welcome_message

Insert Discourse API URL: /admin/site_settings/send_welcome_message

Update Discourse API URL: /admin/site_settings/send_welcome_message

Delete Discourse API URL: /admin/site_settings/send_welcome_message

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/send_welcome_message

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSendWelcomeMessage`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSendWelcomeMessage` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 32.1.24 putAdminSiteSettingsSequentialRepliesThreshold

sequential replies threshold

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/sequential_replies_threshold

Insert Discourse API

URL: /admin/site_settings/sequential_replies_threshold

Update Discourse API

URL: /admin/site_settings/sequential_replies_threshold

Delete Discourse API

URL: /admin/site_settings/sequential_replies_threshold

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/sequential_replies_threshold

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSequentialRepliesThreshold`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSequentialRepliesThreshold` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 32.1.25 putAdminSiteSettingsShowCreateTopicsNotice

show create topics notice

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/show_create_topics_notice`

Insert Discourse API URL: `/admin/site_settings/show_create_topics_notice`

Update Discourse API URL: `/admin/site_settings/show_create_topics_notice`

Delete Discourse API URL: `/admin/site_settings/show_create_topics_notice`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/show_create_topics_notice`

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShowCreateTopicsNotice`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsShowCreateTopicsNotice` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.26 putAdminSiteSettingsSlugGenerationMethod

slug generation method

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/slug_generation_method`

Insert Discourse API URL: `/admin/site_settings/slug_generation_method`

Update Discourse API URL: `/admin/site_settings/slug_generation_method`

Delete Discourse API URL: `/admin/site_settings/slug_generation_method`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/slug_generation_method`

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSlugGenerationMethod`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSlugGenerationMethod` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.27 putAdminSiteSettingsStaffLikeWeight

staff like weight

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/staff_like_weight

Insert Discourse API URL: /admin/site_settings/staff_like_weight

Update Discourse API URL: /admin/site_settings/staff_like_weight

Delete Discourse API URL: /admin/site_settings/staff_like_weight

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/staff_like_weight

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsStaffLikeWeight`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsStaffLikeWeight` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.28 putAdminSiteSettingsSummaryLikesRequired

summary likes required  
Catalog: Discourse  
Schema: SiteSettingsOther  

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/summary_likes_required  
Insert Discourse API URL: /admin/site_settings/summary_likes_required  
Update Discourse API URL: /admin/site_settings/summary_likes_required  
Delete Discourse API URL: /admin/site_settings/summary_likes_required  
Field Selection Method: NotRequired  
Select Discourse API Operation:  
put /admin/site_settings/summary_likes_required

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSummaryLikesRequired`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the...
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSummaryLikesRequired` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.29 putAdminSiteSettingsSummaryMaxResults

summary max results

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/summary_max_results`

Insert Discourse API URL: `/admin/site_settings/summary_max_results`

Update Discourse API URL: `/admin/site_settings/summary_max_results`

Delete Discourse API URL: `/admin/site_settings/summary_max_results`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/summary_max_results

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSummaryMaxResults`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsSummaryMaxResults are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.30 putAdminSiteSettingsSummaryPercentFilter

summary percent filter

Catalog: Discourse
Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/summary_percent_filter
Insert Discourse API URL: /admin/site_settings/summary_percent_filter
Update Discourse API URL: /admin/site_settings/summary_percent_filter
Delete Discourse API URL: /admin/site_settings/summary_percent_filter

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/summary_percent_filter

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSummaryPercentFilter. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSummaryPercentFilter` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.31 putAdminSiteSettingsSummaryPostsRequired**

summary posts required
Catalog: Discourse
Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/summary_posts_required`
Insert Discourse API URL: `/admin/site_settings/summary_posts_required`
Update Discourse API URL: `/admin/site_settings/summary_posts_required`
Delete Discourse API URL: `/admin/site_settings/summary_posts_required`
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/summary_posts_required

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSummaryPostsRequired`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function `putAdminSiteSettingsSummaryPostsRequired` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

32.1.32 `putAdminSiteSettingsSummaryScoreThreshold`

summary score threshold

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/summary_score_threshold`

Insert Discourse API URL: `/admin/site_settings/summary_score_threshold`

Update Discourse API URL: `/admin/site_settings/summary_score_threshold`

Delete Discourse API URL: `/admin/site_settings/summary_score_threshold`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/summary_score_threshold

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSummaryScoreThreshold`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsSummaryScoreThreshold` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 32.1.33 putAdminSiteSettingsSuppressUncategorizedBadge

**suppress uncategorized badge**

**Catalog:** Discourse  
**Schema:** SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**  
URL: /admin/site_settings/suppress_uncategorized_badge

**Insert Discourse API**  
URL: /admin/site_settings/suppress_uncategorized_badge

**Update Discourse API**  
URL: /admin/site_settings/suppress_uncategorized_badge

**Delete Discourse API**  
URL: /admin/site_settings/suppress_uncategorized_badge

Field Selection Method: NotRequired  
Select Discourse API Operation:  
put /admin/site_settings/suppress_uncategorized_badge

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSuppressUncategorizedBadge`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsSuppressUncategorizedBadge` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 32.1.34 putAdminSiteSettingsTopicPageTitleIncludesCategory

topic page title includes category

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/topic_page_title_includes_category`

Insert Discourse API

URL: `/admin/site_settings/topic_page_title_includes_category`

Update Discourse API

URL: `/admin/site_settings/topic_page_title_includes_category`

Delete Discourse API

URL: `/admin/site_settings/topic_page_title_includes_category`

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/topic_page_title_includes_category

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicPageTitleIncludesCategory. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTopicPageTitleIncludesCategory are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.35 putAdminSiteSettingsTopicPostLikeHeatHigh

topic post like heat high

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/topic_post_like_heat_high

Insert Discourse API URL: /admin/site_settings/topic_post_like_heat_high

Update Discourse API URL: /admin/site_settings/topic_post_like_heat_high

Delete Discourse API URL: /admin/site_settings/topic_post_like_heat_high

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/topic_post_like_heat_high

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTopicPostLikeHeatHigh`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTopicPostLikeHeatHigh` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.36 putAdminSiteSettingsTopicPostLikeHeatLow**

topic post like heat low

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/topic_post_like_heat_low
Insert Discourse API URL: /admin/site_settings/topic_post_like_heat_low
Update Discourse API URL: /admin/site_settings/topic_post_like_heat_low
Delete Discourse API URL: /admin/site_settings/topic_post_like_heat_low
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/topic_post_like_heat_low

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicPostLikeHeatLow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsTopicPostLikeHeatLow are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**32.1.37 putAdminSiteSettingsTopicPostLikeHeatMedium**

topic post like heat medium

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/topic_post_like_heat_medium

Insert Discourse API
URL: /admin/site_settings/topic_post_like_heat_medium

Update Discourse API
URL: /admin/site_settings/topic_post_like_heat_medium
Delete Discourse API
URL: /admin/site_settings/topic_post_like_heat_medium
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/topic_post_like_heat_medium

Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicPostLikeHeatMedium. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsTopicPostLikeHeatMedium are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.38 putAdminSiteSettingsTopicViewDurationHours

Topic view duration hours
Catalog: Discourse
Schema: SiteSettingsOther
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/topic_view_duration_hours
Insert Discourse API URL: /admin/site_settings/topic_view_duration_hours
Update Discourse API URL: /admin/site_settings/topic_view_duration_hours
Delete Discourse API URL: /admin/site_settings/topic_view_duration_hours

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/topic_view_duration_hours

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicViewDurationHours. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTopicViewDurationHours are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.39 putAdminSiteSettingsTopicViewsHeatHigh

topic views heat high

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/topic_views_heat_high

Insert Discourse API URL: /admin/site_settings/topic_views_heat_high

Update Discourse API URL: /admin/site_settings/topic_views_heat_high

Delete Discourse API URL: /admin/site_settings/topic_views_heat_high

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/topic_views_heat_high

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicViewsHeatHigh. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTopicViewsHeatHigh are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.40 putAdminSiteSettingsTopicViewsHeatLow

topic views heat low

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/topic_views_heat_low
Insert Discourse API URL: /admin/site_settings/topic_views_heat_low
Update Discourse API URL: /admin/site_settings/topic_views_heat_low
Delete Discourse API URL: /admin/site_settings/topic_views_heat_low

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/topic_views_heat_low

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicViewsHeatLow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTopicViewsHeatLow are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.41 putAdminSiteSettingsTopicViewsHeatMedium

topic views heat medium

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/topic_views_heat_medium

Insert Discourse API URL: /admin/site_settings/topic_views_heat_medium

Update Discourse API URL: /admin/site_settings/topic_views_heat_medium

Delete Discourse API URL: /admin/site_settings/topic_views_heat_medium

Field Selection Method: NotRequired

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Select Discourse API Operation:
put /admin/site_settings/topic_views_heat_medium

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicViewsHeatMedium. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with \`select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with \`select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTopicViewsHeatMedium are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.42 putAdminSiteSettingsUserProfileViewDurationHours

user profile view duration hours

Catalog: Discourse

Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/user_profile_view_duration_hours

Insert Discourse API
URL: /admin/site_settings/user_profile_view_duration_hours

Update Discourse API
URL: /admin/site_settings/user_profile_view_duration_hours
Delete Discourse API
URL: /admin/site_settings/user_profile_view_duration_hours

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/user_profile_view_duration_hours

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsUserProfileViewDurationHours. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsUserProfileViewDurationHours are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.43 putAdminSiteSettingsVacuumDbDays

vacuum db days
Catalog: Discourse
Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/vacuum_db_days
Insert Discourse API URL: /admin/site_settings/vacuum_db_days
Update Discourse API URL: /admin/site_settings/vacuum_db_days
Delete Discourse API URL: /admin/site_settings/vacuum_db_days
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/vacuum_db_days

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsVacuumDbDays. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsVacuumDbDays are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

32.1.44 putAdminSiteSettingsVersionChecks

version checks
Catalog: Discourse
Schema: SiteSettingsOther

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/version_checks
Insert Discourse API URL: /admin/site_settings/version_checks
Update Discourse API URL: /admin/site_settings/version_checks
Delete Discourse API URL: /admin/site_settings/version_checks
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/version_checks

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsVersionChecks. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsVersionChecks are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

33 Schema: SiteSettingsPlugins

33.1 Tables

33.1.1 putAdminSiteSettingsDailyPerformanceReport

"daily performance report"

Catalog: Discourse

Schema: SiteSettingsPlugins

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/daily_performance_report
Insert Discourse API URL: /admin/site_settings/daily_performance_report
Update Discourse API URL: /admin/site_settings/daily_performance_report
Delete Discourse API URL: /admin/site_settings/daily_performance_report
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/daily_performance_report

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDailyPerformanceReport`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDailyPerformanceReport` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

33.1.2 putAdminSiteSettingsDetailsEnabled

"details enabled"

Catalog: Discourse

Schema: SiteSettingsPlugins

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/details_enabled

Insert Discourse API URL: /admin/site_settings/details_enabled

Update Discourse API URL: /admin/site_settings/details_enabled

Delete Discourse API URL: /admin/site_settings/details_enabled

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/details_enabled

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDetailsEnabled. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsDetailsEnabled are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

33.1.3 putAdminSiteSettingsPollEditWindowMins

poll edit window mins

Catalog: Discourse

Schema: SiteSettingsPlugins

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/poll_edit_window_mins

Insert Discourse API URL: /admin/site_settings/poll_edit_window_mins

Update Discourse API URL: /admin/site_settings/poll_edit_window_mins

Delete Discourse API URL: /admin/site_settings/poll_edit_window_mins

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/poll_edit_window_mins
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPollEditWindowMins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsPollEditWindowMins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

33.1.4 putAdminSiteSettingsPollEnabled

"poll enabled"

Catalog: Discourse

Schema: SiteSettingsPlugins

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/poll_enabled

Insert Discourse API URL: /admin/site_settings/poll_enabled

Update Discourse API URL: /admin/site_settings/poll_enabled

Delete Discourse API URL: /admin/site_settings/poll_enabled

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/poll_enabled

Table Function Parameters
The following parameters can be used to control the behavior of the table function `putAdminSiteSettingsPollEnabled`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behavior. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsPollEnabled` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 33.1.5 putAdminSiteSettingsPollMaximumOptions

"poll maximum options"

Catalog: Discourse

Schema: SiteSettingsPlugins

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/poll_maximum_options`

Insert Discourse API URL: `/admin/site_settings/poll_maximum_options`

Update Discourse API URL: `/admin/site_settings/poll_maximum_options`

Delete Discourse API URL: `/admin/site_settings/poll_maximum_options`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/poll_maximum_options

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPollMaximumOptions`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `\texttt{\select* from table(value1, value2, value3)}` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `\texttt{\select* from table(name1 \Rightarrow value1, name3 \Rightarrow value3)}` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPollMaximumOptions` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34 Schema: SiteSettingsPosting

34.1 Tables

34.1.1 `putAdminSiteSettingsAllowDuplicateTopicTitles`

allow duplicate topic titles

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/allow_duplicate_topic_titles

Insert Discourse API

URL: /admin/site_settings/allow_duplicate_topic_titles

Update Discourse API

URL: /admin/site_settings/allow_duplicate_topic_titles

Delete Discourse API

URL: /admin/site_settings/allow_duplicate_topic_titles

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/allow_duplicate_topic_titles

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowDuplicateTopicTitles. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsAllowDuplicateTopicTitles are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.2 putAdminSiteSettingsAllowedHrefSchemes

allowed href schemes

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allowed_href_schemes

Insert Discourse API URL: /admin/site_settings/allowed_href_schemes

Update Discourse API URL: /admin/site_settings/allowed_href_schemes

Delete Discourse API URL: /admin/site_settings/allowed_href_schemes

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/allowed_href_schemes

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsAllowedHrefSchemes. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAllowedHrefSchemes are
shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.3 putAdminSiteSettingsAllowHtmlTables

allow html tables

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_html_tables
Insert Discourse API URL: /admin/site_settings/allow_html_tables
Update Discourse API URL: /admin/site_settings/allow_html_tables
Delete Discourse API URL: /admin/site_settings/allow_html_tables
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/allow_html_tables
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowHtmlTables`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsAllowHtmlTables` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.4 putAdminSiteSettingsAllowUncategorizedTopics

allow uncategorized topics

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_uncategorized_topics

Insert Discourse API URL: /admin/site_settings/allow_uncategorized_topics

Update Discourse API URL: /admin/site_settings/allow_uncategorized_topics

Delete Discourse API URL: /admin/site_settings/allow_uncategorized_topics

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/allow_uncategorized_topics

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowUncategorizedTopics. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsAllowUncategorizedTopics are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 34.1.5 putAdminSiteSettingsAllowUppercasePosts

allow_uppercase_posts

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/allow_uppercase_posts
Insert Discourse API URL: /admin/site_settings/allow_uppercase_posts
Update Discourse API URL: /admin/site_settings/allow_uppercase_posts
Delete Discourse API URL: /admin/site_settings/allow_uppercase_posts
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/allow_uppercase_posts

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowUppercasePosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAllowUppercasePosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.6 `putAdminSiteSettingsApproveNewTopicsUnlessTrustLevel`

approve new topics unless trust level

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/approve_new_topics_unless_trust_level

Insert Discourse API

URL: /admin/site_settings/approve_new_topics_unless_trust_level

Update Discourse API

URL: /admin/site_settings/approve_new_topics_unless_trust_level
Delete Discourse API
URL: /admin/site_settings/approve_new_topics_unless_trust_level
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/approve_new_topics_unless_trust_level

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsApproveNewTopicsUnlessTrustLevel. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsApproveNewTopicsUnlessTrustLevel are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.7 putAdminSiteSettingsApprovePostCount

approve post count
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/approve_post_count
Insert Discourse API URL: /admin/site_settings/approve_post_count
Update Discourse API URL: /admin/site_settings/approve_post_count
Delete Discourse API URL: /admin/site_settings/approve_post_count
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/approve_post_count

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsApprovePostCount`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsApprovePostCount` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.8 `putAdminSiteSettingsApproveUnlessTrustLevel`

approve unless trust level

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/approve_unless_trust_level

Insert Discourse API URL: /admin/site_settings/approve_unless_trust_level
Update Discourse API
URL: /admin/site_settings/approve_unless_trust_level

Delete Discourse API
URL: /admin/site_settings/approve_unless_trust_level

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/approve_unless_trust_level

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsApproveUnlessTrustLevel`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsApproveUnlessTrustLevel` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.9 putAdminSiteSettingsAutoCloseMessagesPostCount

auto close messages post count

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/auto_close_messages_post_count

Insert Discourse API
URL: /admin/site_settings/auto_close_messages_post_count

Update Discourse API
URL: /admin/site_settings/auto_close_messages_post_count

Delete Discourse API
URL: /admin/site_settings/auto_close_messages_post_count

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/auto_close_messages_post_count

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutoCloseMessagesPostCount. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAutoCloseMessagesPostCount are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.10 putAdminSiteSettingsAutoCloseTopicsPostCount
auto close topics post count
Catalog: Discourse
Schema: SiteSettingsPosting
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/auto_close_topics_post_count
Insert Discourse API URL: /admin/site_settings/auto_close_topics_post_count
Update Discourse API URL: /admin/site_settings/auto_close_topics_post_count
Delete Discourse API URL: /admin/site_settings/auto_close_topics_post_count

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/auto_close_topics_post_count

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutoCloseTopicsPostCount. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAutoCloseTopicsPostCount are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.11 putAdminSiteSettingsAutohighlightAllCode

autohighlight all code

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/autohighlight_all_code

Insert Discourse API URL: /admin/site_settings/autohighlight_all_code

Update Discourse API URL: /admin/site_settings/autohighlight_all_code

Delete Discourse API URL: /admin/site_settings/autohighlight_all_code

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/autohighlight_all_code

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutohighlightAllCode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsAutohighlightAllCode are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.12 putAdminSiteSettingsBodyMinEntropy

body min entropy

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/body_min_entropy

Insert Discourse API URL: /admin/site_settings/body_min_entropy

Update Discourse API URL: /admin/site_settings/body_min_entropy

Delete Discourse API URL: /admin/site_settings/body_min_entropy

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/body_min_entropy

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsBodyMinEntropy. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsBodyMinEntropy are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.13 putAdminSiteSettingsCensoredPattern
censored pattern
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/censored_pattern
Insert Discourse API URL: /admin/site_settings/censored_pattern
Update Discourse API URL: /admin/site_settings/censored_pattern
Delete Discourse API URL: /admin/site_settings/censored_pattern
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/censored_pattern

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCensoredPattern. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsCensoredPattern are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.14 putAdminSiteSettingsCensoredWords

censored words

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/censored_words

Insert Discourse API URL: /admin/site_settings/censored_words

Update Discourse API URL: /admin/site_settings/censored_words

Delete Discourse API URL: /admin/site_settings/censored_words

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/censored_words

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCensoredWords. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsCensoredWords are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.15 putAdminSiteSettingsCodeFormattingStyle

code formatting style

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/code_formatting_style`

Insert Discourse API URL: `/admin/site_settings/code_formatting_style`

Update Discourse API URL: `/admin/site_settings/code_formatting_style`

Delete Discourse API URL: `/admin/site_settings/code_formatting_style`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/code_formatting_style

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsCodeFormattingStyle`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsCodeFormattingStyle` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.16 putAdminSiteSettingsDefaultCodeLang

default code lang

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/default_code_lang

Insert Discourse API URL: /admin/site_settings/default_code_lang

Update Discourse API URL: /admin/site_settings/default_code_lang

Delete Discourse API URL: /admin/site_settings/default_code_lang

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/default_code_lang

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultCodeLang. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultCodeLang are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.17 putAdminSiteSettingsDeleteOldHiddenPosts

delete old hidden posts

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/delete_old_hidden_posts
Insert Discourse API URL: /admin/site_settings/delete_old_hidden_posts
Update Discourse API URL: /admin/site_settings/delete_old_hidden_posts
Delete Discourse API URL: /admin/site_settings/delete_old_hidden_posts

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/delete_old_hidden_posts

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDeleteOldHiddenPosts. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDeleteOldHiddenPosts
are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.18 putAdminSiteSettingsDeleteRemovedPostsAfter

delete removed posts after

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/delete_removed_posts_after

Insert Discourse API URL: /admin/site_settings/delete_removed_posts_after

Update Discourse API
URL: /admin/site_settings/delete_removed_posts_after

Delete Discourse API
URL: /admin/site_settings/delete_removed_posts_after

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/delete_removed_posts_after

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDeleteRemovedPostsAfter. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function
putAdminSiteSettingsDeleteRemovedPostsAfter are shown below. Each column
has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.19 putAdminSiteSettingsDisplayNameOnPosts
display name on posts
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/display_name_on_posts
Insert Discourse API URL: /admin/site_settings/display_name_on_posts
Update Discourse API URL: /admin/site_settings/display_name_on_posts
Delete Discourse API URL: /admin/site_settings/display_name_on_posts

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/display_name_on_posts

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDisplayNameOnPosts. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDisplayNameOnPosts are
shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.20 putAdminSiteSettingsEditHistoryVisibleToPublic
edit history visible to public
Catalog: Discourse
Schema: SiteSettingsPosting
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/edit_history_visible_to_public
Insert Discourse API
URL: /admin/site_settings/edit_history_visible_to_public
Update Discourse API
URL: /admin/site_settings/edit_history_visible_to_public
Delete Discourse API
URL: /admin/site_settings/edit_history_visible_to_public
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/edit_history_visible_to_public

Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEditHistoryVisibleToPublic. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsEditHistoryVisibleToPublic are shown below. Each column has an SQL data type.
### 34.1.21 putAdminSiteSettingsEditingGracePeriod

**editing grace period**

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/editing_grace_period`

Insert Discourse API URL: `/admin/site_settings/editing_grace_period`

Update Discourse API URL: `/admin/site_settings/editing_grace_period`

Delete Discourse API URL: `/admin/site_settings/editing_grace_period`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/editing_grace_period`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEditingGracePeriod`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEditingGracePeriod` are shown below. Each column has an SQL data type.
34.1.22 putAdminSiteSettingsEmbedTruncate

embed truncate
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/embed_truncate
Insert Discourse API URL: /admin/site_settings/embed_truncate
Update Discourse API URL: /admin/site_settings/embed_truncate
Delete Discourse API URL: /admin/site_settings/embed_truncate

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/embed_truncate

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEmbedTruncate. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsEmbedTruncate are shown below. Each column has an SQL data type.
### 34.1.23 putAdminSiteSettingsEmojiSet

**emoji set**

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/admin/site_settings/emoji_set`

**Insert Discourse API URL:** `/admin/site_settings/emoji_set`

**Update Discourse API URL:** `/admin/site_settings/emoji_set`

**Delete Discourse API URL:** `/admin/site_settings/emoji_set`

Field Selection Method: NotRequired

**Select Discourse API Operation:** `put /admin/site_settings/emoji_set`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmojiSet`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEmojiSet` are shown below. Each column has an SQL data type.
34.1.24 putAdminSiteSettingsEnableEmoji

enable emoji

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_emoji

Insert Discourse API URL: /admin/site_settings/enable_emoji

Update Discourse API URL: /admin/site_settings/enable_emoji

Delete Discourse API URL: /admin/site_settings/enable_emoji

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/enable_emoji

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEnableEmoji. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsEnableEmoji are shown below. Each column has an SQL data type.
### 34.1.25 `putAdminSiteSettingsEnablePrivateMessages`

**enable private messages**

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_private_messages`

Insert Discourse API URL: `/admin/site_settings/enable_private_messages`

Update Discourse API URL: `/admin/site_settings/enable_private_messages`

Delete Discourse API URL: `/admin/site_settings/enable_private_messages`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/enable_private_messages`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnablePrivateMessages`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnablePrivateMessages` are shown below. Each column has an SQL data type.
34.1.26 putAdminSiteSettingsEnforceSquareEmoji

enforce square emoji

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enforce_square_emoji

Insert Discourse API URL: /admin/site_settings/enforce_square_emoji

Update Discourse API URL: /admin/site_settings/enforce_square_emoji

Delete Discourse API URL: /admin/site_settings/enforce_square_emoji

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/enforce_square_emoji

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEnforceSquareEmoji. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

**Table Function Columns**

The columns of the table function putAdminSiteSettingsEnforceSquareEmoji are shown below. Each column has an SQL data type.
34.1.27 putAdminSiteSettingsHighlightedLanguages

highlighted languages

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/highlighted_languages

Insert Discourse API URL: /admin/site_settings/highlighted_languages

Update Discourse API URL: /admin/site_settings/highlighted_languages

Delete Discourse API URL: /admin/site_settings/highlighted_languages

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/highlighted_languages

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsHighlightedLanguages. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function putAdminSiteSettingsHighlightedLanguages are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
34.1.28 putAdminSiteSettingsMaxMentionsPerPost

max mentions per post
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_mentions_per_post
Insert Discourse API URL: /admin/site_settings/max_mentions_per_post
Update Discourse API URL: /admin/site_settings/max_mentions_per_post
Delete Discourse API URL: /admin/site_settings/max_mentions_per_post
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/max_mentions_per_post

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxMentionsPerPost. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsMaxMentionsPerPost are shown below. Each column has an SQL data type.
34.1.29 putAdminSiteSettingsMaxPostLength

max post length
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_post_length
Insert Discourse API URL: /admin/site_settings/max_post_length
Update Discourse API URL: /admin/site_settings/max_post_length
Delete Discourse API URL: /admin/site_settings/max_post_length
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/max_post_length

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxPostLength. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsMaxPostLength are shown below. Each column has an SQL data type.
34.1.30 putAdminSiteSettingsMaxReplyHistory

max reply history

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_reply_history

Insert Discourse API URL: /admin/site_settings/max_reply_history

Update Discourse API URL: /admin/site_settings/max_reply_history

Delete Discourse API URL: /admin/site_settings/max_reply_history

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/max_reply_history

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxReplyHistory. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsMaxReplyHistory are shown below. Each column has an SQL data type.
34.1.31 putAdminSiteSettingsMaxTopicTitleLength

max topic title length
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_topic_title_length
Insert Discourse API URL: /admin/site_settings/max_topic_title_length
Update Discourse API URL: /admin/site_settings/max_topic_title_length
Delete Discourse API URL: /admin/site_settings/max_topic_title_length

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/max_topic_title_length

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxTopicTitleLength. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsMaxTopicTitleLength are shown below. Each column has an SQL data type.
34.1.32 putAdminSiteSettingsMaxUsersNotifiedPerGroupMention

max users notified per group mention

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/max_users_notified_per_group_mention

Insert Discourse API
URL: /admin/site_settings/max_users_notified_per_group_mention

Update Discourse API
URL: /admin/site_settings/max_users_notified_per_group_mention

Delete Discourse API
URL: /admin/site_settings/max_users_notified_per_group_mention

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_users_notified_per_group_mention

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxUsersNotifiedPerGroupMention. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsMaxUsersNotifiedPerGroupMention are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 34.1.33 putAdminSiteSettingsMinBodyTitleLength

**min body title length**

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_body_title_length`

Insert Discourse API URL: `/admin/site_settings/min_body_title_length`

Update Discourse API URL: `/admin/site_settings/min_body_title_length`

Delete Discourse API URL: `/admin/site_settings/min_body_title_length`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/min_body_title_length`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMinBodyTitleLength. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsMinBodyTitleLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.34 `putAdminSiteSettingsMinFirstPostLength`

min first post length

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_first_post_length`

Insert Discourse API URL: `/admin/site_settings/min_first_post_length`

Update Discourse API URL: `/admin/site_settings/min_first_post_length`

Delete Discourse API URL: `/admin/site_settings/min_first_post_length`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/min_first_post_length`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinFirstPostLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table Function Columns

The columns of the table function `putAdminSiteSettingsMinFirstPostLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 34.1.35 putAdminSiteSettingsMinPostLength

min post length

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_post_length`

Insert Discourse API URL: `/admin/site_settings/min_post_length`

Update Discourse API URL: `/admin/site_settings/min_post_length`

Delete Discourse API URL: `/admin/site_settings/min_post_length`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/min_post_length`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinPostLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Table Function Columns
The columns of the table function `putAdminSiteSettingsMinPostLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 34.1.36 putAdminSiteSettingsMinPrivateMessagePostLength

min private message post length

Catalog: Discourse  
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/min_private_message_post_length`

Insert Discourse API
URL: `/admin/site_settings/min_private_message_post_length`

Update Discourse API
URL: `/admin/site_settings/min_private_message_post_length`

Delete Discourse API
URL: `/admin/site_settings/min_private_message_post_length`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/min_private_message_post_length`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinPrivateMessagePostLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsMinPrivateMessagePostLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.37 `putAdminSiteSettingsMinPrivateMessageTitleLength`

min private message title length

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/min_private_message_title_length`

Insert Discourse API
URL: `/admin/site_settings/min_private_message_title_length`

Update Discourse API
URL: `/admin/site_settings/min_private_message_title_length`

Delete Discourse API
URL: `/admin/site_settings/min_private_message_title_length`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/min_private_message_title_length

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinPrivateMessageTitleLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsMinPrivateMessageTitleLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

34.1.38 putAdminSiteSettingsMinTitleSimilarLength

`min_title_similar_length`

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_title_similar_length`

Insert Discourse API URL: `/admin/site_settings/min_title_similar_length`

Update Discourse API URL: `/admin/site_settings/min_title_similar_length`

Delete Discourse API URL: `/admin/site_settings/min_title_similar_length`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/min_title_similar_length`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinTitleSimilarLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function putAdminSiteSettingsMinTitleSimilarLength are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

34.1.39 putAdminSiteSettingsMinTopicTitleLength

min topic title length
Catalog: Discourse
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/min_topic_title_length
Insert Discourse API URL: /admin/site_settings/min_topic_title_length
Update Discourse API URL: /admin/site_settings/min_topic_title_length
Delete Discourse API URL: /admin/site_settings/min_topic_title_length
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/min_topic_title_length

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMinTopicTitleLength. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
### Table Function Columns

The columns of the table function `putAdminSiteSettingsMinTopicTitleLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 34.1.40 putAdminSiteSettingsNewuserMaxAttachments

newuser max attachments

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/newuser_max_attachments`

Insert Discourse API URL: `/admin/site_settings/newuser_max_attachments`

Update Discourse API URL: `/admin/site_settings/newuser_max_attachments`

Delete Discourse API URL: `/admin/site_settings/newuser_max_attachments`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/newuser_max_attachments`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNewuserMaxAttachments`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsNewuserMaxAttachments` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 34.1.41 `putAdminSiteSettingsNewuserMaxImages`

**newuser max images**

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/admin/site_settings/newuser_max_images`

**Insert Discourse API URL:** `/admin/site_settings/newuser_max_images`

**Update Discourse API URL:** `/admin/site_settings/newuser_max_images`

**Delete Discourse API URL:** `/admin/site_settings/newuser_max_images`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/newuser_max_images`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNewuserMaxImages`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
**Table Function Columns**

The columns of the table function `putAdminSiteSettingsNewuserMaxImages` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**34.1.42 putAdminSiteSettingsNewuserMaxLinks**

newuser max links

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/newuser_max_links`

Insert Discourse API URL: `/admin/site_settings/newuser_max_links`

Update Discourse API URL: `/admin/site_settings/newuser_max_links`

Delete Discourse API URL: `/admin/site_settings/newuser_max_links`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/newuser_max_links`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNewuserMaxLinks`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsNewuserMaxLinks are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.43 putAdminSiteSettingsNewuserMaxMentionsPerPost

newuser max mentions per post

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/newuser_max_mentions_per_post

Insert Discourse API
URL: /admin/site_settings/newuser_max_mentions_per_post

Update Discourse API
URL: /admin/site_settings/newuser_max_mentions_per_post

Delete Discourse API
URL: /admin/site_settings/newuser_max_mentions_per_post

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/newuser_max_mentions_per_post

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsNewuserMaxMentionsPerPost. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsNewuserMaxMentionsPerPost` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

34.1.44 `putAdminSiteSettingsNewuserMaxRepliesPerTopic`

catmax replies per topic

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/newuser_max_replies_per_topic

Insert Discourse API

URL: /admin/site_settings/newuser_max_replies_per_topic

Update Discourse API

URL: /admin/site_settings/newuser_max_replies_per_topic

Delete Discourse API

URL: /admin/site_settings/newuser_max_replies_per_topic

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/newuser_max_replies_per_topic

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNewuserMaxRepliesPerTopic`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsNewuserMaxRepliesPerTopic` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 34.1.45 putAdminSiteSettingsNotifyAboutQueuedPostsAfter

**notify about queued posts after**

**Catalog:** Discourse  
**Schema:** SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**
**URL:** `/admin/site_settings/notify_about_queued_posts_after`

**Insert Discourse API**
**URL:** `/admin/site_settings/notify_about_queued_posts_after`

**Update Discourse API**
**URL:** `/admin/site_settings/notify_about_queued_posts_after`

**Delete Discourse API**
**URL:** `/admin/site_settings/notify_about_queued_posts_after`

**Field Selection Method:** NotRequired

**Select Discourse API Operation:**
`put /admin/site_settings/notify_about_queued_posts_after`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNotifyAboutQueuedPostsAfter`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsNotifyAboutQueuedPostsAfter` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.46 **putAdminSiteSettingsPostEditTimeLimit**

post edit time limit

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/post_edit_time_limit`

Insert Discourse API URL: `/admin/site_settings/post_edit_time_limit`

Update Discourse API URL: `/admin/site_settings/post_edit_time_limit`

Delete Discourse API URL: `/admin/site_settings/post_edit_time_limit`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/post_edit_time_limit

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPostEditTimeLimit`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsPostEditTimeLimit` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.47 putAdminSiteSettingsPostExcerptMaxlength

post excerpt maxlength

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/post_excerpt_maxlength`

Insert Discourse API URL: `/admin/site_settings/post_excerpt_maxlength`

Update Discourse API URL: `/admin/site_settings/post_excerpt_maxlength`

Delete Discourse API URL: `/admin/site_settings/post_excerpt_maxlength`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/post_excerpt_maxlength

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPostExcerptMaxlength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsPostExcerptMaxlength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 34.1.48 putAdminSiteSettingsPostUndoActionWindowMins

post undo action window mins

Catalog: Discourse  
Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**

URL: `/admin/site_settings/post_undo_action_window_mins`

**Insert Discourse API**

URL: `/admin/site_settings/post_undo_action_window_mins`

**Update Discourse API**

URL: `/admin/site_settings/post_undo_action_window_mins`

**Delete Discourse API**

URL: `/admin/site_settings/post_undo_action_window_mins`

Field Selection Method: NotRequired

**Select Discourse API Operation:**

put /admin/site_settings/post_undo_action_window_mins

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPostUndoActionWindowMins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsPostUndoActionWindowMins are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**34.1.49 putAdminSiteSettingsShortProgressTextThreshold**

short progress text threshold

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/short_progress_text_threshold

Insert Discourse API

URL: /admin/site_settings/short_progress_text_threshold

Update Discourse API

URL: /admin/site_settings/short_progress_text_threshold

Delete Discourse API

URL: /admin/site_settings/short_progress_text_threshold

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/short_progress_text_threshold

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShortProgressTextThreshold`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsShortProgressTextThreshold` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.50 putAdminSiteSettingsShowPinnedExcerptDesktop

show pinned excerpt desktop

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/show_pinned_excerpt_desktop

Insert Discourse API
URL: /admin/site_settings/show_pinned_excerpt_desktop

Update Discourse API
URL: /admin/site_settings/show_pinned_excerpt_desktop

Delete Discourse API
URL: /admin/site_settings/show_pinned_excerpt_desktop

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/show_pinned_excerpt_desktop

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShowPinnedExcerptDesktop`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsShowPinnedExcerptDesktop` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.51 putAdminSiteSettingsShowPinnedExcerptMobile

show pinned excerpt mobile

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/show_pinned_excerpt_mobile

Insert Discourse API URL: /admin/site_settings/show_pinned_excerpt_mobile

Update Discourse API
URL: /admin/site_settings/show_pinned_excerpt_mobile
Delete Discourse API
URL: /admin/site_settings/show_pinned_excerpt_mobile

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/show_pinned_excerpt_mobile

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShowPinnedExcerptMobile`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsShowPinnedExcerptMobile` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**34.1.52 putAdminSiteSettingsShowTimeGapDays**

show time gap days

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/show_time_gap_days

Insert Discourse API URL: /admin/site_settings/show_time_gap_days

Update Discourse API URL: /admin/site_settings/show_time_gap_days
Delete Discourse API URL: /admin/site_settings/show_time_gap_days

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/show_time_gap_days

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsShowTimeGapDays. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsShowTimeGapDays are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.53 putAdminSiteSettingsSuppressReplyDirectlyAbove

suppress reply directly above

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/suppress_reply_directly_above

Insert Discourse API
URL: /admin/site_settings/suppress_reply_directly_above

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Update Discourse API
URL: /admin/site_settings/suppress_reply_directly_above

Delete Discourse API
URL: /admin/site_settings/suppress_reply_directly_above

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/suppress_reply_directly_above

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSuppressReplyDirectlyAbove. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>❌</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsSuppressReplyDirectlyAbove are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>❌</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**34.1.54 putAdminSiteSettingsSuppressReplyDirectlyBelow**

suppress reply directly below

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/suppress_reply_directly_below

Insert Discourse API
URL: /admin/site_settings/suppress_reply_directly_below

Update Discourse API
URL: /admin/site_settings/suppress_reply_directly_below

Delete Discourse API
URL: /admin/site_settings/suppress_reply_directly_below

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/suppress_reply_directly_below

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSuppressReplyDirectlyBelow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsSuppressReplyDirectlyBelow are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

34.1.55 putAdminSiteSettingsSuppressReplyWhenQuoting

suppress reply when quoting

Catalog: Discourse

Schema: SiteSettingsPosting
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/suppress_reply_when_quoting

Insert Discourse API
URL: /admin/site_settings/suppress_reply_when_quoting

Update Discourse API
URL: /admin/site_settings/suppress_reply_when_quoting

Delete Discourse API
URL: /admin/site_settings/suppress_reply_when_quoting

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/suppress_reply_when_quoting

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSuppressReplyWhenQuoting. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsSuppressReplyWhenQuoting are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.56 putAdminSiteSettingsTitleFancyEntities

title fancy entities

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/title_fancy_entities

Insert Discourse API URL: /admin/site_settings/title_fancy_entities

Update Discourse API URL: /admin/site_settings/title_fancy_entities

Delete Discourse API URL: /admin/site_settings/title_fancy_entities

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/title_fancy_entities

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTitleFancyEntities. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsTitleFancyEntities are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.57 putAdminSiteSettingsTitleMaxWordLength

title max word length

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/title_max_word_length

Insert Discourse API URL: /admin/site_settings/title_max_word_length

Update Discourse API URL: /admin/site_settings/title_max_word_length

Delete Discourse API URL: /admin/site_settings/title_max_word_length

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/title_max_word_length

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTitleMaxWordLength. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTitleMaxWordLength are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.58 putAdminSiteSettingsTitleMinEntropy

title min entropy

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/title_min_entropy

Insert Discourse API URL: /admin/site_settings/title_min_entropy

Update Discourse API URL: /admin/site_settings/title_min_entropy

Delete Discourse API URL: /admin/site_settings/title_min_entropy

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/title_min_entropy

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsTitleMinEntropy. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTitleMinEntropy are
shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.59 putAdminSiteSettingsTitlePrettify

title_prettify

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/title_prettify

Insert Discourse API URL: /admin/site_settings/title_prettify

Update Discourse API URL: /admin/site_settings/title_prettify

Delete Discourse API URL: /admin/site_settings/title_prettify

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/title_prettify

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTitlePrettify. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTitlePrettify are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
34.1.60 putAdminSiteSettingsTopicFeaturedLinkEnabled

**topic featured link enabled**

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/topic_featured_link_enabled

Insert Discourse API
URL: /admin/site_settings/topic_featured_link_enabled

Update Discourse API
URL: /admin/site_settings/topic_featured_link_enabled

Delete Discourse API
URL: /admin/site_settings/topic_featured_link_enabled

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/topic_featured_link_enabled

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTopicFeaturedLinkEnabled. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsTopicFeaturedLinkEnabled are shown below. Each column has an SQL data type.
34.1.61 putAdminSiteSettingsTraditionalMarkdownLinebreaks

traditional markdown linebreaks

Catalog: Discourse

Schema: SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/traditional_markdown_linebreaks

Insert Discourse API
URL: /admin/site_settings/traditional_markdown_linebreaks

Update Discourse API
URL: /admin/site_settings/traditional_markdown_linebreaks

Delete Discourse API
URL: /admin/site_settings/traditional_markdown_linebreaks

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/traditional_markdown_linebreaks

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTraditionalMarkdownLinebreaks. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The columns of the table function `putAdminSiteSettingsWarnRevivingOldTopicAge` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 34.1.62 putAdminSiteSettingsWarnRevivingOldTopicAge

**warn reviving old topic age**

**Catalog:** Discourse  
**Schema:** SiteSettingsPosting

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**
**URL:** /admin/site_settings/warn_reviving_old_topic_age

**Insert Discourse API**
**URL:** /admin/site_settings/warn_reviving_old_topic_age

**Update Discourse API**
**URL:** /admin/site_settings/warn_reviving_old_topic_age

**Delete Discourse API**
**URL:** /admin/site_settings/warn_reviving_old_topic_age

**Field Selection Method:** NotRequired

**Select Discourse API Operation:**
`put /admin/site_settings/warn_reviving_old_topic_age`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsWarnRevivingOldTopicAge`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsWarnRevivingOldTopicAge` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35 Schema: SiteSettingsRateLimits

35.1 Tables

35.1.1 putAdminSiteSettingsAlertAdminsIfErrorsPerHour

alert admins if errors per hour

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/alert_admins_if_errors_per_hour

Insert Discourse API

URL: /admin/site_settings/alert_admins_if_errors_per_hour

Update Discourse API

URL: /admin/site_settings/alert_admins_if_errors_per_hour

Delete Discourse API

URL: /admin/site_settings/alert_admins_if_errors_per_hour

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/alert_admins_if_errors_per_hour

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAlertAdminsIfErrorsPerHour`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAlertAdminsIfErrorsPerHour` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 35.1.2 putAdminSiteSettingsAlertAdminsIfErrorsPerMinute

alert admins if errors per minute

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/alert_admins_if_errors_per_minute`

Insert Discourse API

URL: `/admin/site_settings/alert_admins_if_errors_per_minute`

Update Discourse API

URL: `/admin/site_settings/alert_admins_if_errors_per_minute`

Delete Discourse API

URL: `/admin/site_settings/alert_admins_if_errors_per_minute`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/alert_admins_if_errors_per_minute

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAlertAdminsIfErrorsPerMinute`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAlertAdminsIfErrorsPerMinute` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35.1.3 `putAdminSiteSettingsMaxBookmarksPerDay`

max bookmarks per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_bookmarks_per_day

Insert Discourse API URL: /admin/site_settings/max_bookmarks_per_day

Update Discourse API URL: /admin/site_settings/max_bookmarks_per_day

Delete Discourse API URL: /admin/site_settings/max_bookmarks_per_day

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_bookmarks_per_day

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxBookmarksPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxBookmarksPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 35.1.4 putAdminSiteSettingsMaxEditsPerDay

max edits per day
Catalog: Discourse

| Schema: SiteSettingsRateLimits |
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_edits_per_day`
Insert Discourse API URL: `/admin/site_settings/max_edits_per_day`
Update Discourse API URL: `/admin/site_settings/max_edits_per_day`
Delete Discourse API URL: `/admin/site_settings/max_edits_per_day`
Field Selection Method: NotRequired

Select Discourse API Operation: **put /admin/site_settings/max_edits_per_day**

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxEditsPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxEditsPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 35.1.5 putAdminSiteSettingsMaxFlagsPerDay

max flags per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_flags_per_day`

Insert Discourse API URL: `/admin/site_settings/max_flags_per_day`

Update Discourse API URL: `/admin/site_settings/max_flags_per_day`

Delete Discourse API URL: `/admin/site_settings/max_flags_per_day`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/max_flags_per_day`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxFlagsPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxFlagsPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**35.1.6 putAdminSiteSettingsMaxInvitesPerDay**

max invites per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_invites_per_day`

Insert Discourse API URL: `/admin/site_settings/max_invites_per_day`

Update Discourse API URL: `/admin/site_settings/max_invites_per_day`

Delete Discourse API URL: `/admin/site_settings/max_invites_per_day`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_invites_per_day

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxInvitesPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxInvitesPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**35.1.7 putAdminSiteSettingsMaxLikesPerDay**

max likes per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_likes_per_day`

Insert Discourse API URL: `/admin/site_settings/max_likes_per_day`

Update Discourse API URL: `/admin/site_settings/max_likes_per_day`

Delete Discourse API URL: `/admin/site_settings/max_likes_per_day`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/max_likes_per_day

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxLikesPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxLikesPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**35.1.8 putAdminSiteSettingsMaxPrintsPerHourPerUser**

max prints per hour per user

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/max_prints_per_hour_per_user

Insert Discourse API

URL: /admin/site_settings/max_prints_per_hour_per_user

Update Discourse API

URL: /admin/site_settings/max_prints_per_hour_per_user

Delete Discourse API

URL: /admin/site_settings/max_prints_per_hour_per_user

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/max_prints_per_hour_per_user

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxPrintsPerHourPerUser`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxPrintsPerHourPerUser` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑️</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 35.1.9 putAdminSiteSettingsMaxPrivateMessagesPerDay

max private messages per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/max_private_messages_per_day

Insert Discourse API

URL: /admin/site_settings/max_private_messages_per_day

Update Discourse API

URL: /admin/site_settings/max_private_messages_per_day

Delete Discourse API

URL: /admin/site_settings/max_private_messages_per_day

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/max_private_messages_per_day

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxPrivateMessagesPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxPrivateMessagesPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑️</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 35.1.10 putAdminSiteSettingsMaxRepliesInFirstDay

**max replies in first day**

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_replies_in_first_day`

Insert Discourse API URL: `/admin/site_settings/max_replies_in_first_day`

Update Discourse API URL: `/admin/site_settings/max_replies_in_first_day`

Delete Discourse API URL: `/admin/site_settings/max_replies_in_first_day`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_replies_in_first_day

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxRepliesInFirstDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxRepliesInFirstDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

## 35.1.11 `putAdminSiteSettingsMaxTopicInvitationsPerDay`

max topic invitations per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/max_topic_invitations_per_day

Insert Discourse API

URL: /admin/site_settings/max_topic_invitations_per_day

Update Discourse API

URL: /admin/site_settings/max_topic_invitations_per_day

Delete Discourse API

URL: /admin/site_settings/max_topic_invitations_per_day

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/max_topic_invitations_per_day

# Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxTopicInvitationsPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxTopicInvitationsPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35.1.12 putAdminSiteSettingsMaxTopicsInFirstDay

max topics in first day
Catalog: Discourse
Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_topics_in_first_day
Insert Discourse API URL: /admin/site_settings/max_topics_in_first_day
Update Discourse API URL: /admin/site_settings/max_topics_in_first_day
Delete Discourse API URL: /admin/site_settings/max_topics_in_first_day
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_topics_in_first_day

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxTopicsInFirstDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxTopicsInFirstDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**35.1.13 putAdminSiteSettingsMaxTopicsPerDay**

max topics per day

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_topics_per_day`

Insert Discourse API URL: `/admin/site_settings/max_topics_per_day`

Update Discourse API URL: `/admin/site_settings/max_topics_per_day`

Delete Discourse API URL: `/admin/site_settings/max_topics_per_day`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/max_topics_per_day

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxTopicsPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxTopicsPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**35.1.14 putAdminSiteSettingsRateLimitCreatePost**

rate limit create post

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/rate_limit_create_post

Insert Discourse API URL: /admin/site_settings/rate_limit_create_post

Update Discourse API URL: /admin/site_settings/rate_limit_create_post

Delete Discourse API URL: /admin/site_settings/rate_limit_create_post

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/rate_limit_create_post

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsRateLimitCreatePost`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsRateLimitCreatePost` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35.1.15 putAdminSiteSettingsRateLimitCreateTopic
rate limit create topic
Catalog: Discourse
Schema: SiteSettingsRateLimits
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/rate_limit_create_topic
Insert Discourse API URL: /admin/site_settings/rate_limit_create_topic
Update Discourse API URL: /admin/site_settings/rate_limit_create_topic
Delete Discourse API URL: /admin/site_settings/rate_limit_create_topic
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/rate_limit_create_topic

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function
`putAdminSiteSettingsRateLimitCreateTopic`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsRateLimitCreateTopic` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**35.1.16 putAdminSiteSettingsRateLimitNewUserCreatePost**

crate limit new user create post

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/rate_limit_new_user_create_post

Insert Discourse API

URL: /admin/site_settings/rate_limit_new_user_create_post

Update Discourse API

URL: /admin/site_settings/rate_limit_new_user_create_post

Delete Discourse API

URL: /admin/site_settings/rate_limit_new_user_create_post

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/rate_limit_new_user_create_post

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsRateLimitNewUserCreatePost`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsRateLimitNewUserCreatePost` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 35.1.17 putAdminSiteSettingsRateLimitNewUserCreateTopic

rate limit new user create topic

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/rate_limit_new_user_create_topic`

Insert Discourse API

URL: `/admin/site_settings/rate_limit_new_user_create_topic`

Update Discourse API

URL: `/admin/site_settings/rate_limit_new_user_create_topic`

Delete Discourse API

URL: `/admin/site_settings/rate_limit_new_user_create_topic`

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/rate_limit_new_user_create_topic

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsRateLimitNewUserCreateTopic. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsRateLimitNewUserCreateTopic are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35.1.18 putAdminSiteSettingsTL2AdditionalLikesPerDayMultiplier

tl2 additional likes per day multiplier

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl2_additional_likes_per_day_multiplier

Insert Discourse API
URL: /admin/site_settings/tl2_additional_likes_per_day_multiplier

Update Discourse API
URL: /admin/site_settings/tl2_additional_likes_per_day_multiplier

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Delete Discourse API
URL: /admin/site_settings/tl2_additional_likes_per_day_multiplier

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl2_additional_likes_per_day_multiplier

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl2AdditionalLikesPerDayMultiplier. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTl2AdditionalLikesPerDayMultiplier are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35.1.19 putAdminSiteSettingsTl3AdditionalLikesPerDayMultiplier

tl3 additional likes per day multiplier

Catalog: Discourse

Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl3_additional_likes_per_day_multiplier
Insert Discourse API
URL: /admin/site_settings/tl3_additional_likes_per_day_multiplier

Update Discourse API
URL: /admin/site_settings/tl3_additional_likes_per_day_multiplier

Delete Discourse API
URL: /admin/site_settings/tl3_additional_likes_per_day_multiplier

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl3_additional_likes_per_day_multiplier

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3AdditionalLikesPerDayMultiplier. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTl3AdditionalLikesPerDayMultiplier are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

35.1.20 putAdminSiteSettingsTl4AdditionalLikesPerDayMultiplier

tl4 additional likes per day multiplier

Catalog: Discourse

Schema: SiteSettingsRateLimits
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl4_additional_likes_per_day_multiplier

Insert Discourse API
URL: /admin/site_settings/tl4_additional_likes_per_day_multiplier

Update Discourse API
URL: /admin/site_settings/tl4_additional_likes_per_day_multiplier

Delete Discourse API
URL: /admin/site_settings/tl4_additional_likes_per_day_multiplier

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl4_additional_likes_per_day_multiplier

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl4AdditionalLikesPerDayMultiplier`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTl4AdditionalLikesPerDayMultiplier` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
35.1.21 putAdminSiteSettingsUniquePostsMin
unique posts mins
Catalog: Discourse
Schema: SiteSettingsRateLimits

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/unique_posts_min
Insert Discourse API URL: /admin/site_settings/unique_posts_min
Update Discourse API URL: /admin/site_settings/unique_posts_min
Delete Discourse API URL: /admin/site_settings/unique_posts_min
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/unique_posts_min

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsUniquePostsMin. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsUniquePostsMin are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36 Schema: SiteSettingsRequired
36.1 Tables
36.1.1 putAdminSiteSettingsAppleTouchIconUrl

Update the apple touch icon url

Catalog: Discourse
Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/apple_touch_icon_url
Insert Discourse API URL: /admin/site_settings/apple_touch_icon_url
Update Discourse API URL: /admin/site_settings/apple_touch_icon_url
Delete Discourse API URL: /admin/site_settings/apple_touch_icon_url

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/apple_touch_icon_url

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsAppleTouchIconUrl. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAppleTouchIconUrl are
shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36.1.2 putAdminSiteSettingsCompanyDomain
Update the company domain
Catalog: Discourse
Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/company_domain
Insert Discourse API URL: /admin/site_settings/company_domain
Update Discourse API URL: /admin/site_settings/company_domain
Delete Discourse API URL: /admin/site_settings/company_domain
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/company_domain

Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCompanyDomain. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsCompanyDomain are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36.1.3 putAdminSiteSettingsCompanyFullName

Update the company full name

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table nativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/company_full_name

Insert Discourse API URL: /admin/site_settings/company_full_name

Update Discourse API URL: /admin/site_settings/company_full_name

Delete Discourse API URL: /admin/site_settings/company_full_name

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/company_full_name

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCompanyFullName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsCompanyFullName are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36.1.4 putAdminSiteSettingsCompanyShortName

Update the company short name

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/company_short_name

Insert Discourse API URL: /admin/site_settings/company_short_name

Update Discourse API URL: /admin/site_settings/company_short_name

Delete Discourse API URL: /admin/site_settings/company_short_name

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/company_short_name

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCompanyShortName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsCompanyShortName are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36.1.5 putAdminSiteSettingsContactEmail

Update the contact email

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/contact_email

Insert Discourse API URL: /admin/site_settings/contact_email

Update Discourse API URL: /admin/site_settings/contact_email

Delete Discourse API URL: /admin/site_settings/contact_email

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/contact_email

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsContactEmail. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsContactEmail are shown
below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36.1.6 putAdminSiteSettingsContactUrl

Update the contact url

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/contact_url
Insert Discourse API URL: /admin/site_settings/contact_url
Update Discourse API URL: /admin/site_settings/contact_url
Delete Discourse API URL: /admin/site_settings/contact_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/contact_url

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsContactUrl. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsContactUrl are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
36.1.7 putAdminSiteSettingsDefaultOpengraphImageUrl

Update the default opengraph image url.

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_opengraph_image_url

Insert Discourse API
URL: /admin/site_settings/default_opengraph_image_url

Update Discourse API
URL: /admin/site_settings/default_opengraph_image_url

Delete Discourse API
URL: /admin/site_settings/default_opengraph_image_url

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_opengraph_image_url

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultOpengraphImageUrl. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsDefaultOpengraphImageUrl are shown below. Each column has an SQL data type.
36.1.8 putAdminSiteSettingsDigestLogoUrl

Update the digest logo url

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/digest_logo_url

Insert Discourse API URL: /admin/site_settings/digest_logo_url

Update Discourse API URL: /admin/site_settings/digest_logo_url

Delete Discourse API URL: /admin/site_settings/digest_logo_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/digest_logo_url

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDigestLogoUrl. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>⊙</td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsDigestLogoUrl are shown below. Each column has an SQL data type.
36.1.9 putAdminSiteSettingsExcludeRelNoFollowDomains

Update the exclude rel nofollow domains

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/exclude_rel_nofollow_domains

Insert Discourse API
URL: /admin/site_settings/exclude_rel_nofollow_domains

Update Discourse API
URL: /admin/site_settings/exclude_rel_nofollow_domains

Delete Discourse API
URL: /admin/site_settings/exclude_rel_nofollow_domains

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/exclude_rel_nofollow_domains

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsExcludeRelNoFollowDomains. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `putAdminSiteSettingsExcludeRelNofollowDomains` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 36.1.10 `putAdminSiteSettingsFaviconUrl`

Update the favicon url

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/favicon_url

Insert Discourse API URL: /admin/site_settings/favicon_url

Update Discourse API URL: /admin/site_settings/favicon_url

Delete Discourse API URL: /admin/site_settings/favicon_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/favicon_url

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFaviconUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns
The columns of the table function putAdminSiteSettingsFaviconUrl are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

36.1.11 putAdminSiteSettingsLogoSmallUrl

Update the small logo url

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/logo_small_url
Insert Discourse API URL: /admin/site_settings/logo_small_url
Update Discourse API URL: /admin/site_settings/logo_small_url
Delete Discourse API URL: /admin/site_settings/logo_small_url

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/logo_small_url

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsLogoSmallUrl. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsLogoSmallUrl are shown below. Each column has an SQL data type.
36.1.12 putAdminSiteSettingsLogoUrl

Update logo url
Catalog: Discourse
Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/logo_url
Insert Discourse API URL: /admin/site_settings/logo_url
Update Discourse API URL: /admin/site_settings/logo_url
Delete Discourse API URL: /admin/site_settings/logo_url
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/logo_url

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLogoUrl`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsLogoUrl` are shown below. Each column has an SQL data type.
36.1.13 putAdminSiteSettingsMobileLogoUrl

Update the mobile logo url

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/mobile_logo_url`

Insert Discourse API URL: `/admin/site_settings/mobile_logo_url`

Update Discourse API URL: `/admin/site_settings/mobile_logo_url`

Delete Discourse API URL: `/admin/site_settings/mobile_logo_url`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/mobile_logo_url`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMobileLogoUrl. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsMobileLogoUrl are shown below. Each column has an SQL data type.
### 36.1.14 putAdminSiteSettingsNotificationEmail

Update the notification email

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/notification_email`

Insert Discourse API URL: `/admin/site_settings/notification_email`

Update Discourse API URL: `/admin/site_settings/notification_email`

Delete Discourse API URL: `/admin/site_settings/notification_email`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/notification_email

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNotificationEmail`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsNotificationEmail` are shown below. Each column has an SQL data type.

---

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
36.1.15 putAdminSiteSettingsSiteContactUsername

Update the site contact username

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/site_contact_username

Insert Discourse API URL: /admin/site_settings/site_contact_username

Update Discourse API URL: /admin/site_settings/site_contact_username

Delete Discourse API URL: /admin/site_settings/site_contact_username

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/site_contact_username

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSiteContactUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function putAdminSiteSettingsSiteContactUsername are shown below. Each column has an SQL data type.
Name | Data Type | Label | Required | Documentation
--- | --- | --- | --- | ---
RESULT | string | Result | ☐ | Outcome of operation as single plain text column.

36.1.16 putAdminSiteSettingsSiteDescription

Update the site description

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/site_description

Insert Discourse API URL: /admin/site_settings/site_description

Update Discourse API URL: /admin/site_settings/site_description

Delete Discourse API URL: /admin/site_settings/site_description

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/site_description

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSiteDescription. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsSiteDescription are shown below. Each column has an SQL data type.
### 36.1.17 putAdminSiteSettingsTitle

Update the site title.

Catalog: Discourse

Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/title`

Insert Discourse API URL: `/admin/site_settings/title`

Update Discourse API URL: `/admin/site_settings/title`

Delete Discourse API URL: `/admin/site_settings/title`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/title`

#### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTitle`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table Function Columns

The columns of the table function `putAdminSiteSettingsTitle` are shown below. Each column has an SQL data type.
36.1.18 putAdminSiteSettingsTwitterSummaryLargeImageUrl

Update the twitter summary large image url

Catalog: Discourse
Schema: SiteSettingsRequired

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/twitter_summary_large_image_url

Insert Discourse API
URL: /admin/site_settings/twitter_summary_large_image_url

Update Discourse API
URL: /admin/site_settings/twitter_summary_large_image_url

Delete Discourse API
URL: /admin/site_settings/twitter_summary_large_image_url

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/twitter_summary_large_image_url

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTwitterSummaryLargeImageUrl. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The columns of the table function 
putAdminSiteSettingsTwitterSummaryLargeImageUrl are shown below. Each 
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

37 Schema: SiteSettingsSearch
37.1 Tables
37.1.1 putAdminSiteSettingsMinSearchTermLength

min search term length
Catalog: Discourse
Schema: SiteSettingsSearch

This is a read-only table function. The Discourse API may not support changing the data or 
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the 
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/min_search_term_length
Insert Discourse API URL: /admin/site_settings/min_search_term_length
Update Discourse API URL: /admin/site_settings/min_search_term_length
Delete Discourse API URL: /admin/site_settings/min_search_term_length

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/min_search_term_length

Table Function Parameters

The following parameters can be used to control the behaviour of the table function 
putAdminSiteSettingsMinSearchTermLength. A value must be provided at all times for 
required parameters, but optional parameters in general do not need to have a value and the 
execution will default to a pre-defined behaviour. Values can be specified by position and by 
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed 
value. For example with `select * from table(value1, value2, value3)` on a table with four 
parameters will use the default value for the fourth parameter and the specified values for the 
first three.

Value specification by name is done by listing all values that require a value. For example 
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the 
default values for the second and fourth parameters and the specified values for the first and 
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsMinSearchTermLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

37.1.2 `putAdminSiteSettingsSearchPreferRecentPosts`

search prefer recent posts

Catalog: Discourse

Schema: SiteSettingsSearch

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/search_prefer_recent_posts

Insert Discourse API URL: /admin/site_settings/search_prefer_recent_posts

Update Discourse API

URL: /admin/site_settings/search_prefer_recent_posts

Delete Discourse API

URL: /admin/site_settings/search_prefer_recent_posts

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/search_prefer_recent_posts

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSearchPreferRecentPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsSearchPreferRecentPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

37.1.3 `putAdminSiteSettingsSearchRecentPostsSize`

search recent posts size

Catalog: Discourse

Schema: SiteSettingsSearch

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/search_recent_posts_size`

Insert Discourse API URL: `/admin/site_settings/search_recent_posts_size`

Update Discourse API URL: `/admin/site_settings/search_recent_posts_size`

Delete Discourse API URL: `/admin/site_settings/search_recent_posts_size`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/search_recent_posts_size`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSearchRecentPostsSize`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsSearchRecentPostsSize are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

37.1.4 putAdminSiteSettingsSearchTokenizeChineseJapaneseKorean

search tokenize chinese japanese korean

Catalog: Discourse

Schema: SiteSettingsSearch

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/search_tokenize_chinese_japanese_korean

Insert Discourse API

URL: /admin/site_settings/search_tokenize_chinese_japanese_korean

Update Discourse API

URL: /admin/site_settings/search_tokenize_chinese_japanese_korean

Delete Discourse API

URL: /admin/site_settings/search_tokenize_chinese_japanese_korean

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/search_tokenize_chinese_japanese_korean

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsSearchTokenizeChineseJapaneseKorean. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsSearchTokenizeChineseJapaneseKorean` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

38 Schema: SiteSettingsSecurity
38.1 Tables
38.1.1 putAdminSiteSettingsAllowIndexInRobotsTxt

allow index in robots txt

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/allow_index_in_robots_txt`

Insert Discourse API URL: `/admin/site_settings/allow_index_in_robots_txt`

Update Discourse API URL: `/admin/site_settings/allow_index_in_robots_txt`

Delete Discourse API URL: `/admin/site_settings/allow_index_in_robots_txt`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/allow_index_in_robots_txt`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowIndexInRobotsTxt`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the...
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAllowIndexInRobotsTxt` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 38.1.2 putAdminSiteSettingsAllowModeratorsToCreateCategories

allow moderators to create categories

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/allow_moderators_to_create_categories`

Insert Discourse API

URL: `/admin/site_settings/allow_moderators_to_create_categories`

Update Discourse API

URL: `/admin/site_settings/allow_moderators_to_create_categories`

Delete Discourse API

URL: `/admin/site_settings/allow_moderators_to_create_categories`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/allow_moderators_to_create_categories

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowModeratorsToCreateCategories`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsAllowModeratorsToCreateCategories are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

38.1.3 putAdminSiteSettingsCorsOrigins

cors origins

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/cors_origins

Insert Discourse API URL: /admin/site_settings/cors_origins

Update Discourse API URL: /admin/site_settings/cors_origins

Delete Discourse API URL: /admin/site_settings/cors_origins

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/cors_origins

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsCorsOrigins. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsCorsOrigins are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**38.1.4 putAdminSiteSettingsEnableEscapedFragments**

enable escaped fragments

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_escaped_fragments

Insert Discourse API URL: /admin/site_settings/enable_escaped_fragments

Update Discourse API URL: /admin/site_settings/enable_escaped_fragments

Delete Discourse API URL: /admin/site_settings/enable_escaped_fragments

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/enable_escaped_fragments

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEnableEscapedFragments. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☑️️️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsEnableEscapedFragments` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☑️️️</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 38.1.5 `putAdminSiteSettingsEnableNoscriptSupport`

enable noscript support

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_noscript_support`

Insert Discourse API URL: `/admin/site_settings/enable_noscript_support`

Update Discourse API URL: `/admin/site_settings/enable_noscript_support`

Delete Discourse API URL: `/admin/site_settings/enable_noscript_support`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/enable_noscript_support`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableNoscriptSupport`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsEnableNoscriptSupport are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**38.1.6 putAdminSiteSettingsForceHttps**

force https

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/force_https

Insert Discourse API URL: /admin/site_settings/force_https

Update Discourse API URL: /admin/site_settings/force_https

Delete Discourse API URL: /admin/site_settings/force_https

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/force_https

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsForceHttps. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsUseAdminIpWhitelist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**38.1.7 putAdminSiteSettingsUseAdminIpWhitelist**

use admin ip whitelist

Catalog: Discourse

Schema: SiteSettingsSecurity

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/use_admin_ip_whitelist`

Insert Discourse API URL: `/admin/site_settings/use_admin_ip_whitelist`

Update Discourse API URL: `/admin/site_settings/use_admin_ip_whitelist`

Delete Discourse API URL: `/admin/site_settings/use_admin_ip_whitelist`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/use_admin_ip_whitelist`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsUseAdminIpWhitelist`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsUseAdminIpWhitelist` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**39 Schema: SiteSettingsSpam**

**39.1 Tables**

**39.1.1 putAdminSiteSettingsAddRelNofollowToUserContent**

add rel nofollow to user content

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/add_rel_nofollow_to_user_content

Insert Discourse API

URL: /admin/site_settings/add_rel_nofollow_to_user_content

Update Discourse API

URL: /admin/site_settings/add_rel_nofollow_to_user_content

Delete Discourse API

URL: /admin/site_settings/add_rel_nofollow_to_user_content

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/add_rel_nofollow_to_user_content

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAddRelNofollowToUserContent`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and
the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsAddRelNofollowToUserContent` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.2 `putAdminSiteSettingsAutoBlockFastTypersMaxTrustLevel`

auto block fast typers max trust level

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/auto_block_fast_typers_max_trust_level

Insert Discourse API
URL: /admin/site_settings/auto_block_fast_typers_max_trust_level

Update Discourse API
URL: /admin/site_settings/auto_block_fast_typers_max_trust_level

Delete Discourse API
URL: /admin/site_settings/auto_block_fast_typers_max_trust_level

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/auto_block_fast_typers_max_trust_level`

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAutoBlockFastTypersMaxTrustLevel`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAutoBlockFastTypersMaxTrustLevel` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 39.1.3 putAdminSiteSettingsAutoBlockFastTypersOnFirstPost

auto block fast typers on first post

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/auto_block_fast_typers_on_first_post

Insert Discourse API

URL: /admin/site_settings/auto_block_fast_typers_on_first_post

Update Discourse API

URL: /admin/site_settings/auto_block_fast_typers_on_first_post

Delete Discourse API

URL: /admin/site_settings/auto_block_fast_typers_on_first_post

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/auto_block_fast_typers_on_first_post

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutoBlockFastTypersOnFirstPost. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAutoBlockFastTypersOnFirstPost are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.4 putAdminSiteSettingsAutoBlockFirstPostRegex

auto block first post regex

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/auto_block_first_post_regex

Insert Discourse API
URL: /admin/site_settings/auto_block_first_post_regex

Update Discourse API
URL: /admin/site_settings/auto_block_first_post_regex

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Delete Discourse API
URL: /admin/site_settings/auto_block_first_post_regex

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/auto_block_first_post_regex

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutoBlockFirstPostRegex. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAutoBlockFirstPostRegex are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.5 putAdminSiteSettingsAutoRespondToFlagActions

auto respond to flag actions
Catalog: Discourse
Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/auto_respond_to_flag_actions
Insert Discourse API
URL: /admin/site_settings/auto_respond_to_flag_actions

Update Discourse API
URL: /admin/site_settings/auto_respond_to_flag_actions

Delete Discourse API
URL: /admin/site_settings/auto_respond_to_flag_actions

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/auto_respond_to_flag_actions

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAutoRespondToFlagActions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAutoRespondToFlagActions are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.6 putAdminSiteSettingsCooldownMinutesAfterHidingPosts

cooldown minutes after hiding posts

Catalog: Discourse

Schema: SiteSettingsSpam
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/cooldown_minutes_after_hiding_posts`

Insert Discourse API URL: `/admin/site_settings/cooldown_minutes_after_hiding_posts`

Update Discourse API URL: `/admin/site_settings/cooldown_minutes_after_hiding_posts`

Delete Discourse API URL: `/admin/site_settings/cooldown_minutes_after_hiding_posts`

Field Selection Method: NotRequired

Select Discourse API Operation: 
`put /admin/site_settings/cooldown_minutes_after_hiding_posts`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsCooldownMinutesAfterHidingPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsCooldownMinutesAfterHidingPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
39.1.7 putAdminSiteSettingsFlagSockpuppets

flag sockpuppets
Catalog: Discourse
Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/flag_sockpuppets
Insert Discourse API URL: /admin/site_settings/flag_sockpuppets
Update Discourse API URL: /admin/site_settings/flag_sockpuppets
Delete Discourse API URL: /admin/site_settings/flag_sockpuppets
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/flag_sockpuppets

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsFlagSockpuppets. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsFlagSockpuppets are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
39.1.8 `putAdminSiteSettingsFlagsRequiredToHidePost`

flags required to hide post

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/flags_required_to_hide_post`

Insert Discourse API

URL: `/admin/site_settings/flags_required_to_hide_post`

Update Discourse API

URL: `/admin/site_settings/flags_required_to_hide_post`

Delete Discourse API

URL: `/admin/site_settings/flags_required_to_hide_post`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/flags_required_to_hide_post`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFlagsRequiredToHidePost`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsFlagsRequiredToHidePost` are shown below. Each column has an SQL data type.
39.1.9 `putAdminSiteSettingsLevenshteinDistanceSpammerEmails`

Levenshtein distance spammer emails

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/levenshtein_distance_spammer_emails`

Insert Discourse API
URL: `/admin/site_settings/levenshtein_distance_spammer_emails`

Update Discourse API
URL: `/admin/site_settings/levenshtein_distance_spammer_emails`

Delete Discourse API
URL: `/admin/site_settings/levenshtein_distance_spammer_emails`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/levenshtein_distance_spammer_emails`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLevenshteinDistanceSpammerEmails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Inventive Software B.V., the Netherlands. All rights reserved.
Table Function Columns

The columns of the table function `putAdminSiteSettingsLevenshteinDistanceSpammerEmails` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.10 putAdminSiteSettingsMaxAgeUnmatchedEmails

max age unmatched emails

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_age_unmatched_emails`

Insert Discourse API URL: `/admin/site_settings/max_age_unmatched_emails`

Update Discourse API URL: `/admin/site_settings/max_age_unmatched_emails`

Delete Discourse API URL: `/admin/site_settings/max_age_unmatched_emails`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/max_age_unmatched_emails

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxAgeUnmatchedEmails`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsMaxAgeUnmatchedEmails are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.11 putAdminSiteSettingsMaxAgeUnmatchedIps

max age unmatched ips

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_age_unmatched_ips

Insert Discourse API URL: /admin/site_settings/max_age_unmatched_ips

Update Discourse API URL: /admin/site_settings/max_age_unmatched_ips

Delete Discourse API URL: /admin/site_settings/max_age_unmatched_ips

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/max_age_unmatched_ips

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxAgeUnmatchedIps. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxAgeUnmatchedIps` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**39.1.12 putAdminSiteSettingsMaxNewAccountsPerRegistrationIp**

max new accounts per registration ip

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/max_new_accounts_per_registration_ip`

Insert Discourse API

URL: `/admin/site_settings/max_new_accounts_per_registration_ip`

Update Discourse API

URL: `/admin/site_settings/max_new_accounts_per_registration_ip`

Delete Discourse API

URL: `/admin/site_settings/max_new_accounts_per_registration_ip`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/max_new_accounts_per_registration_ip`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxNewAccountsPerRegistrationIp`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxNewAccountsPerRegistrationIp` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

39.1.13 putAdminSiteSettingsMinBanEntriesForRollUp

min ban entries for roll up

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/min_ban_entries_for_roll_up`

Insert Discourse API
URL: `/admin/site_settings/min_ban_entries_for_roll_up`

Update Discourse API
URL: `/admin/site_settings/min_ban_entries_for_roll_up`

Delete Discourse API
URL: `/admin/site_settings/min_ban_entries_for_roll_up`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/min_ban_entries_for_roll_up`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinBanEntriesForRollUp`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMinBanEntriesForRollUp` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**39.1.14 putAdminSiteSettingsMinFirstPostTypingTime**

min first post typing time

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_first_post_typing_time`

Insert Discourse API URL: `/admin/site_settings/min_first_post_typing_time`

Update Discourse API URL: `/admin/site_settings/min_first_post_typing_time`

Delete Discourse API URL: `/admin/site_settings/min_first_post_typing_time`

Field Selection Method: NotRequired

Select Discourse API Operation: 
put /admin/site_settings/min_first_post_typing_time

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinFirstPostTypingTime`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsMinFirstPostTypingTime` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.15 putAdminSiteSettingsNewuserSpamHostThreshold

newuser spam host threshold

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/newuser_spam_host_threshold

Insert Discourse API

URL: /admin/site_settings/newuser_spam_host_threshold

Update Discourse API

URL: /admin/site_settings/newuser_spam_host_threshold

Delete Discourse API

URL: /admin/site_settings/newuser_spam_host_threshold

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/newuser_spam_host_threshold

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNewuserSpamHostThreshold`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsNewuserSpamHostThreshold` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**39.1.16 putAdminSiteSettingsNotifyModsWhenUserBlocked**

notify mods when user blocked

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/notify_mods_when_user_blocked

Insert Discourse API
URL: /admin/site_settings/notify_mods_when_user_blocked

Update Discourse API
URL: /admin/site_settings/notify_mods_when_user_blocked

Delete Discourse API
URL: /admin/site_settings/notify_mods_when_user_blocked

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/notify_mods_when_user_blocked

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNotifyModsWhenUserBlocked`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsNotifyModsWhenUserBlocked` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 39.1.17 putAdminSiteSettingsNumFlaggersToCloseTopic

num flaggers to close topic

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/num_flaggers_to_close_topic`

Insert Discourse API

URL: `/admin/site_settings/num_flaggers_to_close_topic`

Update Discourse API

URL: `/admin/site_settings/num_flaggers_to_close_topic`

Delete Discourse API

URL: `/admin/site_settings/num_flaggers_to_close_topic`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/num_flaggers_to_close_topic
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNumFlaggersToCloseTopic`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsNumFlaggersToCloseTopic` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.18 putAdminSiteSettingsNumFlaggersToCloseTopic

num flags to close topic

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/num_flags_to_close_topic`

Insert Discourse API URL: `/admin/site_settings/num_flags_to_close_topic`

Update Discourse API URL: `/admin/site_settings/num_flags_to_close_topic`

Delete Discourse API URL: `/admin/site_settings/num_flags_to_close_topic`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/num_flags_to_close_topic
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsNumFlagsToCloseTopic. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsNumFlagsToCloseTopic are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.19 putAdminSiteSettingsNumSpamFlagsToBlockNewUser

num spam flags to block new user

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/num_spam_flags_to_block_new_user

Insert Discourse API

URL: /admin/site_settings/num_spam_flags_to_block_new_user

Update Discourse API

URL: /admin/site_settings/num_spam_flags_to_block_new_user

Delete Discourse API

URL: /admin/site_settings/num_spam_flags_to_block_new_user

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/num_spam_flags_to_block_new_user

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNumSpamFlagsToBlockNewUser`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsNumSpamFlagsToBlockNewUser` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.20 `putAdminSiteSettingsNumTl3FlagsToBlockNewUser`

num tl3 flags to block new user

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/num_tl3_flags_to_block_new_user

Insert Discourse API
URL: /admin/site_settings/num_tl3_flags_to_block_new_user

Update Discourse API
URL: /admin/site_settings/num_tl3_flags_to_block_new_user
Delete Discourse API
URL: /admin/site_settings/num_tl3_flags_to_block_new_user

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/num_tl3_flags_to_block_new_user

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsNumTl3FlagsToBlockNewUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsNumTl3FlagsToBlockNewUser are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.21 putAdminSiteSettingsNumTl3UsersToBlockNewUser

num tl3 users to block new user

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/num_tl3_users_to_block_new_user
Insert Discourse API
URL: /admin/site_settings/num_tl3_users_to_block_new_user

Update Discourse API
URL: /admin/site_settings/num_tl3_users_to_block_new_user

Delete Discourse API
URL: /admin/site_settings/num_tl3_users_to_block_new_user

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/num_tl3_users_to_block_new_user

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsNumTl3UsersToBlockNewUser. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsNumTl3UsersToBlockNewUser are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

39.1.22 putAdminSiteSettingsNumUsersToBlockNewUser

num users to block new user

Catalog: Discourse

Schema: SiteSettingsSpam
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**
**URL:** `/admin/site_settings/num_users_to_block_new_user`

**Insert Discourse API**
**URL:** `/admin/site_settings/num_users_to_block_new_user`

**Update Discourse API**
**URL:** `/admin/site_settings/num_users_to_block_new_user`

**Delete Discourse API**
**URL:** `/admin/site_settings/num_users_to_block_new_user`

Field Selection Method: `NotRequired`

**Select Discourse API Operation:**
`put /admin/site_settings/num_users_to_block_new_user`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsNumUsersToBlockNewUser`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsNumUsersToBlockNewUser` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
39.1.23 putAdminSiteSettingsWhiteListedSpamHostDomains

white listed spam host domains

Catalog: Discourse

Schema: SiteSettingsSpam

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/white_listed_spam_host_domains

Insert Discourse API
URL: /admin/site_settings/white_listed_spam_host_domains

Update Discourse API
URL: /admin/site_settings/white_listed_spam_host_domains

Delete Discourse API
URL: /admin/site_settings/white_listed_spam_host_domains

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/white_listed_spam_host_domains

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsWhiteListedSpamHostDomains. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsWhiteListedSpamHostDomains are shown below. Each column has an SQL data type.
40 Schema: SiteSettingsTags
40.1 Tables
40.1.1 putAdminSiteSettingsMaxTagLength

"max tag length"

Catalog: Discourse
Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_tag_length
Insert Discourse API URL: /admin/site_settings/max_tag_length
Update Discourse API URL: /admin/site_settings/max_tag_length
Delete Discourse API URL: /admin/site_settings/max_tag_length

Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/max_tag_length

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMaxTagLength. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function putAdminSiteSettingsMaxTagLength are shown below. Each column has an SQL data type.
### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxTagSearchResults`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxTagSearchResults` are shown below. Each column has an SQL data type.
40.1.3 `putAdminSiteSettingsMaxTagsInFilterList`

"max tags in filter list"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_tags_in_filter_list

Insert Discourse API URL: /admin/site_settings/max_tags_in_filter_list

Update Discourse API URL: /admin/site_settings/max_tags_in_filter_list

Delete Discourse API URL: /admin/site_settings/max_tags_in_filter_list

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_tags_in_filter_list

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxTagsInFilterList`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxTagsInFilterList` are shown below. Each column has an SQL data type.
### 40.1.4 putAdminSiteSettingsMaxTagsPerTopic

"max tags per topic"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_tags_per_topic`

Insert Discourse API URL: `/admin/site_settings/max_tags_per_topic`

Update Discourse API URL: `/admin/site_settings/max_tags_per_topic`

Delete Discourse API URL: `/admin/site_settings/max_tags_per_topic`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/max_tags_per_topic`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxTagsPerTopic`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxTagsPerTopic` are shown below. Each column has an SQL data type.
40.1.5 putAdminSiteSettingsMinTrustLevelToTagTopics
"min trust level to tag topics"
Catalog: Discourse
Schema: SiteSettingsTags
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/min_trust_level_to_tag_topics
Insert Discourse API
URL: /admin/site_settings/min_trust_level_to_tag_topics
Update Discourse API
URL: /admin/site_settings/min_trust_level_to_tag_topics
Delete Discourse API
URL: /admin/site_settings/min_trust_level_to_tag_topics
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/min_trust_level_to_tag_topics

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMinTrustLevelToTagTopics. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**
The columns of the table function `putAdminSiteSettingsMinTrustLevelToTagTopics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 40.1.6 putAdminSiteSettingsMinTrustToCreateTag

"min trust to create tag"

**Catalog:** Discourse

**Schema:** SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_trust_to_create_tag`

Insert Discourse API URL: `/admin/site_settings/min_trust_to_create_tag`

Update Discourse API URL: `/admin/site_settings/min_trust_to_create_tag`

Delete Discourse API URL: `/admin/site_settings/min_trust_to_create_tag`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/min_trust_to_create_tag`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinTrustToCreateTag`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

- **Value specification by position** is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

- **Value specification by name** is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**
The columns of the table function `putAdminSiteSettingsMinTrustToCreateTag` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**40.1.7 putAdminSiteSettingsRemoveMutedTagsFromLatest**
"remove muted tags from latest"

**Catalog:** Discourse

**Schema:** SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the `tableNativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API**
URL: `/admin/site_settings/remove_muted_tags_from_latest`

**Insert Discourse API**
URL: `/admin/site_settings/remove_muted_tags_from_latest`

**Update Discourse API**
URL: `/admin/site_settings/remove_muted_tags_from_latest`

**Delete Discourse API**
URL: `/admin/site_settings/remove_muted_tags_from_latest`

**Field Selection Method:** NotRequired

**Select Discourse API Operation:**
`put /admin/site_settings/remove_muted_tags_from_latest`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsRemoveMutedTagsFromLatest`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function
putAdminSiteSettingsRemoveMutedTagsFromLatest are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

40.1.8 putAdminSiteSettingsShowFilterByTag

"show filter by tag"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/show_filter_by_tag

Insert Discourse API URL: /admin/site_settings/show_filter_by_tag

Update Discourse API URL: /admin/site_settings/show_filter_by_tag

Delete Discourse API URL: /admin/site_settings/show_filter_by_tag

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/show_filter_by_tag

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsShowFilterByTag. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsShowFilterByTag` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

40.1.9 putAdminSiteSettingsStaffTags

"staff tags"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/staff_tags`

Insert Discourse API URL: `/admin/site_settings/staff_tags`

Update Discourse API URL: `/admin/site_settings/staff_tags`

Delete Discourse API URL: `/admin/site_settings/staff_tags`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/staff_tags`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsStaffTags`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `putAdminSiteSettingsStaffTags` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 40.1.10 putAdminSiteSettingsSuppressOverlappingTagsInList

"suppress overlapping tags in list"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/suppress_overlapping_tags_in_list

Insert Discourse API

URL: /admin/site_settings/suppress_overlapping_tags_in_list

Update Discourse API

URL: /admin/site_settings/suppress_overlapping_tags_in_list

Delete Discourse API

URL: /admin/site_settings/suppress_overlapping_tags_in_list

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/suppress_overlapping_tags_in_list

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsSuppressOverlappingTagsInList`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsSuppressOverlappingTagsInList` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

40.1.11 putAdminSiteSettingsTaggingEnabled

"tagging enabled"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tagging_enabled`

Insert Discourse API URL: `/admin/site_settings/tagging_enabled`

Update Discourse API URL: `/admin/site_settings/tagging_enabled`

Delete Discourse API URL: `/admin/site_settings/tagging_enabled`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/tagging_enabled

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTaggingEnabled`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsTaggingEnabled` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

40.1.12 `putAdminSiteSettingsTagsSortAlphabetically`

"tags sort alphabetically"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tags_sort_alphabetically`

Insert Discourse API URL: `/admin/site_settings/tags_sort_alphabetically`

Update Discourse API URL: `/admin/site_settings/tags_sort_alphabetically`

Delete Discourse API URL: `/admin/site_settings/tags_sort_alphabetically`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/tags_sort_alphabetically`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTagsSortAlphabetically`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsTagsSortAlphabetically` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

40.1.13 `putAdminSiteSettingsTagStyle`

"tag style"

Catalog: Discourse

Schema: SiteSettingsTags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/tag_style

Insert Discourse API URL: /admin/site_settings/tag_style

Update Discourse API URL: /admin/site_settings/tag_style

Delete Discourse API URL: /admin/site_settings/tag_style

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/tag_style`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTagStyle`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The columns of the table function `putAdminSiteSettingsTagStyle` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 41 Schema: SiteSettingsTrustLevels

#### 41.1 Tables

**41.1.1 putAdminSiteSettingsDefaultInviteeTrustLevel**

default invitee trust level

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

- Select Discourse API URL: `/admin/site_settings/defaultInviteeTrustLevel`
- Insert Discourse API URL: `/admin/site_settings/defaultInviteeTrustLevel`
- Update Discourse API URL: `/admin/site_settings/defaultInviteeTrustLevel`
- Delete Discourse API URL: `/admin/site_settings/defaultInviteeTrustLevel`
- Field Selection Method: `NotRequired`
- Select Discourse API Operation: `put /admin/site_settings/default_invitee_trust_level`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultInviteeTrustLevel`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function
`putAdminSiteSettingsDefaultInviteeTrustLevel` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 41.1.2 putAdminSiteSettingsDefaultTrustLevel

default trust level

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/default_trust_level`

Insert Discourse API URL: `/admin/site_settings/default_trust_level`

Update Discourse API URL: `/admin/site_settings/default_trust_level`

Delete Discourse API URL: `/admin/site_settings/default_trust_level`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/default_trust_level`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultTrustLevel`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultTrustLevel` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 41.1.3 putAdminSiteSettingsMinTrustToAllowSelfWiki

min trust to allow self wiki

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/min_trust_to_allow_self_wiki

Insert Discourse API
URL: /admin/site_settings/min_trust_to_allow_self_wiki

Update Discourse API
URL: /admin/site_settings/min_trust_to_allow_self_wiki

Delete Discourse API
URL: /admin/site_settings/min_trust_to_allow_self_wiki

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/min_trust_to_allow_self_wiki

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinTrustToAllowSelfWiki`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function putAdminSiteSettingsMinTrustToAllowSelfWiki are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 41.1.4 putAdminSiteSettingsMinTrustToCreateTopic

min trust to create topic

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/min_trust_to_create_topic
Insert Discourse API URL: /admin/site_settings/min_trust_to_create_topic
Update Discourse API URL: /admin/site_settings/min_trust_to_create_topic
Delete Discourse API URL: /admin/site_settings/min_trust_to_create_topic

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/min_trust_to_create_topic

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMinTrustToCreateTopic. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMinTrustToCreateTopic` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 41.1.5 putAdminSiteSettingsMinTrustToEditPost

min trust to edit post

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_trust_to_edit_post`

Insert Discourse API URL: `/admin/site_settings/min_trust_to_edit_post`

Update Discourse API URL: `/admin/site_settings/min_trust_to_edit_post`

Delete Discourse API URL: `/admin/site_settings/min_trust_to_edit_post`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/min_trust_to_edit_post`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinTrustToEditPost`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMinTrustToEditPost` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.6 putAdminSiteSettingsMinTrustToEditWikiPost**

min trust to edit wiki post

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the `table NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/min_trust_to_edit_wiki_post`

Insert Discourse API
URL: `/admin/site_settings/min_trust_to_edit_wiki_post`

Update Discourse API
URL: `/admin/site_settings/min_trust_to_edit_wiki_post`

Delete Discourse API
URL: `/admin/site_settings/min_trust_to_edit_wiki_post`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/min_trust_to_edit_wiki_post

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinTrustToEditWikiPost`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsMinTrustToSendMessages are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.7 putAdminSiteSettingsMinTrustToSendMessages**

**min trust to send messages**

*Catalog:* Discourse  
*Schema:* SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

**Select Discourse API**  
**URL:** /admin/site_settings/min_trust_to_send_messages

**Insert Discourse API URL:** /admin/site_settings/min_trust_to_send_messages

**Update Discourse API**  
**URL:** /admin/site_settings/min_trust_to_send_messages

**Delete Discourse API**  
**URL:** /admin/site_settings/min_trust_to_send_messages

**Field Selection Method:** NotRequired

**Select Discourse API Operation:**  
put /admin/site_settings/min_trust_to_send_messages

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMinTrustToSendMessages. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMinTrustToSendMessages` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 41.1.8 putAdminSiteSettingsT11RequiresReadPosts

t11 requires read posts

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/t11_requires_read_posts`

Insert Discourse API URL: `/admin/site_settings/t11_requires_read_posts`

Update Discourse API URL: `/admin/site_settings/t11_requires_read_posts`

Delete Discourse API URL: `/admin/site_settings/t11_requires_read_posts`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/t11_requires_read_posts`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsT11RequiresReadPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use
the default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl1RequiresReadPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.9 putAdminSiteSettingsTl1RequiresTimeSpentMins**

Tl1 requires time spent mins

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl1Requires_time_spent_mins

Insert Discourse API
URL: /admin/site_settings/tl1Requires_time_spent_mins

Update Discourse API
URL: /admin/site_settings/tl1Requires_time_spent_mins

Delete Discourse API
URL: /admin/site_settings/tl1Requires_time_spent_mins

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl1Requires_time_spent_mins

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function
`putAdminSiteSettingsTl1RequiresTimeSpentMins`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTl1RequiresTimeSpentMins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 41.1.10 putAdminSiteSettingsTl1RequiresTopicsEntered

tl1 requires topics entered

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl1_requires_topics_entered

Insert Discourse API
URL: /admin/site_settings/tl1_requires_topics_entered

Update Discourse API
URL: /admin/site_settings/tl1_requires_topics_entered

Delete Discourse API
URL: /admin/site_settings/tl1_requires_topics_entered

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl1_requires_topics_entered

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl1RequiresTopicsEntered`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl1RequiresTopicsEntered` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.11 putAdminSiteSettingsTl2RequiresDaysVisited**

tl2 requires days visited

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tl2_requires_days_visited`

Insert Discourse API URL: `/admin/site_settings/tl2_requires_days_visited`

Update Discourse API URL: `/admin/site_settings/tl2_requires_days_visited`

Delete Discourse API URL: `/admin/site_settings/tl2_requires_days_visited`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl2_requires_days_visited

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl2RequiresDaysVisited`. A value must be provided at all times for...
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl2RequiresDaysVisited` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.12 putAdminSiteSettingsTl2RequiresLikesGiven**

tl2 requires likes given

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tl2_requires_likes_given`

Insert Discourse API URL: `/admin/site_settings/tl2_requires_likes_given`

Update Discourse API URL: `/admin/site_settings/tl2_requires_likes_given`

Delete Discourse API URL: `/admin/site_settings/tl2_requires_likes_given`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/tl2_requires_likes_given

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl2RequiresLikesGiven`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl2RequiresLikesGiven` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.13 putAdminSiteSettingsTl2RequiresLikesReceived**

tl2 requires likes received

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/tl2RequiresLikesReceived`

Insert Discourse API
URL: `/admin/site_settings/tl2RequiresLikesReceived`

Update Discourse API
URL: `/admin/site_settings/tl2RequiresLikesReceived`

Delete Discourse API
URL: `/admin/site_settings/tl2RequiresLikesReceived`

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl2RequiresLikesReceived

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl2RequiresLikesReceived`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTl2RequiresLikesReceived` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 41.1.14 putAdminSiteSettingsTl2RequiresReadPosts

tl2 requires read posts

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tl2_requires_read_posts`

Insert Discourse API URL: `/admin/site_settings/tl2_requires_read_posts`

Update Discourse API URL: `/admin/site_settings/tl2_requires_read_posts`

Delete Discourse API URL: `/admin/site_settings/tl2_requires_read_posts`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/tl2_requires_read_posts

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl2RequiresReadPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTl2RequiresReadPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 41.1.15 putAdminSiteSettingsTl2RequiresTimeSpentMins

`tl2 requires time spent mins`

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/tl2_requires_time_spent_mins

Insert Discourse API

URL: /admin/site_settings/tl2_requires_time_spent_mins

Update Discourse API

URL: /admin/site_settings/tl2_requires_time_spent_mins

Delete Discourse API

URL: /admin/site_settings/tl2_requires_time_spent_mins

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/tl2_requires_time_spent_mins
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl2RequiresTimeSpentMins`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsTl2RequiresTimeSpentMins` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.16 `putAdminSiteSettingsTl2RequiresTopicReplyCount`

tl2 requires topic reply count

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/tl2_requires_topic_reply_count

Insert Discourse API

URL: /admin/site_settings/tl2_requires_topic_reply_count

Update Discourse API

URL: /admin/site_settings/tl2_requires_topic_reply_count

Delete Discourse API

URL: /admin/site_settings/tl2_requires_topic_reply_count
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/tl2_requires_topic_reply_count

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl2RequiresTopicReplyCount. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>[]</td>
<td>[]</td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsTl2RequiresTopicReplyCount are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>[]</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.17 putAdminSiteSettingsTl2RequiresTopicsEntered**

tl2 requires topics entered

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl2_requires_topics_entered

Insert Discourse API
URL: /admin/site_settings/tl2_requires_topics_entered
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl2RequiresTopicsEntered`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsTl2RequiresTopics Entered` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.18 putAdminSiteSettingsTl3LinksNoFollow

tl3 links no follow

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tl3_links_no_follow`
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3LinksNoFollow. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTl3LinksNoFollow are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.19 putAdminSiteSettingsTl3PromotionMinDuration

tl3 promotion min duration

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/tl3_promotion_min_duration
Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTL3PromotionMinDuration. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsTL3PromotionMinDuration are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.20 putAdminSiteSettingsTL3RequiresDaysVisited
TL3 requires days visited
Catalog: Discourse
Schema: SiteSettingsTrustLevels
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API URL: /admin/site_settings/tl3RequiresDaysVisited
Insert Discourse API URL: /admin/site_settings/tl3RequiresDaysVisited
Update Discourse API URL: /admin/site_settings/tl3RequiresDaysVisited
Delete Discourse API URL: /admin/site_settings/tl3RequiresDaysVisited
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/tl3RequiresDaysVisited

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3RequiresDaysVisited. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTl3RequiresDaysVisited are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.21 putAdminSiteSettingsTl3RequiresLikesGiven

tl3 requires likes given

Catalog: Discourse
Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/tl3RequiresLikesGiven
Insert Discourse API URL: /admin/site_settings/tl3_requires_likes_given
Update Discourse API URL: /admin/site_settings/tl3_requires_likes_given
Delete Discourse API URL: /admin/site_settings/tl3_requires_likes_given
Field Selection Method: NotRequired
Select Discourse API Operation: put /admin/site_settings/tl3_requires_likes_given

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl3RequiresLikesGiven`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `*select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `*select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl3RequiresLikesGiven` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.22 putAdminSiteSettingsTl3RequiresLikesReceived

tl3 requires likes received
Catalog: Discourse
Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl3_requires_likes_received

---

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Insert Discourse API
URL: /admin/site_settings/tl3_requires_likes_received

Update Discourse API
URL: /admin/site_settings/tl3_requires_likes_received

Delete Discourse API
URL: /admin/site_settings/tl3_requires_likes_received

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl3_requires_likes_received

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3RequiresLikesReceived. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTl3RequiresLikesReceived are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.23 putAdminSiteSettingsTl3RequiresMaxFlagged

tl3 requires max flagged

Catalog: Discourse

Schema: SiteSettingsTrustLevels
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tl3_requires_max_flagged`
Insert Discourse API URL: `/admin/site_settings/tl3_requires_max_flagged`
Update Discourse API URL: `/admin/site_settings/tl3_requires_max_flagged`
Delete Discourse API URL: `/admin/site_settings/tl3_requires_max_flagged`
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/tl3_requires_max_flagged

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl3RequiresMaxFlagged`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl3RequiresMaxFlagged` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.24 putAdminSiteSettingsTl3 RequiresPostsRead**

tl3 requires posts read
Catalog: Discourse
Schema: SiteSettingsTrustLevels
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/tl3_requires_posts_read`
Insert Discourse API URL: `/admin/site_settings/tl3_requires_posts_read`
Update Discourse API URL: `/admin/site_settings/tl3_requires_posts_read`
Delete Discourse API URL: `/admin/site_settings/tl3_requires_posts_read`

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/tl3_requires_posts_read

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl3RequiresPostsRead`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsTl3RequiresPostsRead` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.25 `putAdminSiteSettingsTl3RequiresPostsReadAllTime`

tl3 requires posts read all time
Catalog: Discourse
Schema: SiteSettingsTrustLevels
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/tl3_requires_posts_read_all_time`

Insert Discourse API
URL: `/admin/site_settings/tl3_requires_posts_read_all_time`

Update Discourse API
URL: `/admin/site_settings/tl3_requires_posts_read_all_time`

Delete Discourse API
URL: `/admin/site_settings/tl3_requires_posts_read_all_time`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/tl3_requires_posts_read_all_time`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl3RequiresPostsReadAllTime`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl3RequiresPostsReadAllTime` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
41.1.26 putAdminSiteSettingsTl3RequiresPostsReadCap

tl3 requires posts read cap

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl3_requires_posts_read_cap

Insert Discourse API
URL: /admin/site_settings/tl3_requires_posts_read_cap

Update Discourse API
URL: /admin/site_settings/tl3_requires_posts_read_cap

Delete Discourse API
URL: /admin/site_settings/tl3_requires_posts_read_cap

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl3_requires_posts_read_cap

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3RequiresPostsReadCap. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsTl3RequiresPostsReadCap are shown below. Each column has an SQL data type.
41.1.27 putAdminSiteSettingsTl3RequiresTopicsRepliedTo

tl3 requires topics replied to

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl3_requires_topics_replied_to

Insert Discourse API
URL: /admin/site_settings/tl3_requires_topics_replied_to

Update Discourse API
URL: /admin/site_settings/tl3_requires_topics_replied_to

Delete Discourse API
URL: /admin/site_settings/tl3_requires_topics_replied_to

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl3_requires_topics_replied_to

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3RequiresTopicsRepliedTo. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsTl3RequiresTopicsRepliedTo are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.28 putAdminSiteSettingsTl3RequiresTopicsViewed

tl3 requires topics viewed

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/tl3_requires_topics_viewed

Insert Discourse API URL: /admin/site_settings/tl3_requires_topics_viewed

Update Discourse API URL: /admin/site_settings/tl3_requires_topics_viewed

Delete Discourse API URL: /admin/site_settings/tl3_requires_topics_viewed

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/tl3_requires_topics_viewed

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3RequiresTopicsViewed. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsTl3RequiresTopicsViewed are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

41.1.29 putAdminSiteSettingsTl3RequiresTopicsViewedAllTime

tl3 requires topics viewed all time

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/tl3.requires_topics_viewed_all_time

Insert Discourse API
URL: /admin/site_settings/tl3.requires_topics_viewed_all_time

Update Discourse API
URL: /admin/site_settings/tl3.requires_topics_viewed_all_time

Delete Discourse API
URL: /admin/site_settings/tl3.requires_topics_viewed_all_time

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/tl3.requires_topics_viewed_all_time

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsTl3RequiresTopicsViewedAllTime. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsTl3RequiresTopicsViewedAllTime` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**41.1.30 putAdminSiteSettingsTl3RequiresTopicsViewedCap**

tl3 requires topics viewed cap

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/tl3RequiresTopicsViewedCap`

Insert Discourse API

URL: `/admin/site_settings/tl3RequiresTopicsViewedCap`

Update Discourse API

URL: `/admin/site_settings/tl3RequiresTopicsViewedCap`

Delete Discourse API

URL: `/admin/site_settings/tl3RequiresTopicsViewedCap`

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/tl3RequiresTopicsViewedCap`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl3RequiresTopicsViewedCap`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl3RequiresTopicsViewedCap` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**41.1.31 putAdminSiteSettingsTl3TimePeriod**

tl2 time period

Catalog: Discourse

Schema: SiteSettingsTrustLevels

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/tl3_time_period

Insert Discourse API URL: /admin/site_settings/tl3_time_period

Update Discourse API URL: /admin/site_settings/tl3_time_period

Delete Discourse API URL: /admin/site_settings/tl3_time_period

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/tl3_time_period

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsTl3TimePeriod`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsTl3TimePeriod` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 42 Schema: SiteSettingsUserAPI

#### 42.1 Tables

#### 42.1.1 putAdminSiteSettingsAllowedUserApiAuthRedirects

"allowed user api auth redirects"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/allowed_user_api_auth_redirects

Insert Discourse API

URL: /admin/site_settings/allowed_user_api_auth_redirects

Update Discourse API

URL: /admin/site_settings/allowed_user_api_auth_redirects

Delete Discourse API

URL: /admin/site_settings/allowed_user_api_auth_redirects

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/allowed_user_api_auth_redirects

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowedUserApiAuthRedirects`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAllowedUserApiAuthRedirects` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**42.1.2 putAdminSiteSettingsAllowedUserApiPushUrls**

"allowed user api push urls"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/allowed_user_api_push_urls

Insert Discourse API URL: /admin/site_settings/allowed_user_api_push_urls

Update Discourse API

URL: /admin/site_settings/allowed_user_api_push_urls

Delete Discourse API

URL: /admin/site_settings/allowed_user_api_push_urls

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/allowed_user_api_push_urls

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowedUserApiPushUrls`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAllowedUserApiPushUrls` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**42.1.3 putAdminSiteSettingsAllowUserApiKeys**

"allow user api keys"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/allow_user_api_keys`

Insert Discourse API URL: `/admin/site_settings/allow_user_api_keys`

Update Discourse API URL: `/admin/site_settings/allow_user_api_keys`

Delete Discourse API URL: `/admin/site_settings/allow_user_api_keys`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/allow_user_api_keys`

**Table Function Parameters**
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowUserApiKeys. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsAllowUserApiKeys` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 42.1.4 putAdminSiteSettingsAllowUserApiKeyScopes

"allow user api key scopes"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/allow_user_api_key_scopes`

Insert Discourse API URL: `/admin/site_settings/allow_user_api_key_scopes`

Update Discourse API URL: `/admin/site_settings/allow_user_api_key_scopes`

Delete Discourse API URL: `/admin/site_settings/allow_user_api_key_scopes`

Field Selection Method: NotRequired

Select Discourse API Operation: put `/admin/site_settings/allow_user_api_key_scopes`

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAllowUserApiKeyScopes. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function putAdminSiteSettingsAllowUserApiKeyScopes are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 42.1.5 putAdminSiteSettingsMaxUserApiReqsPerDay

"max user api reqs per day"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_user_api_reqs_per_day

Insert Discourse API URL: /admin/site_settings/max_user_api_reqs_per_day

Update Discourse API URL: /admin/site_settings/max_user_api_reqs_per_day

Delete Discourse API URL: /admin/site_settings/max_user_api_reqs_per_day

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/max_user_api_reqs_per_day

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxUserApiReqsPerDay`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMaxUserApiReqsPerDay` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**42.1.6 putAdminSiteSettingsMaxUserApiReqsPerMinute**

"max user api reqs per minute"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/max_user_api_reqs_per_minute

Insert Discourse API

URL: /admin/site_settings/max_user_api_reqs_per_minute

Update Discourse API

URL: /admin/site_settings/max_user_api_reqs_per_minute

Delete Discourse API

URL: /admin/site_settings/max_user_api_reqs_per_minute

Field Selection Method: NotRequired

Select Discourse API Operation:

`put /admin/site_settings/max_user_api_reqs_per_minute`
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxUserApiReqsPerMinute`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxUserApiReqsPerMinute` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

42.1.7 putAdminSiteSettingsMinTrustLevelForUserApiKey

"min trust level for user api key"

Catalog: Discourse

Schema: SiteSettingsUserAPI

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/min_trust_level_for_user_api_key

Insert Discourse API
URL: /admin/site_settings/min_trust_level_for_user_api_key

Update Discourse API
URL: /admin/site_settings/min_trust_level_for_user_api_key

Delete Discourse API
URL: /admin/site_settings/min_trust_level_for_user_api_key
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/min_trust_level_for_user_api_key

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsMinTrustLevelForUserApiKey. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsMinTrustLevelForUserApiKey are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**43 Schema: SiteSettingsUserPreferences**

**43.1 Tables**

**43.1.1 putAdminSiteSettingsDefaultCategoriesMuted**

"default categories muted"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/default_categories_muted
Insert Discourse API URL: /admin/site_settings/default_categories_muted
Update Discourse API URL: /admin/site_settings/default_categories_muted
Delete Discourse API URL: /admin/site_settings/default_categories_muted
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/default_categories_muted

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultCategoriesMuted. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultCategoriesMuted are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.2 putAdminSiteSettingsDefaultCategoriesTracking

"default categories tracking"
Catalog: Discourse
Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/default_categories_tracking
Insert Discourse API URL: /admin/site_settings/default_categories_tracking
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultCategoriesTracking`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultCategoriesTracking` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.3 putAdminSiteSettingsDefaultCategoriesWatching

"default categories watching"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/default_categories_watching

Insert Discourse API
URL: /admin/site_settings/default_categories_watching

Update Discourse API
URL: /admin/site_settings/default_categories_watching

Delete Discourse API
URL: /admin/site_settings/default_categories_watching

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_categories_watching

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultCategoriesWatching. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultCategoriesWatching are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.4 putAdminSiteSettingsDefaultCategoriesWatchingFirstPost

"default categories watching first post"

Catalog: Discourse

Schema: SiteSettingsUserPreferences
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_categories_watching_first_post

Insert Discourse API
URL: /admin/site_settings/default_categories_watching_first_post

Update Discourse API
URL: /admin/site_settings/default_categories_watching_first_post

Delete Discourse API
URL: /admin/site_settings/default_categories_watching_first_post

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_categories_watching_first_post

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDefaultCategoriesWatchingFirstPost. A value must be provided at all
times for required parameters, but optional parameters in general do not need to have a
value and the execution will default to a pre-defined behaviour. Values can be specified by
position and by name. In both cases, all parameters not specified will be treated using their
default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function
putAdminSiteSettingsDefaultCategoriesWatchingFirstPost are shown
below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
43.1.5 putAdminSiteSettingsDefaultEmailAlways
"default email always"
Catalog: Discourse
Schema: SiteSettingsUserPreferences
This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.
Select Discourse API URL: /admin/site_settings/default_email_always
Insert Discourse API URL: /admin/site_settings/default_email_always
Update Discourse API URL: /admin/site_settings/default_email_always
Delete Discourse API URL: /admin/site_settings/default_email_always
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/default_email_always

Table Function Parameters
The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDefaultEmailAlways. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.
Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function putAdminSiteSettingsDefaultEmailAlways are
shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
43.1.6 putAdminSiteSettingsDefaultEmailDigestFrequency

"default email digest frequency"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_email_digest_frequency

Insert Discourse API
URL: /admin/site_settings/default_email_digest_frequency

Update Discourse API
URL: /admin/site_settings/default_email_digest_frequency

Delete Discourse API
URL: /admin/site_settings/default_email_digest_frequency

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_email_digest_frequency

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultEmailDigestFrequency. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultEmailDigestFrequency are shown below. Each column has an SQL data type.
43.1.7 putAdminSiteSettingsDefaultEmailDirect

"default email direct"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/default_email_direct

Insert Discourse API URL: /admin/site_settings/default_email_direct

Update Discourse API URL: /admin/site_settings/default_email_direct

Delete Discourse API URL: /admin/site_settings/default_email_direct

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/default_email_direct

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultEmailDirect. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultEmailDirect are shown below. Each column has an SQL data type.
43.1.8 putAdminSiteSettingsDefaultEmailInReplyTo

"default email in reply to"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/default_email_in_reply_to

Insert Discourse API URL: /admin/site_settings/default_email_in_reply_to

Update Discourse API URL: /admin/site_settings/default_email_in_reply_to

Delete Discourse API URL: /admin/site_settings/default_email_in_reply_to

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_email_in_reply_to

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultEmailInReplyTo. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultEmailInReplyTo are shown below. Each column has an SQL data type.
43.1.9 putAdminSiteSettingsDefaultEmailMailingListMode

"default email mailing list mode"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_email_mailing_list_mode

Insert Discourse API
URL: /admin/site_settings/default_email_mailing_list_mode

Update Discourse API
URL: /admin/site_settings/default_email_mailing_list_mode

Delete Discourse API
URL: /admin/site_settings/default_email_mailing_list_mode

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_email_mailing_list_mode

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultEmailMailingListMode. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns
The columns of the table function `putAdminSiteSettingsDefaultEmailMailingListMode` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.10 `putAdminSiteSettingsDefaultEmailMailingListModeFrequency`

"default email mailing list mode frequency"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/default_email_mailing_list_mode_frequency`

Insert Discourse API URL: `/admin/site_settings/default_email_mailing_list_mode_frequency`

Update Discourse API URL: `/admin/site_settings/default_email_mailing_list_mode_frequency`

Delete Discourse API URL: `/admin/site_settings/default_email_mailing_list_mode_frequency`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/default_email_mailing_list_mode_frequency`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultEmailMailingListModeFrequency`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultEmailMailingListModeFrequency are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.11 putAdminSiteSettingsDefaultEmailPreviousReplies
"default email previous replies"

Catalog: Discourse
Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_email_previous_replies

Insert Discourse API
URL: /admin/site_settings/default_email_previous_replies

Update Discourse API
URL: /admin/site_settings/default_email_previous_replies

Delete Discourse API
URL: /admin/site_settings/default_email_previous_replies

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_email_previous_replies

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultEmailPreviousReplies. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
### Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultEmailPreviousReplies` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>[ ]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 43.1.12 putAdminSiteSettingsDefaultEmailPrivateMessages

"default email private messages"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/default_email_private_messages`

Insert Discourse API

URL: `/admin/site_settings/default_email_private_messages`

Update Discourse API

URL: `/admin/site_settings/default_email_private_messages`

Delete Discourse API

URL: `/admin/site_settings/default_email_private_messages`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/default_email_private_messages

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultEmailPrivateMessages`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultEmailPrivateMessages` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 43.1.13 putAdminSiteSettingsDefaultIncludeTl0InDigests

"default include tl0 in digests"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_include_tl0_in_digests

Insert Discourse API
URL: /admin/site_settings/default_include_tl0_in_digests

Update Discourse API
URL: /admin/site_settings/default_include_tl0_in_digests

Delete Discourse API
URL: /admin/site_settings/default_include_tl0_in_digests

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_include_tl0_in_digests

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultIncludeTl0InDigests`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultIncludeTl0InDigests` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

43.1.14 putAdminSiteSettingsDefaultOtherAutoTrackTopicAfterMsecs

"default other auto track topics after msecs"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/default_other_auto_track_topic_after_msecs

Insert Discourse API

URL: /admin/site_settings/default_other_auto_track_topic_after_msecs

Update Discourse API

URL: /admin/site_settings/default_other_auto_track_topic_after_msecs

Delete Discourse API

URL: /admin/site_settings/default_other_auto_track_topic_after_msecs

Field Selection Method: NotRequired
Select Discourse API Operation:

```
put /admin/site_settings/default_other_auto_track_topic_after_msecs
```

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultOtherAutoTrackTopicAfterMsecs`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsDefaultOtherAutoTrackTopicAfterMsecs` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**43.1.15 putAdminSiteSettingsDefaultOtherDisableJumpReply**

"default other disable jump reply"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: `/admin/site_settings/default_other_disable_jump_reply`

Insert Discourse API

URL: `/admin/site_settings/default_other_disable_jump_reply`
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultOtherDisableJumpReply`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultOtherDisableJumpReply` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.16 putAdminSiteSettingsDefaultOtherDynamicFavicon

default other dynamic favicon

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API
URL: /admin/site_settings/default_other_dynamic_favicon

Insert Discourse API
URL: /admin/site_settings/default_other_dynamic_favicon

Update Discourse API
URL: /admin/site_settings/default_other_dynamic_favicon

Delete Discourse API
URL: /admin/site_settings/default_other_dynamic_favicon

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_other_dynamic_favicon

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsDefaultOtherDynamicFavicon. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function
putAdminSiteSettingsDefaultOtherDynamicFavicon are shown below. Each
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.17 putAdminSiteSettingsDefaultOtherEnableQuoting

"default other enable quoting"

Catalog: Discourse

Schema: SiteSettingsUserPreferences
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_other_enable_quoting

Insert Discourse API
URL: /admin/site_settings/default_other_enable_quoting

Update Discourse API
URL: /admin/site_settings/default_other_enable_quoting

Delete Discourse API
URL: /admin/site_settings/default_other_enable_quoting

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_other_enable_quoting

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultOtherEnableQuoting. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultOtherEnableQuoting are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
43.1.18 putAdminSiteSettingsDefaultOtherExternalLinksInNewTab
"default other external links in new tab"

Catalog: Discourse
Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table \texttt{NativePlatformScalarRequests} to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_other_external_links_in_new_tab

Insert Discourse API
URL: /admin/site_settings/default_other_external_links_in_new_tab

Update Discourse API
URL: /admin/site_settings/default_other_external_links_in_new_tab

Delete Discourse API
URL: /admin/site_settings/default_other_external_links_in_new_tab

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/default_other_external_links_in_new_tab

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDefaultOtherExternalLinksInNewTab. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with \texttt{select * from table(value1, value2, value3)} on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with \texttt{select * from table(name1 => value1, name3 => value3)} on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDefaultOtherExternalLinksInNewTab are shown below. Each column has an SQL data type.
**43.1.19 putAdminSiteSettingsDefaultOtherLikeNotificationFrequency**

"default other like notification frequency"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/default_other_like_notification_frequency`

Insert Discourse API
URL: `/admin/site_settings/default_other_like_notification_frequency`

Update Discourse API
URL: `/admin/site_settings/default_other_like_notification_frequency`

Delete Discourse API
URL: `/admin/site_settings/default_other_like_notification_frequency`

Field Selection Method: NotRequired

Select Discourse API Operation:
put `/admin/site_settings/default_other_like_notification_frequency`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultOtherLikeNotificationFrequency`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultOtherLikeNotificationFrequency` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.20 `putAdminSiteSettingsDefaultOtherNewTopicDurationMinutes`

"default other new topic duration minutes"

Catalog: Discourse
Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_other_new_topic_duration_minutes

Insert Discourse API
URL: /admin/site_settings/default_other_new_topic_duration_minutes

Update Discourse API
URL: /admin/site_settings/default_other_new_topic_duration_minutes

Delete Discourse API
URL: /admin/site_settings/default_other_new_topic_duration_minutes

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_other_new_topic_duration_minutes

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultOtherNewTopicDurationMinutes`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultOtherNewTopicDurationMinutes` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 43.1.21 putAdminSiteSettingsDefaultOtherNotificationLevelWhenReplying

"default other notification level when replying"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: `/admin/site_settings/default_other_notification_level_when_reponding`

Insert Discourse API
URL: `/admin/site_settings/default_other_notification_level_when_reponding`

Update Discourse API
URL: `/admin/site_settings/default_other_notification_level_when_reponding`

Delete Discourse API
URL: `/admin/site_settings/default_other_notification_level_when_reponding`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/default_other_notification_level_when_reponding`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultOtherNotificationLevelWhenReplying`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.
Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsDefaultOtherNotificationLevelWhenReplying` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**43.1.22 putAdminSiteSettingsDefaultTopicsAutomaticUnpin**

"default topics automatic unpin"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/default_topics_automatic_unpin

Insert Discourse API
URL: /admin/site_settings/default_topics_automatic_unpin

Update Discourse API
URL: /admin/site_settings/default_topics_automatic_unpin

Delete Discourse API
URL: /admin/site_settings/default_topics_automatic_unpin

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/default_topics_automatic_unpin

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDefaultTopicsAutomaticUnpin`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDefaultTopicsAutomaticUnpin` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

43.1.23 putAdminSiteSettingsDisableMailingListMode

"disable mailing list mode"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/disable_mailing_list_mode

Insert Discourse API URL: /admin/site_settings/disable_mailing_list_mode

Update Discourse API URL: /admin/site_settings/disable_mailing_list_mode

Delete Discourse API URL: /admin/site_settings/disable_mailing_list_mode

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/disable_mailing_list_mode

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDisableMailingListMode`. A value must be provided at all times for
required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsDisableMailingListMode` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 43.1.24 putAdminSiteSettingsMaxApiKeysPerUser

"max api keys per user"

Catalog: Discourse

Schema: SiteSettingsUserPreferences

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/max_api_keys_per_user`

Insert Discourse API URL: `/admin/site_settings/max_api_keys_per_user`

Update Discourse API URL: `/admin/site_settings/max_api_keys_per_user`

Delete Discourse API URL: `/admin/site_settings/max_api_keys_per_user`

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/max_api_keys_per_user

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxApiKeysPerUser`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxApiKeysPerUser` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Parameters**

44 Schema: SiteSettingsUsers

44.1 Tables

44.1.1 putAdminSiteSettingsAllowAnonymousPosting

allow anonymous posting

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/allow_anonymousPosting`

Insert Discourse API URL: `/admin/site_settings/allow_anonymousPosting`

Update Discourse API URL: `/admin/site_settings/allow_anonymousPosting`

Delete Discourse API URL: `/admin/site_settings/allow_anonymousPosting`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/allow_anonymousPosting`

**Table Function Parameters**

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsAllowAnonymousPosting`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsAllowAnonymousPosting` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.2 putAdminSiteSettingsAnonymousAccountDurationMinutes
anonymous account duration minutes

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/anonymous_account_duration_minutes

Insert Discourse API
URL: /admin/site_settings/anonymous_account_duration_minutes

Update Discourse API
URL: /admin/site_settings/anonymous_account_duration_minutes

Delete Discourse API
URL: /admin/site_settings/anonymous_account_duration_minutes

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/anonymous_account_duration_minutes
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsAnonymousAccountDurationMinutes. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsAnonymousAccountDurationMinutes are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.3 putAdminSiteSettingsAnonymousPostingMinTrustLevel

anonymous posting min trust level

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/anonymousPostingMinTrustLevel

Insert Discourse API
URL: /admin/site_settings/anonymousPostingMinTrustLevel

Update Discourse API
URL: /admin/site_settings/anonymousPostingMinTrustLevel

Delete Discourse API
URL: /admin/site_settings/anonymousPostingMinTrustLevel
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/anonymous_posting_min_trust_level

Table Function Parameters

The following parameters can be used to control the behaviour of the table function
putAdminSiteSettingsAnonymousPostingMinTrustLevel. A value must be provided at all times
for required parameters, but optional parameters in general do not need to have a value and
the execution will default to a pre-defined behaviour. Values can be specified by position and
by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function
putAdminSiteSettingsAnonymousPostingMinTrustLevel are shown below. Each
column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.4 putAdminSiteSettingsBlockCommonPasswords

block common passwords

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/block_common_passwords

Insert Discourse API URL: /admin/site_settings/block_common_passwords

Update Discourse API URL: /admin/site_settings/block_common_passwords

Delete Discourse API URL: /admin/site_settings/block_common_passwords
Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/block_common_passwords

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsBlockCommonPasswords. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsBlockCommonPasswords are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.5 putAdminSiteSettingsDeleteAllPostsMax
delete all posts max

Catalog: Discourse
Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/delete_all_posts_max
Insert Discourse API URL: /admin/site_settings/delete_all_posts_max
Update Discourse API URL: /admin/site_settings/delete_all_posts_max
Delete Discourse API URL: /admin/site_settings/delete_all_posts_max
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/delete_all_posts_max

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsDeleteAllPostsMax`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsDeleteAllPostsMax` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.6 `putAdminSiteSettingsDeleteUserMaxPostAge`
delete user max post age

Catalog: Discourse
Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/delete_user_max_post_age
Insert Discourse API URL: /admin/site_settings/delete_user_max_post_age
Update Discourse API URL: /admin/site_settings/delete_user_max_post_age
Delete Discourse API URL: /admin/site_settings/delete_user_max_post_age
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/delete_user_max_post_age

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsDeleteUserMaxPostAge. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsDeleteUserMaxPostAge are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.7 putAdminSiteSettingsEmailEditable

e-mail editable

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_editable
Insert Discourse API URL: /admin/site_settings/email_editable
Update Discourse API URL: /admin/site_settings/email_editable
Delete Discourse API URL: /admin/site_settings/email_editable
Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/email_editable
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEmailEditable`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsEmailEditable` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.8 `putAdminSiteSettingsEmailTokenValidHours`

e-mail token valid hours

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/email_token_valid_hours

Insert Discourse API URL: /admin/site_settings/email_token_valid_hours

Update Discourse API URL: /admin/site_settings/email_token_valid_hours

Delete Discourse API URL: /admin/site_settings/email_token_valid_hours

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/email_token_valid_hours
Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsEmailTokenValidHours. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsEmailTokenValidHours are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.9 putAdminSiteSettingsEnableNames

enable names

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/enable_names

Insert Discourse API URL: /admin/site_settings/enable_names

Update Discourse API URL: /admin/site_settings/enable_names

Delete Discourse API URL: /admin/site_settings/enable_names

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/enable_names

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableNames`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>√</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableNames` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>√</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 44.1.10 putAdminSiteSettingsEnableUserDirectory

enable user directory

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/enable_user_directory`

Insert Discourse API URL: `/admin/site_settings/enable_user_directory`

Update Discourse API URL: `/admin/site_settings/enable_user_directory`

Delete Discourse API URL: `/admin/site_settings/enable_user_directory`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /admin/site_settings/enable_user_directory`

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsEnableUserDirectory`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsEnableUserDirectory` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.11 putAdminSiteSettingsFullNameRequired

full name required

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/full_name_required

Insert Discourse API URL: /admin/site_settings/full_name_required

Update Discourse API URL: /admin/site_settings/full_name_required

Delete Discourse API URL: /admin/site_settings/full_name_required

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/full_name_required

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsFullNameRequired`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsFullNameRequired` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 44.1.12 putAdminSiteSettingsHideUserProfilesFromPublic

hide user profiles from public

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/hide_user_profiles_from_public

Insert Discourse API

URL: /admin/site_settings/hide_user_profiles_from_public

Update Discourse API

URL: /admin/site_settings/hide_user_profiles_from_public

Delete Discourse API

URL: /admin/site_settings/hide_user_profiles_from_public

Field Selection Method: NotRequired

Select Discourse API Operation:

put /admin/site_settings/hide_user_profiles_from_public
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsHideUserProfilesFromPublic`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsHideUserProfilesFromPublic` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.13 putAdminSiteSettingsInviteExpireDays

invite expiry days
Catalog: Discourse
Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/invite_expiry_days`
Insert Discourse API URL: `/admin/site_settings/invite_expiry_days`
Update Discourse API URL: `/admin/site_settings/invite_expiry_days`
Delete Discourse API URL: `/admin/site_settings/invite_expiry_days`
Field Selection Method: NotRequired
Select Discourse API Operation:
put `/admin/site_settings/invite_expiry_days`
**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsInviteExpiryDays. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function putAdminSiteSettingsInviteExpiryDays are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**44.1.14 putAdminSiteSettingsInvitePassthroughHours**

invite passthrough hours

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/invite_passthrough_hours`

Insert Discourse API URL: `/admin/site_settings/invite_passthrough_hours`

Update Discourse API URL: `/admin/site_settings/invite_passthrough_hours`

Delete Discourse API URL: `/admin/site_settings/invite_passthrough_hours`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/invite_passthrough_hours`
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsInvitePassthroughHours`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsInvitePassthroughHours` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.15 `putAdminSiteSettingsInvitesPerPage`

invites per page

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/invites_per_page

Insert Discourse API URL: /admin/site_settings/invites_per_page

Update Discourse API URL: /admin/site_settings/invites_per_page

Delete Discourse API URL: /admin/site_settings/invites_per_page

Field Selection Method: NotRequired

Select Discourse API Operation: put /admin/site_settings/invites_per_page

Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsInvitesPerPage`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsInvitesPerPage` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 44.1.16 putAdminSiteSettingsLogoutRedirect

**logout redirect**

**Catalog:** Discourse

**Schema:** SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /admin/site_settings/logout_redirect

**Insert Discourse API URL:** /admin/site_settings/logout_redirect

**Update Discourse API URL:** /admin/site_settings/logout_redirect

**Delete Discourse API URL:** /admin/site_settings/logout_redirect

**Field Selection Method:** NotRequired

**Select Discourse API Operation:** put /admin/site_settings/logout_redirect

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsLogoutRedirect`. A value must be provided at all times for required
parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>☐</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsLogoutRedirect` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>☐</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 44.1.17 putAdminSiteSettingsMaxUsernameLength

<table>
<thead>
<tr>
<th>max username length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog: Discourse</td>
</tr>
<tr>
<td>Schema: SiteSettingsUsers</td>
</tr>
</tbody>
</table>

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/max_username_length
Insert Discourse API URL: /admin/site_settings/max_username_length
Update Discourse API URL: /admin/site_settings/max_username_length
Delete Discourse API URL: /admin/site_settings/max_username_length
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/max_username_length

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMaxUsernameLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
Execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMaxUsernameLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 44.1.18 putAdminSiteSettingsMinAdminPasswordLength

- **min admin password length**
- **Catalog**: Discourse
- **Schema**: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

- **Select Discourse API URL**: `/admin/site_settings/min_admin_password_length`
- **Insert Discourse API URL**: `/admin/site_settings/min_admin_password_length`
- **Update Discourse API URL**: `/admin/site_settings/min_admin_password_length`
- **Delete Discourse API URL**: `/admin/site_settings/min_admin_password_length`

- **Field Selection Method**: NotRequired
- **Select Discourse API Operation**: `put /admin/site_settings/min_admin_password_length`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinAdminPasswordLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsMinAdminPasswordLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.19 `putAdminSiteSettingsMinPasswordLength`

min password length

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_password_length`

Insert Discourse API URL: `/admin/site_settings/min_password_length`

Update Discourse API URL: `/admin/site_settings/min_password_length`

Delete Discourse API URL: `/admin/site_settings/min_password_length`

Field Selection Method: NotRequired

Select Discourse API Operation:

put `/admin/site_settings/min_password_length`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinPasswordLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsMinPasswordLength` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td>□</td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 44.1.20 putAdminSiteSettingsMinUsernameLength

min username length

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/min_username_length`

Insert Discourse API URL: `/admin/site_settings/min_username_length`

Update Discourse API URL: `/admin/site_settings/min_username_length`

Delete Discourse API URL: `/admin/site_settings/min_username_length`

Field Selection Method: NotRequired

Select Discourse API Operation:
`put /admin/site_settings/min_username_length`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsMinUsernameLength`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function putAdminSiteSettingsMinUsernameLength are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.21 putAdminSiteSettingsPrioritizeUsernameInUx

prioritize username in ux

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/prioritize_username_in_ux

Insert Discourse API URL: /admin/site_settings/prioritize_username_in_ux

Update Discourse API URL: /admin/site_settings/prioritize_username_in_ux

Delete Discourse API URL: /admin/site_settings/prioritize_username_in_ux

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/prioritize_username_in_ux

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsPrioritizeUsernameInUx. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `putAdminSiteSettingsPrioritizeUsernameInUx` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.22 putAdminSiteSettingsPublicUserCustomFields

public user custom fields
Catalog: Discourse
Schema: SiteSettingsUsers
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/public_user_custom_fields
Insert Discourse API URL: /admin/site_settings/public_user_custom_fields
Update Discourse API URL: /admin/site_settings/public_user_custom_fields
Delete Discourse API URL: /admin/site_settings/public_user_custom_fields
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/public_user_custom_fields

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPublicUserCustomFields`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by
name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed
value. For example with `select * from table(value1, value2, value3)` on a table with four
parameters will use the default value for the fourth parameter and the specified values for the
first three.

Value specification by name is done by listing all values that require a value. For example
with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and
third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putAdminSiteSettingsPublicUserCustomFields` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 44.1.23 putAdminSiteSettingsPurgeUnactivatedUsersGracePeriodDays

purge unactivated users grace period days

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or
the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the
`table NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API
URL: /admin/site_settings/purge_unactivated_users_grace_period_days

Insert Discourse API
URL: /admin/site_settings/purge_unactivated_users_grace_period_days

Update Discourse API
URL: /admin/site_settings/purge_unactivated_users_grace_period_days

Delete Discourse API
URL: /admin/site_settings/purge_unactivated_users_grace_period_days

Field Selection Method: NotRequired

Select Discourse API Operation:
put /admin/site_settings/purge_unactivated_users_grace_period_days

### Table Function Parameters
The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsPurgeUnactivatedUsersGracePeriodDays`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsPurgeUnactivatedUsersGracePeriodDays` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

### 44.1.24 putAdminSiteSettingsRedirectUsersToTopPage

redirect users to top page

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API

URL: /admin/site_settings/redirect_users_to_top_page

Insert Discourse API URL: /admin/site_settings/redirect_users_to_top_page

Update Discourse API

URL: /admin/site_settings/redirect_users_to_top_page

Delete Discourse API

URL: /admin/site_settings/redirect_users_to_top_page

Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/redirect_users_to_top_page

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsRedirectUsersToTopPage. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsRedirectUsersToTopPage are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.25 putAdminSiteSettingsReservedUsernames

reserved usernames

Catalog: Discourse
Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/reserved_usernames
Insert Discourse API URL: /admin/site_settings/reserved_usernames
Update Discourse API URL: /admin/site_settings/reserved_usernames
Delete Discourse API URL: /admin/site_settings/reserved_usernames
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/reserved_usernames

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsReservedUsernames. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsReservedUsernames are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

44.1.26 putAdminSiteSettingsShowEmailOnProfile

show email on profile

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/show_email_on_profile
Insert Discourse API URL: /admin/site_settings/show_email_on_profile
Update Discourse API URL: /admin/site_settings/show_email_on_profile
Delete Discourse API URL: /admin/site_settings/show_email_on_profile
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/show_email_on_profile

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsShowEmailOnProfile`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsShowEmailOnProfile` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**44.1.27 putAdminSiteSettingsStaffUsersCustomFields**

staff user custom fields

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /admin/site_settings/staff_users_custom_fields
Insert Discourse API URL: /admin/site_settings/staff_users_custom_fields
Update Discourse API URL: /admin/site_settings/staff_users_custom_fields
Delete Discourse API URL: /admin/site_settings/staff_users_custom_fields
Field Selection Method: NotRequired
Select Discourse API Operation:
`put /admin/site_settings/staff_users_custom_fields`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putAdminSiteSettingsStaffUsersCustomFields`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putAdminSiteSettingsStaffUsersCustomFields` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

**44.1.28 putAdminSiteSettingsUsernameChangePeriod**

username change period

Catalog: Discourse

Schema: SiteSettingsUsers

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/site_settings/username_change_period`
Insert Discourse API URL: `/admin/site_settings/username_change_period`
Update Discourse API URL: `/admin/site_settings/username_change_period`
Delete Discourse API URL: `/admin/site_settings/username_change_period`
Field Selection Method: NotRequired
Select Discourse API Operation:
put /admin/site_settings/username_change_period

Table Function Parameters

The following parameters can be used to control the behaviour of the table function putAdminSiteSettingsUsernameChangePeriod. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function putAdminSiteSettingsUsernameChangePeriod are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

45 Schema: Tags

45.1 Tables

45.1.1 TagGroups: Discourse Tag Groups

Get a list of tag groups

Catalog: Discourse
Schema: Tags
Label: Tag Groups
Retrieve: true
Insert: true
Update: false
Delete: false

Select Discourse API URL: /tag_groups.json
Insert Discourse API URL: /tag_groups.json
Update Discourse API URL: /tag_groups.json
Delete Discourse API URL: /tag_groups.json
Field Selection Method: NotRequired
Base Path: tag_groups[*]
Select Discourse API Operation: get /tag_groups.json
Insert Discourse API Operation: post /tag_groups.json

**Table Columns**

The columns of the table TagGroups are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>one_per_topic</td>
<td>boolean</td>
<td>One per Topic</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>parent_tag_name</td>
<td>string</td>
<td>Parent Tag Name</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**45.1.2 TagGroups_Tag_groupParent_tag_nameById**

Get a single tag group
Catalog: Discourse
Schema: Tags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /tag_groups/{id}.json
Insert Discourse API URL: /tag_groups/{id}.json
Update Discourse API URL: /tag_groups/{id}.json
Delete Discourse API URL: /tag_groups/{id}.json
Field Selection Method: NotRequired
Base Path: tag_group.parent_tag_name[*]
Select Discourse API Operation: get /tag_groups/{id}.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function TagGroups_Tag_groupParent_tag_nameById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the
execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `TagGroups_Tag_groupParent_tag_nameById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>one_per_topic</td>
<td>boolean</td>
<td>One per Topic</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>tag_name</td>
<td>string</td>
<td>Tag Name</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

#### 45.1.3 TagGroups_Tag_groupsTag_names: Discourse Tag Group Tags

Get a list of tag groups
Catalog: Discourse
Schema: Tags
Label: Tag Group Tags

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /tag_groups.json
Insert Discourse API URL: /tag_groups.json
Update Discourse API URL: /tag_groups.json
Delete Discourse API URL: /tag_groups.json
Field Selection Method: NotRequired
Base Path: tag_groups[*].tag_names[*]
Select Discourse API Operation: get /tag_groups.json
Table Columns

The columns of the table TagGroups_Tag_groupsTag_names are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>one_per_topic</td>
<td>boolean</td>
<td>One per Topic</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>parent_tag_name</td>
<td>string</td>
<td>Parent Tag Name</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

45.1.4 TagGroups_Tag_groupTag_namesById

Get a single tag group

Catalog: Discourse

Schema: Tags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /tag_groups/{id}.json

Insert Discourse API URL: /tag_groups/{id}.json

Update Discourse API URL: /tag_groups/{id}.json

Delete Discourse API URL: /tag_groups/{id}.json

Field Selection Method: NotRequired

Base Path: tag_group.tag_names[*]

Select Discourse API Operation: get /tag_groups/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function TagGroups_Tag_groupTag_namesById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function TagGroups_Tag_groupTag_namesById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>one_per_topic</td>
<td>boolean</td>
<td>One per Topic</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

45.1.5 TagGroupsById

Get a single tag group
Catalog: Discourse
Schema: Tags
Retrieve: true
Insert: false
Update: true
Delete: false
Select Discourse API URL: /tag_groups/{id}.json
Insert Discourse API URL: /tag_groups/{id}.json
Update Discourse API URL: /tag_groups/{id}.json
Delete Discourse API URL: /tag_groups/{id}.json
Field Selection Method: NotRequired
Base Path: tag_group
Select Discourse API Operation: get /tag_groups/{id}.json
Update Discourse API Operation: put /tag_groups/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function TagGroupsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function TagGroupsById are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during update.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>one_per_topic</td>
<td>boolean</td>
<td>One per Topic</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

### 45.1.6 Tags: Discourse Tags

Get a list of tags

Catalog: Discourse

Schema: Tags

Label: Tags

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /tags

Insert Discourse API URL: /tags

Update Discourse API URL: /tags

Delete Discourse API URL: /tags

Field Selection Method: NotRequired

Base Path: tags[*]

Select Discourse API Operation: get /tags

### Table Columns

The columns of the table Tags are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>decimal</td>
<td>Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pm_count</td>
<td>decimal</td>
<td>PM Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>target_tag</td>
<td>string</td>
<td>Target Tag</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
45.1.7 TagsByTag

Get a specific tag

Catalog: Discourse

Schema: Tags

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/tags/{tag}`

Insert Discourse API URL: `/tags/{tag}`

Update Discourse API URL: `/tags/{tag}`

Delete Discourse API URL: `/tags/{tag}`

Field Selection Method: NotRequired

Select Discourse API Operation: `get /tags/{tag}`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `TagsByTag`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `TagsByTag` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single</td>
</tr>
</tbody>
</table>
### 46 Schema: Topics

#### 46.1 Tables

**46.1.1 Latest**

Get the latest topics

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/latest.json`

Insert Discourse API URL: `/latest.json`

Update Discourse API URL: `/latest.json`

Delete Discourse API URL: `/latest.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `get /latest.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function Latest. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default value for the fourth parameter and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascending</td>
<td>string</td>
<td>□</td>
<td>defaults to desc, add <code>ascending=true</code> to sort asc</td>
<td></td>
</tr>
<tr>
<td>order</td>
<td>string</td>
<td>□</td>
<td>Values: default, created, activity, views, posts, category, likes, op_likes, posters</td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `Latest` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>int64</td>
<td>Topic List Per Page</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

46.1.2 Latest_Topic_listTopics

Get the latest topics

Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/latest.json`
Insert Discourse API URL: `/latest.json`
Update Discourse API URL: `/latest.json`
Delete Discourse API URL: `/latest.json`
Field Selection Method: NotRequired
Base Path: `topic_list.topics[*]`
Select Discourse API Operation: get `/latest.json`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `Latest_Topic_listTopics`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `Latest_Topic_listTopics` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>int64</td>
<td>Highest Post Number</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>int64</td>
<td>#Likes</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>int64</td>
<td>Per Page</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>int64</td>
<td>#Posts</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>int64</td>
<td>#Replies</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>int64</td>
<td>Views</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
46.1.3 Latest_Topic_listTopicsPosters

Get the latest topics

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /latest.json

Insert Discourse API URL: /latest.json

Update Discourse API URL: /latest.json

Delete Discourse API URL: /latest.json

Field Selection Method: NotRequired

Base Path: topic_list.topics[*].posters[*]

Select Discourse API Operation: get /latest.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function Latest_Topic_listTopicsPosters. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascending</td>
<td>string</td>
<td>☐</td>
<td></td>
<td>defaults to desc, add <code>ascending=true</code> to sort asc</td>
</tr>
<tr>
<td>order</td>
<td>string</td>
<td>☐</td>
<td></td>
<td>Values: default, created, activity, views, posts, category, likes, op_likes, posters.</td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function Latest_Topic_listTopicsPosters are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extras</td>
<td>string</td>
<td>Extras</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>int64</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>int64</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>int64</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>int64</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>int64</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>int64</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>int64</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**46.1.4 Latest_Users**

Get the latest topics

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: /latest.json

Insert Discourse API URL: /latest.json
Table Function Parameters

The following parameters can be used to control the behaviour of the table function `Latest_Users`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascending</td>
<td>string</td>
<td>☐</td>
<td>defaults to desc, add <code>ascending=true</code> to sort asc</td>
<td></td>
</tr>
<tr>
<td>order</td>
<td>string</td>
<td>☐</td>
<td>Values: default, created, activity, views, posts, category, likes, op_likes, posters.</td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `Latest_Users` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>int64</td>
<td>Topic List Per Page</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>
46.1.5 posttByIdNotifications

set notification level
Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}/notifications
Insert Discourse API URL: /t/{id}/notifications
Update Discourse API URL: /t/{id}/notifications
Delete Discourse API URL: /t/{id}/notifications
Field Selection Method: NotRequired
Base Path: success
Select Discourse API Operation: post /t/{id}/notifications

Table Function Parameters

The following parameters can be used to control the behaviour of the table function posttByIdNotifications. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function posttByIdNotifications are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
46.1.6 postByIdTimer

Create a timed topic

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}/timer

Insert Discourse API URL: /t/{id}/timer

Update Discourse API URL: /t/{id}/timer

Delete Discourse API URL: /t/{id}/timer

Field Selection Method: NotRequired

Select Discourse API Operation: post /t/{id}/timer

Table Function Parameters

The following parameters can be used to control the behaviour of the table function postByIdTimer. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function postByIdTimer are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>based_on_last_post</td>
<td>boolean</td>
<td>Based on Last Post</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>duration</td>
<td>string</td>
<td>Duration</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>execute_at</td>
<td>string</td>
<td>Execute at</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>
### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `puttByIdBookmark`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `puttByIdBookmark` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single</td>
</tr>
</tbody>
</table>
Discourse Data Model

46.1.8 `putByIdChangeTimestamp`

Update a topics timestamp

Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}/change-timestamp`
Insert Discourse API URL: `/t/{id}/change-timestamp`
Update Discourse API URL: `/t/{id}/change-timestamp`
Delete Discourse API URL: `/t/{id}/change-timestamp`

Field Selection Method: NotRequired

Base Path: `success`

Select Discourse API Operation: `put /t/{id}/change-timestamp`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putByIdChangeTimestamp`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `putByIdChangeTimestamp` are shown below. Each column has an SQL data type.
### Table: puttByIdStatus

Update a topic's status

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}/status`

Insert Discourse API URL: `/t/{id}/status`

Update Discourse API URL: `/t/{id}/status`

Delete Discourse API URL: `/t/{id}/status`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /t/{id}/status`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `puttByIdStatus`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `puttByIdStatus` are shown below. Each column has an SQL data type.
### puttBySlugId

**Update a single topic**

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table [NativePlatformScalarRequests](#) to upload data to the Discourse API.

Select Discourse API URL: `/t/{slug}/{id}.json`

Insert Discourse API URL: `/t/{slug}/{id}.json`

Update Discourse API URL: `/t/{slug}/{id}.json`

Delete Discourse API URL: `/t/{slug}/{id}.json`

Field Selection Method: NotRequired

Base Path: `basic_topic`

Select Discourse API Operation: `put /t/{slug}/{id}.json`

#### Table Function Parameters

The following parameters can be used to control the behaviour of the table function puttBySlugId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td></td>
<td></td>
<td>ProTip: You can just use a <code>~</code> instead of the actual slug and the topic will be found by the id.</td>
</tr>
</tbody>
</table>

#### Table Function Columns

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The columns of the table function `putBySlugId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

### 46.1.11 t_Actions_summaryById

Get a single topic

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}.json`

Insert Discourse API URL: `/t/{id}.json`

Update Discourse API URL: `/t/{id}.json`

Delete Discourse API URL: `/t/{id}.json`

Field Selection Method: NotRequired

Base Path: `actions_summary[*]`

Select Discourse API Operation: `get /t/{id}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `t_Actions_summaryById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `t_Actions_summaryById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>can_act</td>
<td>boolean</td>
<td>Can Act</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>count</td>
<td>decimal</td>
<td>Count</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>details_auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Detail Auto Close Based On Last Post</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_can_flag_topic</td>
<td>boolean</td>
<td>Detail Can Flag Topic</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_created_by_avatar_template</td>
<td>string</td>
<td>Detail Created by Avatar Template</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_created_by_id</td>
<td>decimal</td>
<td>Detail Created by ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_created_by_username</td>
<td>string</td>
<td>Detail Created by Username</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_last_poster_avatar_template</td>
<td>string</td>
<td>Detail Last Poster Avatar Template</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_last_poster_id</td>
<td>decimal</td>
<td>Detail Last Poster ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_last_poster_username</td>
<td>string</td>
<td>Detail Last Poster User Name</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_notification_level</td>
<td>decimal</td>
<td>Detail Notification Level</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>☑</td>
<td></td>
</tr>
</tbody>
</table>
### t_DetailsParticipantsById

Get a single topic

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}.json`

Insert Discourse API URL: `/t/{id}.json`

Update Discourse API URL: `/t/{id}.json`

Delete Discourse API URL: `/t/{id}.json`

Field Selection Method: NotRequired

Base Path: `details.participants[*]`

Select Discourse API Operation: get `/t/{id}.json`

#### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `t_DetailsParticipantsById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>☑️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Function Columns

The columns of the table function `t_DetailsParticipantsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Auto Close Based On Last Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_flag_topic</td>
<td>boolean</td>
<td>Can Flag Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_by_avatar_template</td>
<td>string</td>
<td>Created by Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_by_id</td>
<td>decimal</td>
<td>Created by ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_by_username</td>
<td>string</td>
<td>Created by User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_avatar_template</td>
<td>string</td>
<td>Last Poster Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_id</td>
<td>decimal</td>
<td>Last Poster ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### t_DetailsSuggested_topicsById

Get a single topic

**Catalog:** Discourse

**Schema:** Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/t/{id}.json`

**Insert Discourse API URL:** `/t/{id}.json`

**Update Discourse API URL:** `/t/{id}.json`

**Delete Discourse API URL:** `/t/{id}.json`

**Field Selection Method:** NotRequired

**Base Path:** `details.suggested_topics[*]`

**Select Discourse API Operation:** `get /t/{id}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `t_DetailsSuggested_topicsById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td></td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>
### Table Function Columns

The columns of the table function `t_DetailsSuggested_topicsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archetype_1</td>
<td>string</td>
<td>Archetype 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived_1</td>
<td>boolean</td>
<td>Archived 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Auto Close Based On Last Post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_flag_topic</td>
<td>boolean</td>
<td>Can Flag Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id_1</td>
<td>decimal</td>
<td>Category ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed_1</td>
<td>boolean</td>
<td>Closed 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at_1</td>
<td>string</td>
<td>Created at 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_by_avatar_template</td>
<td>string</td>
<td>Created by Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_by_id</td>
<td>decimal</td>
<td>Created by ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_by_username</td>
<td>string</td>
<td>Created by User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title_1</td>
<td>string</td>
<td>Fancy Title 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number_1</td>
<td>decimal</td>
<td>Highest Post Number 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>image_url</td>
<td>string</td>
<td>Image URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_avatar_template</td>
<td>string</td>
<td>Last Poster Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_id</td>
<td>decimal</td>
<td>Last Poster ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count_1</td>
<td>decimal</td>
<td>Like Count 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>--------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_1</td>
<td>boolean</td>
<td>Pinned 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count_1</td>
<td>decimal</td>
<td>Posts Count 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count_1</td>
<td>decimal</td>
<td>Reply Count 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug_1</td>
<td>string</td>
<td>Slug 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title_1</td>
<td>string</td>
<td>Title 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views_1</td>
<td>decimal</td>
<td>Views 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible_1</td>
<td>boolean</td>
<td>Visible 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

46.1.14 t_DetailsSuggested_topicsPostersById

Get a single topic
Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json
Field Selection Method: NotRequired
Base Path: details.suggested_topics[*].posters[*]
Select Discourse API Operation: get /t/{id}.json

Table Function Parameters
The following parameters can be used to control the behaviour of the table function t_DetailsSuggested_topicsPostersById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function t_DetailsSuggested_topicsPostersById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>archetype_1</td>
<td>string</td>
<td>Archetype 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>archived_1</td>
<td>boolean</td>
<td>Archived 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Auto Close Based On Last Post</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_flag_topic</td>
<td>boolean</td>
<td>Can Flag Topic</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td>□️</td>
<td></td>
</tr>
<tr>
<td>category_id_1</td>
<td>decimal</td>
<td>Category ID 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>closed_1</td>
<td>boolean</td>
<td>Closed 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>created_at_1</td>
<td>string</td>
<td>Created at 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>extras</td>
<td>string</td>
<td>Extras</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>fancy_title_1</td>
<td>string</td>
<td>Fancy Title 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>highest_post_number_1</td>
<td>decimal</td>
<td>Highest Post Number 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>image_url</td>
<td>string</td>
<td>Image URL</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count_1</td>
<td>decimal</td>
<td>Like Count 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>pinned_1</td>
<td>boolean</td>
<td>Pinned 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>posts_count_1</td>
<td>decimal</td>
<td>Posts Count 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>reply_count_1</td>
<td>decimal</td>
<td>Reply Count 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>slug_1</td>
<td>string</td>
<td>Slug 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>title_1</td>
<td>string</td>
<td>Title 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>user_id_1</td>
<td>decimal</td>
<td>User ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_username</td>
<td>string</td>
<td>User Name</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>views_1</td>
<td>decimal</td>
<td>Views 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>visible_1</td>
<td>boolean</td>
<td>Visible 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

46.1.15 t_Post_streamPostsActions_summaryById
Get a single topic
Catalog: Discourse
Schema: Topics
This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}.json`

Insert Discourse API URL: `/t/{id}.json`

Update Discourse API URL: `/t/{id}.json`

Delete Discourse API URL: `/t/{id}.json`

Field Selection Method: NotRequired

Base Path: `post_stream.posts[*].actions_summary[*]`

Select Discourse API Operation: `get /t/{id}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `t_Post_streamPostsActions_summaryById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td></td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `t_Post_streamPostsActions_summaryById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_recover</td>
<td>boolean</td>
<td>Can Recover</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wiki</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at_1</td>
<td>string</td>
<td>Created at 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display Username</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_name</td>
<td>string</td>
<td>Poll Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_status</td>
<td>string</td>
<td>Poll Status</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_type</td>
<td>string</td>
<td>Poll Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_voters</td>
<td>int64</td>
<td>Poll Voters</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>boolean</td>
<td>Read</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count_1</td>
<td>decimal</td>
<td>Reply Count 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>user_id_1</td>
<td>decimal</td>
<td>User ID 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>decimal</td>
<td>Version</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>wiki</td>
<td>boolean</td>
<td>Wiki</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>yours</td>
<td>boolean</td>
<td>Yours</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

**46.1.16 t_Post_streamPostsById**

Get a single topic

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}.json`

Insert Discourse API URL: `/t/{id}.json`

Update Discourse API URL: `/t/{id}.json`

Delete Discourse API URL: `/t/{id}.json`

Field Selection Method: NotRequired

Base Path: `post_stream.posts[*]`

Select Discourse API Operation: `get /t/{id}.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `t_Post_streamPostsById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>[✓]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td>[☐]</td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `t_Post_streamPostsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>can_recover</td>
<td>boolean</td>
<td>Can Recover</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wikify</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>created_at_1</td>
<td>string</td>
<td>Created at 1</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display User Name</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>[☐]</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td>[☐]</td>
<td></td>
</tr>
</tbody>
</table>
### Discourse Data Model

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_name</td>
<td>string</td>
<td>Poll Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_status</td>
<td>string</td>
<td>Poll Status</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_type</td>
<td>string</td>
<td>Poll Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>polls_poll_voters</td>
<td>int64</td>
<td>Poll Voters</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>boolean</td>
<td>Read</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count_1</td>
<td>decimal</td>
<td>Reply Count 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id_1</td>
<td>decimal</td>
<td>User ID 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>decimal</td>
<td>Version</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>wiki</td>
<td>boolean</td>
<td>Wiki</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>yours</td>
<td>boolean</td>
<td>Yours</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**46.1.17 t_Post_streamPostsPolls_votesPollById**

Get a single topic

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table **NativePlatformScalarRequests** to upload data to the Discourse API.

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json
Field Selection Method: NotRequired
Base Path: post_stream.posts[*].polls_votes.poll[*]
Select Discourse API Operation: get /t/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `t_Post_streamPostsPolls_vo tesPollById`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td></td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `t_Post_streamPostsPolls_vo tesPollById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_recover</td>
<td>boolean</td>
<td>Can Recover</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wikify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>---------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>created_at_1</td>
<td>string</td>
<td>Created at 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display User Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>polls_name</td>
<td>string</td>
<td>Poll Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>polls_status</td>
<td>string</td>
<td>Poll Status</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>polls_type</td>
<td>string</td>
<td>Poll Type</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>polls_voters</td>
<td>int64</td>
<td>Poll Voters</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>boolean</td>
<td>Read</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>reply_count_1</td>
<td>decimal</td>
<td>Reply Count 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>
46.1.18 t_Post_streamPostsPollsPollOptionsById

Get a single topic
Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json
Field Selection Method: NotRequired
Base Path: post_stream.posts[*].polls.poll.options[*]
Select Discourse API Operation: get /t/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function t_Post_streamPostsPollsPollOptionsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
## Table Function Columns

The columns of the table function `t_Post_streamPostsPollsPollOptionsById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>can_delete</td>
<td>boolean</td>
<td>Can Delete</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>can_recover</td>
<td>boolean</td>
<td>Can Recover</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>can_view_edit_history</td>
<td>boolean</td>
<td>Can View Edit History</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>can_wiki</td>
<td>boolean</td>
<td>Can Wikify</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>cooked</td>
<td>string</td>
<td>Cooked</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>created_at_1</td>
<td>string</td>
<td>Created at 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>display_username</td>
<td>string</td>
<td>Display User Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>html</td>
<td>string</td>
<td>HTML</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td>ID</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>id_2</td>
<td>decimal</td>
<td>ID 2</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>incoming_link_count</td>
<td>decimal</td>
<td>#Incoming Links</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>name_1</td>
<td>string</td>
<td>Name 1</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td>----------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_number</td>
<td>decimal</td>
<td>Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_type</td>
<td>decimal</td>
<td>Post Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>quote_count</td>
<td>decimal</td>
<td>#Quotes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>boolean</td>
<td>Read</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reads</td>
<td>decimal</td>
<td>#Reads</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count_1</td>
<td>decimal</td>
<td>Reply Count 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>score</td>
<td>decimal</td>
<td>Score</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>boolean</td>
<td>Staff</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>string</td>
<td>Status</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>decimal</td>
<td>Topic ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>topic_slug</td>
<td>string</td>
<td>Topic Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>updated_at</td>
<td>string</td>
<td>Updated at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_deleted</td>
<td>boolean</td>
<td>User Deleted</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id_1</td>
<td>decimal</td>
<td>User ID 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>decimal</td>
<td>Version</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>voters</td>
<td>int64</td>
<td>Voters</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>votes</td>
<td>int64</td>
<td>Votes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>wiki</td>
<td>boolean</td>
<td>Wiki</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>yours</td>
<td>boolean</td>
<td>Yours</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**46.1.19 t_Post_streamStreamById**

Get a single topic

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json
Field Selection Method: NotRequired
Base Path: post_stream.stream[*]
Select Discourse API Operation: get /t/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function t_Post_streamStreamById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td></td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function t_Post_streamStreamById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
46.1.20 t_Timeline_lookup0ById

Get a single topic

Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json

Field Selection Method: NotRequired
Base Path: timeline_lookup[*].0[*]
Select Discourse API Operation: get /t/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function t_Timeline_lookup0ById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td></td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `t_Timeline_lookup0ById` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 46.1.21 t_Timeline_lookupById

Get a single topic

Catalog: Discourse
Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json
Field Selection Method: NotRequired
Base Path: timeline_lookup[*]
Select Discourse API Operation: get /t/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function t_Timeline_lookupById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td></td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function t_Timeline_lookupById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------</td>
<td>----------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>details_auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Detail Auto Close Based On Last Post</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_can_flag_topic</td>
<td>boolean</td>
<td>Detail Can Flag Topic</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_created_by_avatar_template</td>
<td>string</td>
<td>Detail Created by Avatar Template</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_created_by_id</td>
<td>decimal</td>
<td>Detail Created by ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_created_by_username</td>
<td>string</td>
<td>Detail Created by User Name</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_last_poster_avatar_template</td>
<td>string</td>
<td>Detail Last Poster Avatar Template</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_last_poster_id</td>
<td>decimal</td>
<td>Detail Last Poster ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_last_poster_username</td>
<td>string</td>
<td>Detail Last Poster User Name</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>details_notification_level</td>
<td>decimal</td>
<td>Detail Notification Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**46.1.22 tById**

Get a single topic
Catalog: Discourse
Schema: Topics
Retrieve: true
Insert: false
Update: false
Delete: true
Select Discourse API URL: /t/{id}.json
Insert Discourse API URL: /t/{id}.json
Update Discourse API URL: /t/{id}.json
Delete Discourse API URL: /t/{id}.json
Field Selection Method: NotRequired
Select Discourse API Operation: get /t/{id}.json
Delete Discourse API Operation: delete /t/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function tById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td>☑</td>
<td></td>
<td></td>
</tr>
<tr>
<td>print</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Will return up to 1000 posts in a topic</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function tById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>chunk_size</td>
<td>decimal</td>
<td>Chunk Size</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>details_auto_close_based_on_last_post</td>
<td>boolean</td>
<td>Detail Auto Close Based On Last Post</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>details_can_flag_topic</td>
<td>boolean</td>
<td>Detail Can Flag Topic</td>
<td>☑</td>
<td></td>
</tr>
</tbody>
</table>
### Discourse Data Model

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>details_created_by_avatar_template</td>
<td>string</td>
<td>Detail Created by Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>details_created_by_id</td>
<td>decimal</td>
<td>Detail Created by ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>details_created_by_username</td>
<td>string</td>
<td>Detail Created by Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>details_last_poster_avatar_template</td>
<td>string</td>
<td>Detail Last Poster Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>details_last_poster_id</td>
<td>decimal</td>
<td>Detail Last Poster ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>details_last_poster_username</td>
<td>string</td>
<td>Detail Last Poster User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>details_notification_level</td>
<td>decimal</td>
<td>Detail Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>participant_count</td>
<td>decimal</td>
<td>Participant Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_at</td>
<td>string</td>
<td>Pinned at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 46.1.23 tById_Post_streamPostsPosts

Get specific posts in a topic. Used for infinite scrolling or pagination.

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /t/{id}/posts.json

Insert Discourse API URL: /t/{id}/posts.json

Update Discourse API URL: /t/{id}/posts.json
Delete Discourse API URL: `/t/{id}/posts.json`
Field Selection Method: NotRequired
Base Path: `post_stream.posts[*]`
Select Discourse API Operation: `get /t/{id}/posts.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `tById_Post_streamPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>post_ids[]</td>
<td>string</td>
<td>□</td>
<td></td>
<td>post_ids can be found in the post_stream</td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `tById_Post_streamPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**46.1.24 tByIdPosts**

Get specific posts in a topic. Used for infinite scrolling or pagination.

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/t/{id}/posts.json`
Insert Discourse API URL: `/t/{id}/posts.json`
Update Discourse API URL: `/t/{id}/posts.json`

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Delete Discourse API URL: `/t/{id}/posts.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `get /t/{id}/posts.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `tByIdPosts`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>post_ids[]</td>
<td>string</td>
<td></td>
<td></td>
<td>post_ids can be found in the post_stream</td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `tByIdPosts` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**46.1.25 Top**

Get the top topics

Catalog: Discourse

Schema: Topics

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/top.json`

Insert Discourse API URL: `/top.json`

Update Discourse API URL: `/top.json`

Delete Discourse API URL: `/top.json`
Field Selection Method: NotRequired
Select Discourse API Operation: get /top.json

Table Columns

The columns of the table Top are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_sequence</td>
<td>decimal</td>
<td>Topic List Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_for_period</td>
<td>string</td>
<td>Topic List for Period</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>decimal</td>
<td>Topic List Per Page</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

46.1.26 Top_Topic_listTopics

Get the top topics
Catalog: Discourse
Schema: Topics

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /top.json
Insert Discourse API URL: /top.json
Update Discourse API URL: /top.json
Delete Discourse API URL: /top.json
Field Selection Method: NotRequired
Base Path: topic_list.topics[*]
Select Discourse API Operation: get /top.json

Table Columns

The columns of the table Top_Topic_listTopics are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>for_period</td>
<td>string</td>
<td>For Period</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unread</td>
<td>decimal</td>
<td>Unread</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unseen</td>
<td>boolean</td>
<td>Unseen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>views</td>
<td>decimal</td>
<td>Views</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**46.1.27 Top Topic list**

Get the top topics

Catalog: Discourse

Schema: Topics

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /top.json

Insert Discourse API URL: /top.json
Update Discourse API URL: /top.json
Delete Discourse API URL: /top.json
Field Selection Method: NotRequired
Base Path: topic_list.topics[*].posters[*]
Select Discourse API Operation: get /top.json

**Table Columns**

The columns of the table `Top_Topic_listTopicsPosters` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>archetype</td>
<td>string</td>
<td>Archetype</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bookmarked</td>
<td>boolean</td>
<td>Bookmarked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped</td>
<td>boolean</td>
<td>Bumped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bumped_at</td>
<td>string</td>
<td>Bumped at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_create_topic</td>
<td>boolean</td>
<td>Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>decimal</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_key</td>
<td>string</td>
<td>Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draft_sequence</td>
<td>decimal</td>
<td>Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extras</td>
<td>string</td>
<td>Extras</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fancy_title</td>
<td>string</td>
<td>Fancy Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>for_period</td>
<td>string</td>
<td>For Period</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_summary</td>
<td>boolean</td>
<td>Has Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest_post_number</td>
<td>decimal</td>
<td>Highest Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_poster_username</td>
<td>string</td>
<td>Last Poster Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_read_post_number</td>
<td>decimal</td>
<td>Last Read Post Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>like_count</td>
<td>decimal</td>
<td>#Likes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>liked</td>
<td>boolean</td>
<td>Liked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new_posts</td>
<td>decimal</td>
<td>New Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notification_level</td>
<td>decimal</td>
<td>Notification Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>per_page</td>
<td>decimal</td>
<td>Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td>boolean</td>
<td>Pinned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pinned_globally</td>
<td>boolean</td>
<td>Pinned Globally</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_count</td>
<td>decimal</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reply_count</td>
<td>decimal</td>
<td>#Replies</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
46.1.28 Top_Users: Discourse Top Users

Get the top topics
Catalog: Discourse
Schema: Topics
Label: Top Users

This is a read-only table. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /top.json
Insert Discourse API URL: /top.json
Update Discourse API URL: /top.json
Delete Discourse API URL: /top.json
Field Selection Method: NotRequired
Base Path: users[*]
Select Discourse API Operation: get /top.json

Table Columns

The columns of the table Top_Users are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_can_create_topic</td>
<td>boolean</td>
<td>Topic List Can Create Topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_key</td>
<td>string</td>
<td>Topic List Draft Key</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_draft_sequence</td>
<td>decimal</td>
<td>Topic List Draft Sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_for_period</td>
<td>string</td>
<td>Topic List for Period</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_list_per_page</td>
<td>decimal</td>
<td>Topic List Per Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
46.1.29 TopByFlag

Get the top topics filtered by specified flag

Catalog: Discourse

Schema: Topics

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/top/{flag}.json`

Insert Discourse API URL: `/top/{flag}.json`

Update Discourse API URL: `/top/{flag}.json`

Delete Discourse API URL: `/top/{flag}.json`

Field Selection Method: NotRequired

Select Discourse API Operation: get /top/{flag}.json

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `TopByFlag`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>flag</td>
<td>string</td>
<td></td>
<td></td>
<td>Values: all, yearly, quarterly, monthly, weekly, daily.</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `TopByFlag` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>
47 Schema: Upload
47.1 Tables
47.1.1 postUploads

Upload a file like an image or an avatar

Catalog: Discourse

Schema: Upload

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/uploads.json`

Insert Discourse API URL: `/uploads.json`

Update Discourse API URL: `/uploads.json`

Delete Discourse API URL: `/uploads.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `post /uploads.json`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postUploads`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `postUploads` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>filesize</td>
<td>int64</td>
<td>File Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>height</td>
<td>int64</td>
<td>Height</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table Columns

The columns of the table `AdminCustomizeByUserFields` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxx</td>
<td>decimal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
48.1.2 deleteAdminCustomizeUserFieldsById

delete a user field

Catalog: Discourse

Schema: UserFields

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/customize/user_fields/{id}

Insert Discourse API URL: /admin/customize/user_fields/{id}

Update Discourse API URL: /admin/customize/user_fields/{id}

Delete Discourse API URL: /admin/customize/user_fields/{id}

Field Selection Method: NotRequired

Base Path: success

Select Discourse API Operation: delete /admin/customize/user_fields/{id}

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function deleteAdminCustomizeUserFieldsById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function deleteAdminCustomizeUserFieldsById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
49 Schema: Users
49.1 Tables
49.1.1 AdminUsersListByFlag

Returns a list of users

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/users/list/{flag}.json`

Insert Discourse API URL: `/admin/users/list/{flag}.json`

Update Discourse API URL: `/admin/users/list/{flag}.json`

Delete Discourse API URL: `/admin/users/list/{flag}.json`

Field Selection Method: NotRequired

Select Discourse API Operation: `get /admin/users/list/{flag}.json`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `AdminUsersListByFlag`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascending</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flag</td>
<td>string</td>
<td>☑</td>
<td></td>
<td>Values: active, new, staff, suspended, blocked, suspect.</td>
</tr>
<tr>
<td>order</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Values: created, last_emailed, seen, username, email, trust_level, days_visited, posts_read, topics_viewed, posts, read_time.</td>
</tr>
<tr>
<td>show_emails</td>
<td>string</td>
<td>□</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**
The columns of the table function `AdminUsersListByFlag` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>blocked</td>
<td>boolean</td>
<td>Blocked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at_age</td>
<td>string</td>
<td>Created at Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>days_visited</td>
<td>int64</td>
<td>Days Visited</td>
<td></td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>string</td>
<td>Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flag_level</td>
<td>int64</td>
<td>Flag Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_emailed_age</td>
<td>string</td>
<td>Last Emailed Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_emailed_at</td>
<td>string</td>
<td>Last Emailed at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_seen_age</td>
<td>string</td>
<td>Last Seen Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_seen_at</td>
<td>string</td>
<td>Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>int64</td>
<td>#Posts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>posts_read_count</td>
<td>int64</td>
<td>Posts Read Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staged</td>
<td>boolean</td>
<td>Staged</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time_read</td>
<td>string</td>
<td>Time Read</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topics_entered</td>
<td>int64</td>
<td>Topics Entered</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>int64</td>
<td>Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level_locked</td>
<td>boolean</td>
<td>Trust Level Locked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username_lower</td>
<td>string</td>
<td>Username Lowercase</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**49.1.2 deleteAdminUsersById**

Delete a user.

**Catalog:** Discourse

**Schema:** Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/admin/users/{id}.json`

**Insert Discourse API URL:** `/admin/users/{id}.json`

**Update Discourse API URL:** `/admin/users/{id}.json`

**Delete Discourse API URL:** `/admin/users/{id}.json`
Field Selection Method: NotRequired
Base Path: deleted
Select Discourse API Operation: delete /admin/users/{id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function deleteAdminUsersById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function deleteAdminUsersById are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>

49.1.3 DirectoryItems

Get a public list of users
Catalog: Discourse
Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /directory_items.json?period={period}
Insert Discourse API URL: /directory_items.json?period={period}
Update Discourse API URL: /directory_items.json?period={period}
Discourse Data Model

Delete Discourse API URL: /directory_items.json?period={period}

Field Selection Method: NotRequired

Select Discourse API Operation: get /directory_items.json?period={period}

Table Function Parameters

The following parameters can be used to control the behaviour of the table function DirectoryItems. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>asc</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Values: True.</td>
</tr>
<tr>
<td>order</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Values: likes_received, likes_given, topic_count, post_count, topics_entered, posts_read, days_visited.</td>
</tr>
<tr>
<td>period</td>
<td>string</td>
<td>☑</td>
<td></td>
<td>Values: daily, weekly, monthly, quarterly, yearly, all.</td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function DirectoryItems are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>load_more_directory_items</td>
<td>string</td>
<td>Load More Directory Items</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>total_rows_directory_items</td>
<td>decimal</td>
<td>Total Rows Directory Items</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

49.1.4 DirectoryItems_Directory_items

Get a public list of users

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.
**Select Discourse API URL:** `/directory_items.json?period={period}`

**Insert Discourse API URL:** `/directory_items.json?period={period}`

**Update Discourse API URL:** `/directory_items.json?period={period}`

**Delete Discourse API URL:** `/directory_items.json?period={period}`

**Field Selection Method:** NotRequired

**Base Path:** `directory_items[*]`

**Select Discourse API Operation:** `get /directory_items.json?period={period}`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `DirectoryItems_Directory_items`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>asc</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Values: True.</td>
</tr>
<tr>
<td>order</td>
<td>string</td>
<td>□</td>
<td></td>
<td>Values: likes_received, likes_given, topic_count, post_count, topics_entered, posts_read, days_visited.</td>
</tr>
<tr>
<td>period</td>
<td>string</td>
<td>✓</td>
<td></td>
<td>Values: daily, weekly, monthly, quarterly, yearly, all.</td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `DirectoryItems_Directory_items` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>days_visited</td>
<td>decimal</td>
<td>Days Visited</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>likes_given</td>
<td>decimal</td>
<td>Likes Given</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>likes_received</td>
<td>decimal</td>
<td>Likes Received</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>load_more_directory_items</td>
<td>string</td>
<td>Load More Directory Items</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>post_count</td>
<td>decimal</td>
<td>#Posts</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Discourse Data Model

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>posts_read</td>
<td>decimal</td>
<td>Posts Read</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>topic_count</td>
<td>decimal</td>
<td>#Topics</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>topics Entered</td>
<td>decimal</td>
<td>Topics Entered</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>total_rows_directory_items</td>
<td>decimal</td>
<td>Total Rows Directory Items</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>user_name</td>
<td>string</td>
<td>User Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_username</td>
<td>string</td>
<td>User Name</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>

49.1.5 postAdminUsersByIdLogOut

Log a user out
Catalog: Discourse
Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/users/{id}/log_out
Insert Discourse API URL: /admin/users/{id}/log_out
Update Discourse API URL: /admin/users/{id}/log_out
Delete Discourse API URL: /admin/users/{id}/log_out
Field Selection Method: NotRequired
Base Path: success
Select Discourse API Operation: post /admin/users/{id}/log_out

Table Function Parameters

The following parameters can be used to control the behaviour of the table function postAdminUsersByIdLogOut. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
### Table Function Columns

The columns of the table function `postAdminUsersByIdLogOut` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 49.1.6 postUserAvatarByUsernameRefreshGravatar

Refresh a users gravatar

Catalog: Discourse  
Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/user_avatar/{username}/refresh_gravatar.json`  
Insert Discourse API URL: `/user_avatar/{username}/refresh_gravatar.json`  
Update Discourse API URL: `/user_avatar/{username}/refresh_gravatar.json`  
Delete Discourse API URL: `/user_avatar/{username}/refresh_gravatar.json`  
Field Selection Method: NotRequired

Select Discourse API Operation: post `/user_avatar/{username}/refresh_gravatar.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postUserAvatarByUsernameRefreshGravatar`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
Table Function Columns

The columns of the table function `postUserAvatarByUsernameRefreshGravatar` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>gravatar_avatar_template</td>
<td>string</td>
<td>Gravatar Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gravatar_upload_id</td>
<td>int64</td>
<td>Gravatar Upload ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

49.1.7 postUsers

Creates a new user
Catalog: Discourse
Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/users`
Insert Discourse API URL: `/users`
Update Discourse API URL: `/users`
Delete Discourse API URL: `/users`
Field Selection Method: NotRequired
Select Discourse API Operation: `post /users`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postUsers`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.
**Table Function Columns**

The columns of the table function `postUsers` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>message</td>
<td>string</td>
<td>Message</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>success</td>
<td>boolean</td>
<td>Success</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

### 49.1.8 `putUsersByUsernamePreferencesAvatarPick`

**Update avatar**

**Catalog:** Discourse  
**Schema:** Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/users/{username}/preferences/avatar/pick`  
**Insert Discourse API URL:** `/users/{username}/preferences/avatar/pick`  
**Update Discourse API URL:** `/users/{username}/preferences/avatar/pick`  
**Delete Discourse API URL:** `/users/{username}/preferences/avatar/pick`

**Field Selection Method:** NotRequired  
**Base Path:** `success`

**Select Discourse API Operation:** `put /users/{username}/preferences/avatar/pick`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `putUsersByUsernamePreferencesAvatarPick`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

### Table Function Columns

The columns of the table function `putUsersByUsernamePreferencesAvatarPick` are shown below. Each column has an SQL data type.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY</td>
<td>string(1)</td>
<td>Dummy</td>
<td></td>
<td>Default column added since the specification specifies that no data is returned.</td>
</tr>
</tbody>
</table>
```

#### 49.1.9 putUsersByUsernamePreferencesEmail

Update email

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/users/{username}/preferences/email`

Insert Discourse API URL: `/users/{username}/preferences/email`

Update Discourse API URL: `/users/{username}/preferences/email`

Delete Discourse API URL: `/users/{username}/preferences/email`

Field Selection Method: NotRequired

Select Discourse API Operation: `put /users/{username}/preferences/email`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `putUsersByUsernamePreferencesEmail`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `putUsersByUsernamePreferencesEmail` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT</td>
<td>string</td>
<td>Result</td>
<td></td>
<td>Outcome of operation as single plain text column.</td>
</tr>
</tbody>
</table>

#### 49.1.10 `uByExternal_User_badgesByExternalId`

Get a single user by SSO external ID

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/u/by-external/{external_id}.json`

Insert Discourse API URL: `/u/by-external/{external_id}.json`

Update Discourse API URL: `/u/by-external/{external_id}.json`

Delete Discourse API URL: `/u/by-external/{external_id}.json`

Field Selection Method: NotRequired

Base Path: `user_badges[*]`

Select Discourse API Operation: `get /u/by-external/{external_id}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `uByExternal_User_badgesByExternalId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `uByExternal_User_badgesByExternalId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_admin</td>
<td>boolean</td>
<td>User Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_badge_count</td>
<td>decimal</td>
<td>User Badge Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit</td>
<td>boolean</td>
<td>User Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_email</td>
<td>boolean</td>
<td>User Can Edit Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_name</td>
<td>boolean</td>
<td>User Can Edit Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_username</td>
<td>boolean</td>
<td>User Can Edit Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_message_to_user</td>
<td>boolean</td>
<td>User Can Send Private Message to User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_messages</td>
<td>boolean</td>
<td>User Can Send Private Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_created_at</td>
<td>string</td>
<td>User Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_posted_at</td>
<td>string</td>
<td>User Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_seen_at</td>
<td>string</td>
<td>User Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_moderator</td>
<td>boolean</td>
<td>User Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_pending_count</td>
<td>decimal</td>
<td>User Pending Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_profile_view_count</td>
<td>decimal</td>
<td>User Profile View Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_trust_level</td>
<td>decimal</td>
<td>User Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**49.1.11 uByExternal_UserFeatured_user_badge_idsByExternalId**

Get a single user by SSO external ID

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.
Select Discourse API URL: /u/by-external/{external_id}.json
Insert Discourse API URL: /u/by-external/{external_id}.json
Update Discourse API URL: /u/by-external/{external_id}.json
Delete Discourse API URL: /u/by-external/{external_id}.json
Field Selection Method: NotRequired
Base Path: user.featured_user_badge_ids[*]
Select Discourse API Operation: get /u/by-external/{external_id}.json

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function uByExternal_UserFeatured_user_badge_idsByExternalId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function uByExternal_UserFeatured_user_badge_idsByExternalId are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>badge_count</td>
<td>decimal</td>
<td>#Badges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit_email</td>
<td>boolean</td>
<td>Can Edit Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit_name</td>
<td>boolean</td>
<td>Can Edit Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit_username</td>
<td>boolean</td>
<td>Can Edit Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_send_private_message_to_user</td>
<td>boolean</td>
<td>Can Send Private Message to User</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 49.1.12 uByExternal_UserGroupsByExternalId

Get a single user by SSO external ID

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

**Select Discourse API URL:** `/u/by-external/{external_id}.json`

**Insert Discourse API URL:** `/u/by-external/{external_id}.json`

**Update Discourse API URL:** `/u/by-external/{external_id}.json`

**Delete Discourse API URL:** `/u/by-external/{external_id}.json`

Field Selection Method: NotRequired

Base Path: `user.groups[*]`

Select Discourse API Operation: `GET /u/by-external/{external_id}.json`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function uByExternal_UserGroupsByExternalId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the
default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_id</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `uByExternal_UserGroupsByExternalId` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>alias_level</td>
<td>decimal</td>
<td>Alias Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>automatic</td>
<td>boolean</td>
<td>Automatic</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>automatic_membership_retroactive</td>
<td>boolean</td>
<td>Automatic Membership Retroactive</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>badge_count</td>
<td>decimal</td>
<td>#Badges</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit_email</td>
<td>boolean</td>
<td>Can Edit Email</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit_name</td>
<td>boolean</td>
<td>Can Edit Name</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit_username</td>
<td>boolean</td>
<td>Can Edit Username</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_send_private_message_to_user</td>
<td>boolean</td>
<td>Can Send Private Message to User</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_send_private_messages</td>
<td>boolean</td>
<td>Can Send Private Messages</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>has_messages</td>
<td>boolean</td>
<td>Has Messages</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>is_member</td>
<td>boolean</td>
<td>Is Member</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_seen_at</td>
<td>string</td>
<td>Last Seen at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>mentionable</td>
<td>boolean</td>
<td>Mentionable</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>pending_count</td>
<td>decimal</td>
<td>Pending Count</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>primary_group</td>
<td>boolean</td>
<td>Primary Group</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>profile_view_count</td>
<td>decimal</td>
<td>Profile View Count</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>user_count</td>
<td>decimal</td>
<td>#Users</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
49.1.13 uByExternalByExternalId

Get a single user by SSO external ID

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /u/by-external/{external_id}.json

Insert Discourse API URL: /u/by-external/{external_id}.json

Update Discourse API URL: /u/by-external/{external_id}.json

Delete Discourse API URL: /u/by-external/{external_id}.json

Field Selection Method: NotRequired

Select Discourse API Operation: get /u/by-external/{external_id}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function uByExternalByExternalId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_id</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function uByExternalByExternalId are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_admin</td>
<td>boolean</td>
<td>User Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_badge_count</td>
<td>decimal</td>
<td>User Badge Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>user_can_edit</td>
<td>boolean</td>
<td>User Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_email</td>
<td>boolean</td>
<td>User Can Edit Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_name</td>
<td>boolean</td>
<td>User Can Edit Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_username</td>
<td>boolean</td>
<td>User Can Edit Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_message_to_user</td>
<td>boolean</td>
<td>User Can Send Private Message to User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_messages</td>
<td>boolean</td>
<td>User Can Send Private Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_created_at</td>
<td>string</td>
<td>User Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_posted_at</td>
<td>string</td>
<td>User Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_seen_at</td>
<td>string</td>
<td>User Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_moderator</td>
<td>boolean</td>
<td>User Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_pending_count</td>
<td>decimal</td>
<td>User Pending Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_profile_view_count</td>
<td>decimal</td>
<td>User Profile View Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_trust_level</td>
<td>decimal</td>
<td>User Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 49.1.14 UserActions

Get a list of user actions

**Catalog:** Discourse

**Schema:** Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** /user_actions.json

**Insert Discourse API URL:** /user_actions.json

**Update Discourse API URL:** /user_actions.json

**Delete Discourse API URL:** /user_actions.json

**Field Selection Method:** NotRequired

**Base Path:** user_actions[*]

**Select Discourse API Operation:** get /user_actions.json

## Table Function API Operation

The following parameters can be used to control the behaviour of the table function UserActions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-
defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>offset</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `UserActions` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>acting_avatar_template</td>
<td>string</td>
<td>Acting Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acting_name</td>
<td>string</td>
<td>Acting Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acting_user_id</td>
<td>int64</td>
<td>Acting User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>acting_username</td>
<td>string</td>
<td>Acting User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>action_type</td>
<td>int64</td>
<td>Action Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archived</td>
<td>boolean</td>
<td>Archived</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>category_id</td>
<td>int64</td>
<td>Category ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td>boolean</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td>boolean</td>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>excerpt</td>
<td>string</td>
<td>Excerpt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>boolean</td>
<td>Hidden</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post_id</td>
<td>int64</td>
<td>Post ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slug</td>
<td>string</td>
<td>Slug</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target_name</td>
<td>string</td>
<td>Target Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target_user_id</td>
<td>string</td>
<td>Target User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>target_username</td>
<td>string</td>
<td>Target User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topic_id</td>
<td>int64</td>
<td>Topic ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>int64</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
49.1.15 Users_User_badgesByUsername

Get a single user by username

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /users/{username}.json
Insert Discourse API URL: /users/{username}.json
Update Discourse API URL: /users/{username}.json
Delete Discourse API URL: /users/{username}.json

Field Selection Method: NotRequired

Base Path: user_badges[*]

Select Discourse API Operation: get /users/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function Users_User_badgesByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Table Function Columns

The columns of the table function Users_User_badgesByUsername are shown below. Each column has an SQL data type.
### 49.1.16 Users_UserFeatured_user_badge_idsByUsername

Get a single user by username

**Catalog:** Discourse

**Schema:** Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

**Select Discourse API URL:** `/users/{username}.json`

**Insert Discourse API URL:** `/users/{username}.json`

**Update Discourse API URL:** `/users/{username}.json`

**Delete Discourse API URL:** `/users/{username}.json`

**Field Selection Method:** NotRequired

**Base Path:** `user.featured_user_badge_ids[*]`

**Select Discourse API Operation:** get `/users/{username}.json`

### Table Function Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_admin</td>
<td>boolean</td>
<td>User Admin</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_badge_count</td>
<td>decimal</td>
<td>User Badge Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit</td>
<td>boolean</td>
<td>User Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_email</td>
<td>boolean</td>
<td>User Can Edit Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_name</td>
<td>boolean</td>
<td>User Can Edit Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_username</td>
<td>boolean</td>
<td>User Can Edit Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_message_to_user</td>
<td>boolean</td>
<td>User Can Send Private Message to User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_messages</td>
<td>boolean</td>
<td>User Can Send Private Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_created_at</td>
<td>string</td>
<td>User Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_posted_at</td>
<td>string</td>
<td>User Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_seen_at</td>
<td>string</td>
<td>User Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_moderator</td>
<td>boolean</td>
<td>User Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_pending_count</td>
<td>decimal</td>
<td>User Pending Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_profile_view_count</td>
<td>decimal</td>
<td>User Profile View Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_trust_level</td>
<td>decimal</td>
<td>User Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
The following parameters can be used to control the behaviour of the table function Users_UserFeatured_user_badge_idsByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function Users_UserFeatured_user_badge_idsByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>badge_count</td>
<td>decimal</td>
<td>#Badges</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit_email</td>
<td>boolean</td>
<td>Can Edit Email</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit_name</td>
<td>boolean</td>
<td>Can Edit Name</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_edit_username</td>
<td>boolean</td>
<td>Can Edit Username</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_send_private_message_to_user</td>
<td>boolean</td>
<td>Can Send Private Message to User</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>can_send_private_messages</td>
<td>boolean</td>
<td>Can Send Private Messages</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>last_seen_at</td>
<td>string</td>
<td>Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pending_count</td>
<td>decimal</td>
<td>Pending Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>profile_view_count</td>
<td>decimal</td>
<td>Profile View Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
49.1.17 Users_UserGroupsByUsername

Get a single user by username

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/users/{username}.json`

Insert Discourse API URL: `/users/{username}.json`

Update Discourse API URL: `/users/{username}.json`

Delete Discourse API URL: `/users/{username}.json`

Field Selection Method: NotRequired

Base Path: `user.groups[*]`

Select Discourse API Operation: get `/users/{username}.json`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `Users_UserGroupsByUsername`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function `Users_UserGroupsByUsername` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>boolean</td>
<td>Administrator</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>alias_level</td>
<td>decimal</td>
<td>Alias Level</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>automatic</td>
<td>boolean</td>
<td>Automatic</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Data Type</td>
<td>Label</td>
<td>Required</td>
<td>Documentation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>--------------------------------------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>automatic_membership_retroactive</td>
<td>boolean</td>
<td>Automatic Membership Retroactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>avatar_template</td>
<td>string</td>
<td>Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>badge_count</td>
<td>decimal</td>
<td>#Badges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit</td>
<td>boolean</td>
<td>Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit_email</td>
<td>boolean</td>
<td>Can Edit Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit_name</td>
<td>boolean</td>
<td>Can Edit Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_edit_username</td>
<td>boolean</td>
<td>Can Edit Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_send_private_message_to_user</td>
<td>boolean</td>
<td>Can Send Private Message to User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>can_send_private_messages</td>
<td>boolean</td>
<td>Can Send Private Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td>string</td>
<td>Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_messages</td>
<td>boolean</td>
<td>Has Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>decimal</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>decimal</td>
<td>ID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>is_member</td>
<td>boolean</td>
<td>Is Member</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_posted_at</td>
<td>string</td>
<td>Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_seen_at</td>
<td>string</td>
<td>Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mentionable</td>
<td>boolean</td>
<td>Mentionable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moderator</td>
<td>boolean</td>
<td>Moderator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pending_count</td>
<td>decimal</td>
<td>Pending Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>primary_group</td>
<td>boolean</td>
<td>Primary Group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>profile_view_count</td>
<td>decimal</td>
<td>Profile View Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trust_level</td>
<td>decimal</td>
<td>Trust Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_count</td>
<td>decimal</td>
<td>#Users</td>
<td></td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>string</td>
<td>User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>boolean</td>
<td>Visible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**49.1.18 UsersByUsername**

Get a single user by username

Catalog: Discourse

Schema: Users

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/users/{username}.json`

Insert Discourse API URL: `/users/{username}.json`

Update Discourse API URL: `/users/{username}.json`
Delete Discourse API URL: /users/{username}.json
Field Selection Method: NotRequired
Select Discourse API Operation: get /users/{username}.json

Table Function Parameters

The following parameters can be used to control the behaviour of the table function UsersByUsername. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Function Columns

The columns of the table function UsersByUsername are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_admin</td>
<td>boolean</td>
<td>User Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_avatar_template</td>
<td>string</td>
<td>User Avatar Template</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_badge_count</td>
<td>decimal</td>
<td>User Badge Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit</td>
<td>boolean</td>
<td>User Can Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_email</td>
<td>boolean</td>
<td>User Can Edit Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_edit_username</td>
<td>boolean</td>
<td>User Can Edit Username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_message_to_user</td>
<td>boolean</td>
<td>User Can Send Private Message to User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_can_send_private_messages</td>
<td>boolean</td>
<td>User Can Send Private Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_created_at</td>
<td>string</td>
<td>User Created at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_id</td>
<td>decimal</td>
<td>User ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_posted_at</td>
<td>string</td>
<td>User Last Posted at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_last_seen_at</td>
<td>string</td>
<td>User Last Seen at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_moderator</td>
<td>boolean</td>
<td>User Moderator</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
50 Schema: WebHooks

50.1 Tables

50.1.1 postAdminApi_Web_hookCategory_idsWebHooks

create a web hook

Catalog: Discourse

Schema: WebHooks

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the Discourse API.

Select Discourse API URL: /admin/api/web_hooks

Insert Discourse API URL: /admin/api/web_hooks

Update Discourse API URL: /admin/api/web_hooks

Delete Discourse API URL: /admin/api/web_hooks

Field Selection Method: NotRequired

Base Path: web_hook.category_ids[*]

Select Discourse API Operation: post /admin/api/web_hooks

Table Function API Operation Parameters

The following parameters can be used to control the behaviour of the table function postAdminApi_Web_hookCategory_idsWebHooks. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name | Data Type | Required | Default Value | Documentation
--- | --- | --- | --- | ---
body | string | | |
Table Function Columns

The columns of the table function `postAdminApi_Web_hookCategory_idsWebHooks` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>content_type</td>
<td>int64</td>
<td>Content Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_delivery_status</td>
<td>int64</td>
<td>Last Delivery Status</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>payload_url</td>
<td>string</td>
<td>Payload URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>secret</td>
<td>string</td>
<td>Secret</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>verify_certificate</td>
<td>string</td>
<td>Verify Certificate</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>wildcard_web_hook</td>
<td>boolean</td>
<td>Wildcard Web Hook</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

50.1.2 `postAdminApi_Web_hookGroup_idsWebHooks`

create a web hook

Catalog: Discourse

Schema: WebHooks

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/api/web_hooks`

Insert Discourse API URL: `/admin/api/web_hooks`

Update Discourse API URL: `/admin/api/web_hooks`

Delete Discourse API URL: `/admin/api/web_hooks`

Field Selection Method: NotRequired

Base Path: `web_hook.group_ids[*]`

Select Discourse API Operation: `post /admin/api/web_hooks`

Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postAdminApi_Web_hookGroup_idsWebHooks`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.
Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Function Columns

The columns of the table function `postAdminApi_Web_hookGroup_idsWebHooks` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>content_type</td>
<td>int64</td>
<td>Content Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_delivery_status</td>
<td>int64</td>
<td>Last Delivery Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>payload_url</td>
<td>string</td>
<td>Payload URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>secret</td>
<td>string</td>
<td>Secret</td>
<td></td>
<td></td>
</tr>
<tr>
<td>verify_certificate</td>
<td>string</td>
<td>Verify Certificate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wildcard_web_hook</td>
<td>boolean</td>
<td>Wildcard Web Hook</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 50.1.3 postAdminApi_Web_hookWeb_hook_event_typesWebHooks

create a web hook

Catalog: Discourse

Schema: WebHooks

This is a read-only table function. The Discourse API may not support changing the data or the Inventive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/api/web_hooks`

Insert Discourse API URL: `/admin/api/web_hooks`

Update Discourse API URL: `/admin/api/web_hooks`

Delete Discourse API URL: `/admin/api/web_hooks`

Field Selection Method: NotRequired

Base Path: `web_hook.web_hook_event_types[*]`

Select Discourse API Operation: `post /admin/api/web_hooks`

### Table Function Parameters

The following parameters can be used to control the behaviour of the table function `postAdminApi_Web_hookWeb_hook_event_typesWebHooks`. A value must be provided at all
times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td>□</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `postAdminApi_Web_hookWeb_hook_event_typesWebHooks` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>content_type</td>
<td>int64</td>
<td>Content Type</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>id_1</td>
<td>int64</td>
<td>ID 1</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>last_delivery_status</td>
<td>int64</td>
<td>Last Delivery Status</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Name</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>payload_url</td>
<td>string</td>
<td>Payload URL</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>secret</td>
<td>string</td>
<td>Secret</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>verify_certificate</td>
<td>string</td>
<td>Verify Certificate</td>
<td>□</td>
<td></td>
</tr>
<tr>
<td>wildcard_web_hook</td>
<td>boolean</td>
<td>Wildcard Web Hook</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**50.1.4 postAdminApiWebHooks**

create a web hook

Catalog: Discourse

Schema: WebHooks

This is a read-only table function. The Discourse API may not support changing the data or the Invantive SQL driver for Discourse does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Discourse API.

Select Discourse API URL: `/admin/api/web_hooks`

Insert Discourse API URL: `/admin/api/web_hooks`
Update Discourse API URL: `/admin/api/web_hooks`
Delete Discourse API URL: `/admin/api/web_hooks`
Field Selection Method: NotRequired
Base Path: `web_hook`
Select Discourse API Operation: `post /admin/api/web_hooks`

**Table Function Parameters**

The following parameters can be used to control the behaviour of the table function `postAdminApiWebHooks`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Default Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Function Columns**

The columns of the table function `postAdminApiWebHooks` are shown below. Each column has an SQL data type.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Label</th>
<th>Required</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>boolean</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>content_type</td>
<td>int64</td>
<td>Content Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>int64</td>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_delivery_status</td>
<td>int64</td>
<td>Last Delivery Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>payload_url</td>
<td>string</td>
<td>Payload URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>secret</td>
<td>string</td>
<td>Secret</td>
<td></td>
<td></td>
</tr>
<tr>
<td>verify_certificate</td>
<td>string</td>
<td>Verify Certificate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wildcard_web_hook</td>
<td>boolean</td>
<td>Wildcard Web Hook</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Index

- # -

#Badges 17, 20, 704, 706, 712, 714
#Incoming Links 106, 108, 109, 111, 665, 668, 670, 673
#Likes 17, 20, 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 680, 685, 686
#Quotes 106, 108, 109, 111, 665, 668, 670, 673
#Reads 106, 108, 109, 111, 665, 668, 670, 673
#Topics 17, 20, 71, 75, 76, 696
#Users 17, 29, 85, 706, 714

- A -

Acting Avatar Template 709
Acting Name 709
Acting User ID 709
Acting User Name 709
acting_avatar_template 709
acting_name 709
acting_user_id 709
acting_username 709
action 79
Action Type 709
action_id 92
action_name 92
action_type 709
Active 17, 20, 693, 700, 717, 718, 719, 720
admin 17, 20, 106, 108, 109, 111, 665, 668, 670, 673, 693, 704, 706, 712, 714
Admin Customized Templates by Email 41
Admin Settings by Email 40
Admin Site Setting Choices 16
Admin Site Settings 15
Admin Site Settings Valid Values 17
Admin_admin_badgesProtected_system_fieldsByBadges 45
Admin_Badge_grouping_idsByBadges 46
Admin_Badge_groupingsByBadges 46
Admin_Badge_idsByBadges 47
Admin_Badge_type_idsByBadges 48
Admin_Badge_typesByBadges 49
Admin_BadgesByBadges 50
Admin_SettingsByEmail 40
Admin_Site_settingsBySiteSettings 15
Admin_Site_settingsChoicesBySiteSettings 16
Admin_Site_settingsValid_valuesBySiteSettings 17
AdminCustomizeByEmailTemplates 41
AdminCustomizeByUserFields 691
AdminEmailsByAction 79
AdminFlags_PostsByType 80
AdminFlags_TopicsByType 81
AdminFlags_UsersByType 83
AdminFlagsByType 84
Administrators 13
AdminLogsByStaffActionLogs 92
AdminPlugins 101
AdminUsers_GroupsById 17
AdminUsersById 20
AdminUsersListByFlag 693
Alias Level 29, 706, 714
alias_level 29, 706, 714
Allow Badges 75, 76
Allow Membership Requests 17, 85
Allow Title 50
allow_badges 75, 76
allow_membership_requests 17, 85
allow_title 50
analysis-enforce-row-uniqueness 2
api-client-id 2
api-client-secret 2
api-redirect-url 2
api-refresh-token 2
api-scope 2
api-token-url 2
api-url 2
Archetype 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
Archetype_1 661, 663
archetype_1 661, 663
Archived 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 680, 685, 686, 709
Archived_1 661, 663
archived_1 661, 663
asc 695, 696
ascending 645, 646, 648, 649, 693
Auto Close Based On Last Post 75, 76, 659, 661, 663

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Index

723

Auto Revoke 50
body
24, 28, 29, 31, 34, 35, 38, 39, 41, 44, 53, 54,
55,
56,
57,
58, 59, 75, 76, 78, 88, 89, 90, 91, 98, 99,
auto_close_based_on_last_post
75, 76, 659, 661,
100,
102,
103,
104, 105, 113, 131, 143, 144, 145, 146,
663
147,
148,
149,
150, 151, 152, 153, 154, 155, 156, 157,
auto_revoke 50
158,
159,
160,
161, 162, 163, 164, 165, 166, 167, 168,
Automatic
17, 29, 85, 706, 714
169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179,
Automatic Membership Retroactive 17, 29, 85, 706,
180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190,
714
191, 192, 193, 195, 196, 197, 198, 199, 200, 201, 202,
automatic_membership_retroactive 17, 29, 85, 706,
203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213,
714
214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224,
109, 111, 126, 136, 649, 659, 665, 668, 670, 673, 688,
236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
693, 704, 706, 709, 712, 714
247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257,
avatar_template 13, 17, 20, 25, 70, 91, 106, 108, 258, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269,
109, 111, 126, 136, 649, 659, 665, 668, 670, 673, 688,
270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280,
693, 704, 706, 709, 712, 714
281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291,
292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302,
303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313,
314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324,
325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335,
Background URL 71, 75, 76
337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347,
background_url 71, 75, 76
348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358,
Backups
42
359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369,
370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380,
Badge Grouping ID 50
381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391,
Badge Settings
51
392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 403,
Badge Type ID 50
404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414,
badge_count
17, 20, 704, 706, 712, 714
415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425,
badge_grouping_id 50
426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436,
badge_type_id 50
437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447,
BadgeSettings
51
449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459,
Based on Last Post
652
460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470,
471, 472, 473, 474, 475, 477, 478, 479, 480, 481, 482,
based_on_last_post
652
483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493,
BLOB Preferred 94
494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504,
BLOB_PREFERRED 94
505, 506, 507, 508, 509, 510, 511, 512, 514, 515, 516,
Blocked 693
517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527,
Blurb 136
528, 529, 531, 532, 533, 534, 535, 536, 537, 538, 539,
540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550,
551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561,
562, 563, 564, 565, 566, 567, 568, 569, 571, 572, 573,
574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584,
585, 586, 587, 589, 590, 591, 592, 593, 594, 595, 596,
597, 598, 599, 600, 601, 603, 604, 605, 606, 607, 608,
609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619,
620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630,
631, 632, 633, 634, 635, 636, 637, 651, 652, 654, 655,
656, 690, 694, 700, 701, 702, 717, 718, 719, 720
BOL_RESPONSE_CACHE_MAX_AGE_SEC 94
Bookmarked 114, 116, 118, 121, 122, 124, 685,
686
Bounce Score 17, 20
bounce_score 17, 20
Bounced 79
bulk-delete-page-size-rows
2
bulk-insert-page-size-bytes
2
bulk-insert-page-size-rows
2

- B-

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.


Discourse Data Model

- C -

c_Topic_listTopicsByI d  67
c_Topic_listTopicsPostersByI d  68
c_UsersByI d  70
Can Act  109, 657
Can Activate  17, 20
Can Be Anonymized  17, 20
Can Be Deleted  17, 20
Can Create Category  71, 74
Can Create Topic  67, 68, 71, 74, 114, 116, 118, 119, 121, 122, 124, 646, 648, 685, 686
Can Deactivate  17, 20
Can Delete All Posts  17, 20
Can Disable Second Factor  17, 20
can_deactivate  17, 20
can_delete  75, 76, 106, 108, 109, 111, 665, 668, 670, 673
can_delete_all_posts  17, 20
can_disable_second_factor  17, 20
can_edit  71, 75, 76, 106, 108, 109, 111, 665, 668, 670, 673, 704, 706, 712, 714
can_edit_email  704, 706, 712, 714
can_edit_name  704, 706, 712, 714
can_edit_username  704, 706, 712, 714
can_flag_topic  659, 661, 663
can_grant_admin  17, 20
can_grant_moderation  17, 20
can_impersonate  17, 20
can_recover  108, 111, 665, 668, 670, 673
can_revert  41
can_revoke_admin  17, 20
can_revoke_moderation  17, 20
can_send_activation_email  17, 20
can_send_private_message_to_user  704, 706, 712, 714
Can Send Private Messages  704, 706, 712, 714
can_view_action_logs  17, 20
can_view_edit_history  106, 108, 109, 111, 665, 668, 670, 673
can_wiki  106, 108, 109, 111, 665, 668, 670, 673
Can View Action Logs  17, 20
Can View Edit History  106, 108, 109, 111, 665, 668, 670, 673
Can Wikify  106, 108, 109, 111, 665, 668, 670, 673
Can Act  109, 657
can_activate  17, 20
can_be_anonymized  17, 20
can_be_deleted  17, 20
can_create_category  71, 74
can_create_topic  67, 68, 71, 74, 114, 116, 118, 119, 121, 122, 124, 646, 648, 685, 686
(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
category_auto_close_based_on_last_post 78
category_background_url 78
category_can_delete 78
category_can_edit 78
category_color 78
category_email_in_allow_strangers 78
category_id 67, 68, 78, 106, 109, 128, 130, 137, 139, 646, 648, 652, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686, 709
category_id_1 661, 663
category_logo_url 78
category_name 78
category_position 78
category_post_count 78
category_read_restricted 78
category_slug 78
category_suppress_from_homepage 78
category_text_color 78
category_topic_count 78
category_topic_template 78
category_topic_url 78
CategoryById 73
CategorySettings 74
Chunk Size 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680
chunk_size 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680
Closed 1 661, 663
closed_1 661, 663
color 71, 75, 76
Colour 71, 75, 76
Content Type 94, 717, 718, 719, 720
description_type 94, 717, 718, 719, 720
Count 643, 657
Created at 1 661, 663, 665, 668, 670, 673
Created at Age 17, 20, 693
Created by Avatar Template 659, 661
Created by ID 659, 661
Created by User Name 659, 661
created_at 1 661, 663, 665, 668, 670, 673
created_at_age 17, 20, 693
created_by_avatar_template 659, 661
created_by_id 659, 661
created_by_username 659, 661

-D-

Data Display Username 97
Data Original Post ID 97
Data Original Post Type 97
Data Original User Name 97
Data Topic Title 97
data_display_username 97
data_original_post_id 97
data_original_post_type 97
data_original_username 97
data_topic_title 97
DATE_ENDED 94
DATE_STARTED 94
Days Visited 17, 20, 693, 696
days_visited 17, 20, 693, 696
dce 1
Default 15, 16, 17
Default Notification Level 17, 85
default_notification_level 17, 85
Delete 709
deleteAdminCustomizeUserFieldsById 692
deleteAdminUsersById 694
deleteAdminUsersByIdGroupsGroupId 23
deleteGroupsByIdMembers 23
deletePostActionsById 104
deleteUserBadgesById 52
Delivery Method 40, 42
delivery_method 40, 42
Description Excerpt 71
description_excerpt 71
description_text 71
detail Auto Close Based On Last Post 657, 678, 680
detail_can_flag_topic 657, 678, 680
detail Created by Avatar Template 659, 661
detail Created by ID 657, 678, 680
detail Created by User Name 657, 678, 680
detail Last Poster Avatar Template 657, 678, 680
detail Last Poster ID 657, 678, 680
detail Last Poster User Name 657, 678, 680
detail Notification Level 657, 678, 680
details_auto_close_based_on_last_post 657, 678, 680

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Discourse Data Model

details_can_flag_topic  657, 678, 680
details_created_by_avatar_template  657, 678, 680
details_created_by_id  657, 678, 680
details_created_by_username  657, 678, 680
details_last_poster_avatar_template  657, 678, 680
details_last_poster_id  657, 678, 680
details_last_poster_username  657, 678, 680
details_notification_level  657, 678, 680
DirectoryItems  695
DirectoryItems_Directory_items  696
download-error-io-exception-max-tries 2
download-error-io-exception-sleep-initial-ms 2
download-error-io-exception-sleep-max-ms 2
download-error-io-exception-sleep-multiplicator 2
download-error-json-exception-max-tries 2
download-error-json-exception-sleep-initial-ms 2
download-error-json-exception-sleep-max-ms 2
download-error-json-exception-sleep-multiplicator 2
download-error-other-exception-max-tries 2
download-error-other-exception-sleep-initial-ms 2
download-error-other-exception-sleep-max-ms 2
download-error-other-exception-sleep-multiplicator 2
download-error-socket-exception-max-tries 2
download-error-socket-exception-sleep-initial-ms 2
download-error-socket-exception-sleep-max-ms 2
download-error-socket-exception-sleep-multiplicator 2
download-error-web-exception-max-tries 2
download-error-web-exception-sleep-initial-ms 2
download-error-web-exception-sleep-max-ms 2
download-error-web-exception-sleep-multiplicator 2
download-error-web-not-implemented-max-tries 2
download-error-web-not-implemented-sleep-initial-ms 2
download-error-web-not-implemented-sleep-max-ms 2
download-error-web-not-implemented-sleep-multiplicator 2
download-error-web-timeout-max-tries 2
download-error-web-timeout-sleep-initial-ms 2
download-error-web-timeout-sleep-max-ms 2
download-error-web-timeout-sleep-multiplicator 2
download-error-web-unauthorized-max-tries 2
download-error-web-unauthorized-sleep-initial-ms 2
download-error-web-unauthorized-sleep-max-ms 2
download-error-web-unauthorized-sleep-multiplicator 2
Draft 67, 68, 71, 74
Draft Sequence 67, 68, 71, 74, 114, 116, 118, 119, 121, 122, 124, 685, 686
draft_sequence 67, 68, 71, 74, 114, 116, 118, 119, 121, 122, 124, 685, 686
Driver 1
DRY_RUN 94
DUMMY 23, 24, 32, 40, 45, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 68, 80, 81, 83, 84, 86, 88, 89, 90, 98, 131, 132, 133, 134, 135, 140, 141, 651, 654, 692, 694, 698, 701
Duration 652
Email 693
Email in Allow Strangers 75, 76
Email Settings 42
Email Type 79
email_in_allow_strangers 75, 76
email_type 79
Error Message Code 94
Error Message Text 94
ERROR_MESSAGE_CODE 94
ERROR_MESSAGE_TEXT 94
Excerpt 67, 68, 646, 648, 661, 663
Execute at 652
Extras 68, 118, 124, 648, 663, 686
Fancy Title 67, 68, 97, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 656, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
Fancy Title 1 661, 663
Fancy_title 67, 68, 97, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 656, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
Fancy_title_1 661, 663
File Size 690
filename 42, 43
Filter 709
flag 689, 693
Flag Level 17, 20, 693
flag_level 17, 20, 693
Flags Given Count 17, 20
Flags Received Count 17, 20
flags_given_count 17, 20
flags_received_count 17, 20
For Period 685, 686
for_period 685, 686
force-case-sensitive-identifiers 2
forced-casing-identifiers 2
Full Name 85
Full Suspend Reason 17, 20
full_name 85
full_suspend_reason 17, 20

- G -

Grant Count 50
grant_count 50
Gravatar Avatar Template 699
Gravatar Upload ID 699
gravatar_avatar_template 699
gravatar_upload_id 699
Group Allow Membership Requests 87
Group Automatic 87
Group Automatic Membership Retroactive 87
Group Default Notification Level 87
Group Full Name 87
Group Has Messages 87
Group ID 87, 128, 130
Group is Group Owner 87
Group is Group User 87
Group Mentionable 87
Group Mentionable Level 87
Group Messageable 87
Group Messageable Level 87
Group Name 76, 87
Group Primary Group 87
Group Public Admission 87
Group Public Exit 87
Group User Count 87
Group Visibility Level 87
group_allow_membership_requests 87
group_automatic 87
group_automatic_membership_retroactive 87
group_default_notification_level 87
group_full_name 87
group_has_messages 87
group_id 23, 87, 128, 130
group_is_group_owner 87
group_is_group_user 87
group_mentionable 87
group_mentionable_level 87
group_messageable 87
group_messageable_level 87
group_name 25, 26, 27, 76, 87

group_primary_group 87
group_public_admission 87
group_public_exit 87
group_user_count 87
group_visibility_level 87
Groups 85
GroupsExtrasVisible_group_namesByName 86
GroupsByGroupNameMembersMembers 25
GroupsByGroupName_OwnersMembers 26
GroupsByGroupNameMembers 27
GroupsByName 87

- H -

Has Children 71
Has Messages 17, 29, 85, 706, 714
Has Summary 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
has_children 71
has_messages 17, 29, 85, 706, 714
has_summary 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
Height 690
Hidden 106, 108, 109, 111, 657, 665, 668, 670, 673, 709
Hidden 1 109
hidden_1 109
Highest Post Number 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
Highest Post Number 1 661, 663
highest_post_number 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
highest_post_number_1 661, 663
HTML 673
HTTP Disk Cache Maximum Age (sec) 94
HTTP Memory Cache Maximum Age (sec) 94
HTTP Method 94
HTTP_DISK_CACHE_MAX_AGE_SEC 94
HTTP_DISK_CACHE_SAVE 94
HTTP_DISK_CACHE_USE 94
HTTP_MEMORY_CACHE_MAX_AGE_SEC 94
HTTP_MEMORY_CACHE_SAVE 94
HTTP_MEMORY_CACHE_USE 94
HTTP_METHOD 94
http-disk-cache-compression-level 2
http-disk-cache-directory 2
http-disk-cache-ignore-write-errors 2
http-disk-cache-max-age-sec 2
http-get-timeout-ms 2
http-memory-cache-compression-level 2
http-memory-cache-max-age-sec 2
http-post-timeout-ms 2
HTTPS 13, 14

-I-
Icon 50
ID 1 17, 109, 657, 659, 661, 663, 665, 668, 670, 673, 706, 714, 719
ID 2 673
id_1 17, 109, 657, 659, 661, 663, 665, 668, 670, 673, 706, 714, 719
id_2 673
ignore-http-400-errors 2
ignore-http-401-errors 2
ignore-http-402-errors 2
ignore-http-403-errors 2
ignore-http-404-errors 2
ignore-http-422-errors 2
ignore-http-429-errors 2
ignore-http-500-errors 2
ignore-http-502-errors 2
ignore-values-unknown-path 2
Image 50
Image URL 67, 68, 661, 663
image_url 67, 68, 661, 663
include_blurbs 132, 133, 134, 135, 136, 137, 139, 140, 141
incoming_link_count 106, 108, 109, 111, 665, 668, 670, 673
invalid-json-on-get-max-tries 2
invalid-json-on-get-sleep-initial-ms 2
invalid-json-on-get-sleep-max-ms 2
invalid-json-on-get-sleep-multiplicator 2
invalid-json-on-post-max-tries 2
invalid-json-on-post-sleep-initial-ms 2
invalid-json-on-post-sleep-max-ms 2
invalid-json-on-post-sleep-multiplicator 2
invantive-sql-correct-invalid-date 2
invantive-sql-forward-filters-to-data-containers 2
invantive-sql-shuffle-fetch-results-data-containers 2
invantive-use-cache 2
IP Address 17, 20
ip_address 17, 20
Is Group User 85
Is Member 29, 706, 714
is_group_user 85
is_member 29, 706, 714

-J-
join-set-points-per-request 2

-K-
Key 28

-L-
Last Delivery Status 717, 718, 719, 720
Last Emailed Age 17, 20, 693
Last Emailed at 17, 20, 693
Last Posted at 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 661, 663, 685, 686, 704, 706, 712, 714
Last Poster Avatar Template 659, 661
Last Poster ID 659, 661
Last Poster Username 67, 68, 114, 116, 118, 121, 122, 124, 646, 648, 659, 661, 685, 686
Last Read Post Number 114, 116, 118, 121, 122, 124, 685, 686
Last Seen Age 17, 20, 693
Last Seen at 17, 20, 693, 704, 706, 712, 714
last_delivery_status 717, 718, 719, 720
last_emailed_age 17, 20, 693
last_emailed_at 17, 20, 693
last_posted_at 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 661, 663, 685, 686, 704, 706, 712, 714
last_poster_avatar_template 659, 661
last_poster_id 659, 661
last_poster_username 67, 68, 114, 116, 118, 121, 122, 124, 646, 648, 659, 661, 685, 686
last_read_post_number 114, 116, 118, 121, 122, 124, 685, 686
last_seen_age 17, 20, 693
last_seen_at 17, 20, 693, 704, 706, 712, 714
Latest 645
Latest_Topic_listTopics 646
Latest_Topic_listTopicsPosters 648
Latest_Users 649
Like Count 1 661, 663
Like Given Count 17, 20
like_count 17, 20, 661, 663
like_count_1 661, 663
like_given_count 17, 20
Liked 114, 116, 118, 121, 122, 124, 685, 686
Likes Given 696
Likes Received 696
likes_given 696
likes_received 696
limit-partition-calls-left 2
Link 42
Listable 50
Load More Directory Items 695, 696
Load More Notifications 96, 97
load_more_directory_items 695, 696
load_more_notifications 96, 97
Locale 13, 14
log-native-calls-to-disk 2
log-native-calls-to-trace 2
Logo URL 71, 75, 76
logo_url 71, 75, 76
Long Description 50
long_description 50

- M -
maximum-discovered-column-count 2
maximum-length-identifiers 2
max-odata-filters 2
max-url-length-accepted 2
max-url-length-desired 2
Mentionable 29, 706, 714
Mentionable Level 17, 85
mentionable_level 17, 85
Message 700
Messageable Level 17, 85
messageable_level 17, 85
Meta Limit 25, 26, 27
Meta Offset 25, 26, 27
Meta Total 25, 26, 27
meta_limit 25, 26, 27
meta_offset 25, 26, 27
meta_total 25, 26, 27
metadata-cache-max-age-sec 2
Moderators 13
Multiple Grant 50
multiple_grant 50

- N -
Name 1 17, 673
name_1 17, 673
Native Platform Scalar Requests 94
NATIVEPLATFORMSCALARREQUESTS 94
New Posts 114, 116, 118, 121, 122, 124, 685, 686
new_posts 114, 116, 118, 121, 122, 124, 685, 686
Notification Level 71, 114, 116, 118, 121, 122, 124, 659, 661, 663, 685, 686
Notification Type 97
notification_level 71, 114, 116, 118, 121, 122, 124, 659, 661, 663, 685, 686
notification_type 97
Notifications 96
Notifications_Notifications 97
npt 94

- O -
offset 79, 80, 81, 83, 84, 709
One per Topic 638, 639, 640, 641, 642
one_per_topic 638, 639, 640, 641, 642
order 645, 646, 648, 649, 693, 695, 696
ORIG_SYSTEM_GROUP 94
ORIG_SYSTEM_REFERENCE 94
Original Filename 690
Original System Group 94
Original System Reference 94
original_filename 690

- P -
page-size-rows 2
PageViewTotalReqs 128
PageViewTotalReqs_ReportData 130
Parent Tag Name 638, 640
parent_tag_name 638, 640
Participant Count 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680
participant_count 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680
partition-slot-based-rate-limit-length-ms 2
partition-slot-based-rate-limit-slots 2
Payload 94
Payload URL 717, 718, 719, 720
PAYLOAD_TEXT 94
payload_url  717, 718, 719, 720
Pending Count    704, 706, 712, 714
pending_count   704, 706, 712, 714
Per Page      67, 68, 114, 116, 118, 119, 121, 122, 124, 646, 648, 685, 686
per_page       67, 68, 114, 116, 118, 119, 121, 122, 124, 646, 648, 685, 686
period      695, 696
Permission    71
Permission Type 76
permission_type 76
Pinned       67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686
Pinned 1 661, 663
Pinned at 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 680
pinned_1 661, 663
pinned_at 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 680
pinned_globally    67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 680
Plugins     101
PM Count    643
pm_count     643
Poll Name 102, 103, 665, 668, 670
Poll Option HTML    102, 103
Poll Option ID    102, 103
Poll Option Votes 102, 103
Poll Status 102, 103, 665, 668, 670
Poll Type    102, 103, 665, 668, 670
Poll Voters 102, 103, 665, 668, 670
poll_name 102, 103
poll_options_html 102, 103
poll_options_id 102, 103
poll_options_votes 102, 103
poll_status 102, 103
poll_type 102, 103
poll_voters 102, 103
polis_name 670
polls_poll_name 665, 668
polls_poll_status 665, 668
polls_poll_type 665, 668
polls_poll_voters 665, 668
polls_status 670
polls_type 670
polls_voters 670
Position    46, 71, 75, 76
Post ID    709
Post Number  97, 106, 108, 109, 111, 136, 665, 668, 670, 673
Post Type    106, 108, 109, 111, 665, 668, 670, 673
post_count    17, 20, 71, 75, 76, 659, 693, 696
post_id 709
post_ids[] 682, 683
post_number    97, 106, 108, 109, 111, 136, 665, 668, 670, 673
post_type    106, 108, 109, 111, 665, 668, 670, 673
postAdmin_Api_web_hookCategory_idsWebHooks
postAdmin_Api_web_hookGroup_idsWebHooks 718
postAdmin_Api_web_hookWeb_hook_event_typesWebHooks 719
postAdminApiKeys 28
postAdminApiWebHooks 720
postAdminGroups 29
postAdminUsersByldGroups 31
postAdminUsersByldLogOut 698
postExportCsvByExportEntity 131
postInvites 89
postInvitesLink 90
postPostActions 105
Posts 106
Posts Count 1 661, 663
Posts Latest Post Actions Summary 109
Posts Read 696
Posts Read Count 17, 20, 693
Posts_SummaryByld 108
posts_count 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 656, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 680, 685, 686
posts_count_1 661, 663
Posts_Latest_postsActions_summary 109
posts_read 696
posts_read_count 17, 20, 693
PostsByld 111
postSessionForgotPassword 99
postByldInvite 91
postByldNotifications 651
postByldTimer 652
postUploads 690
postUserAvatarByUsernameRefreshGravatar 699
postUserBadges 53
postUserBadges_Badge_types 54
postUserBadges_Badges 55
postUserBadges_Granted_bies 56
postUserBadges_Topics 57
postUserBadges_User_badges 58
postUserBadges_Users 59
postUsers 700
<table>
<thead>
<tr>
<th>Discourse Data Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>putAdminSiteSettingsCensoredPattern 405</td>
</tr>
<tr>
<td>putAdminSiteSettingsCensoredWords 406</td>
</tr>
<tr>
<td>putAdminSiteSettingsCleanOrphanUploadsGracePeriodHours 261</td>
</tr>
<tr>
<td>putAdminSiteSettingsCleanUpUploads 262</td>
</tr>
<tr>
<td>putAdminSiteSettingsCodeFormattingStyle 407</td>
</tr>
<tr>
<td>putAdminSiteSettingsColdAgeDaysHigh 345</td>
</tr>
<tr>
<td>putAdminSiteSettingsColdAgeDaysLow 346</td>
</tr>
<tr>
<td>putAdminSiteSettingsColdAgeDaysMedium 347</td>
</tr>
<tr>
<td>putAdminSiteSettingsCompanyDomain 479</td>
</tr>
<tr>
<td>putAdminSiteSettingsCompanyFullName 480</td>
</tr>
<tr>
<td>putAdminSiteSettingsCompanyShortName 481</td>
</tr>
<tr>
<td>putAdminSiteSettingsContactEmail 482</td>
</tr>
<tr>
<td>putAdminSiteSettingsContactUrl 483</td>
</tr>
<tr>
<td>putAdminSiteSettingsConvertPastedImagesQuality 263</td>
</tr>
<tr>
<td>putAdminSiteSettingsConvertPastedImagesToHqJpg 264</td>
</tr>
<tr>
<td>putAdminSiteSettingsCooldownMinutesAfterHidingPosts 512</td>
</tr>
<tr>
<td>putAdminSiteSettingsCorsOrigins 502</td>
</tr>
<tr>
<td>putAdminSiteSettingsCrawlImages 265</td>
</tr>
<tr>
<td>putAdminSiteSettingsCreateThumbnails 266</td>
</tr>
<tr>
<td>putAdminSiteSettingsDailyPerformanceReport 387</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultAvatars 267</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultCategoriesMuted 584</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultCategoriesTracking 585</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultCategoriesWatching 586</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultCategoriesWatchingFirstPost 587</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultCodeLang 408</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailAlways 589</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailDigestFrequency 590</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailDirect 591</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailInReplyTo 592</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailMailingListMode 593</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailMailingListModeFrequency 594</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailPreviousReplies 595</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultEmailPrivateMessages 596</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultIncludeTl0InDigests 597</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultInviteeTrustLevel 546</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultLocale 157</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOpenGraphImageUrl 484</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherAutoTrackTopicAfterMsecs 598</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherDisableJumpReply 599</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherDynamicFavicon 600</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherEnableQuoting 601</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherExternalLinksInNewTab 603</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherLikeNotificationFrequency 604</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherNewTopicDurationMinutes 605</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultOtherNotificationLevelWhenReplying 606</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultTopicsAutomaticUnpin 607</td>
</tr>
<tr>
<td>putAdminSiteSettingsDefaultTrustLevel 546</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteAllPostsMax 614</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteDigestEmailAfterDays 204</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteDigestEmails 211</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteDraftsOlderThanNDays 348</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteEmailLogsAfterDays 205</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteOldHiddenPosts 409</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteRemovedPostsAfter 410</td>
</tr>
<tr>
<td>putAdminSiteSettingsDeleteUserMaxPostAge 615</td>
</tr>
<tr>
<td>putAdminSiteSettingsDesktopCategoryPageStyle 158</td>
</tr>
<tr>
<td>putAdminSiteSettingsDetailsEnabled 388</td>
</tr>
<tr>
<td>putAdminSiteSettingsDigestLogoUrl 485</td>
</tr>
<tr>
<td>putAdminSiteSettingsDigestMinExcerptLength 206</td>
</tr>
<tr>
<td>putAdminSiteSettingsDigestOtherTopics 207</td>
</tr>
<tr>
<td>putAdminSiteSettingsDigestPosts 208</td>
</tr>
<tr>
<td>putAdminSiteSettingsDigestSuppressCategories 209</td>
</tr>
<tr>
<td>putAdminSiteSettingsDigestTopics 210</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableAvatarEducationMessages 349</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableDigestEmails 211</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableImageDownloadDomain 268</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableDiscourseNarrativeBotWelcomePost 332</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableEditNotifications 350</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableEmails 212</td>
</tr>
<tr>
<td>putAdminSiteSettingsDisableMailingListMode 608</td>
</tr>
<tr>
<td>putAdminSiteSettingsDiscourseNarrativeBotDisablePublicReplies 333</td>
</tr>
<tr>
<td>putAdminSiteSettingsDiscourseNarrativeBotEnabled 334</td>
</tr>
</tbody>
</table>
putAdminSiteSettingsDiscourseNarrativeBotIgnoredUsernames 335
putAdminSiteSettingsDiscourseNarrativeBotWelcomePostDelay 337
putAdminSiteSettingsDiscourseNarrativeBotWelcomePostType 338
putAdminSiteSettingsDisplayNameOnEmailFrom 213
putAdminSiteSettingsDisplayNameOnPosts 411
putAdminSiteSettingsDomingatingTopicMinimumPercentage 351
putAdminSiteSettingsDownloadRemoteImagesMaxDaysOld 269
putAdminSiteSettingsDownloadRemoteImagesThreshold 270
putAdminSiteSettingsDownloadRemoteImagesToLocal 271
putAdminSiteSettingsEditHistoryVisibleToPublic 412
putAdminSiteSettingsEditingGracePeriod 413
putAdminSiteSettingsEducateUntilPosts 352
putAdminSiteSettingsEmailAccentBgColor 214
putAdminSiteSettingsEmailAccentFgColor 215
putAdminSiteSettingsEmailCustomHeaders 216
putAdminSiteSettingsEmailDomainsBlacklist 292
putAdminSiteSettingsEmailDomainsWhitelist 293
putAdminSiteSettingsEmailEditable 616
putAdminSiteSettingsEmailIn 217
putAdminSiteSettingsEmailInMinTrust 218
putAdminSiteSettingsEmailLinkColor 219
putAdminSiteSettingsEmailPrefix 220
putAdminSiteSettingsEmailSiteTitle 221
putAdminSiteSettingsEmailSubject 222
putAdminSiteSettingsEmailTimeWindowMins 223
putAdminSiteSettingsEmailTokenValidHours 617
putAdminSiteSettingsEmbedTruncate 414
putAdminSiteSettingsEmojiSet 415
putAdminSiteSettingsEnableBadges 159
putAdminSiteSettingsEnableEmoji 416
putAdminSiteSettingsEnableEscapedFragments 503
putAdminSiteSettingsEnableFacebookLogins 395
putAdminSiteSettingsEnableFlashVideoOnebox 339
putAdminSiteSettingsEnableForwardedEmails 222
putAdminSiteSettingsEnableGithubLogins 295
putAdminSiteSettingsEnableGoogleOauth2Logins 296
putAdminSiteSettingsEnableInstagramLogins 297
putAdminSiteSettingsEnableLocalLogins 298
putAdminSiteSettingsEnableLongPolling 182
putAdminSiteSettingsEnableMobileTheme 160
putAdminSiteSettingsEnableNames 618
putAdminSiteSettingsEnableNoscriptSupport 504
putAdminSiteSettingsEnablePrivateMessages 417
putAdminSiteSettingsEnableS3Backups 148
putAdminSiteSettingsEnableS3Uploads 272
putAdminSiteSettingsEnableSignupCta 299
putAdminSiteSettingsEnableSso 300
putAdminSiteSettingsEnableSsoProvider 301
putAdminSiteSettingsEnableStagedUsers 225
putAdminSiteSettingsEnableTwitterLogins 302
putAdminSiteSettingsEnableUserDirectory 619
putAdminSiteSettingsEnableWhispers 161
putAdminSiteSettingsEnableYahooLogins 303
putAdminSiteSettingsExcludeRelNoFollowDomains 418
putAdminSiteSettingsExternalSystemAvatarsEnabled 273
putAdminSiteSettingsExternalSystemAvatarsUrl 274
putAdminSiteSettingsFacebookAppId 304
putAdminSiteSettingsFacebookAppSecret 305
putAdminSiteSettingsFacebookRequestExtraProfileDetails 306
putAdminSiteSettingsFaviconUrl 288
putAdminSiteSettingsFavIconUrl 487
putAdminSiteSettingsFixedCategoryPositions 162
putAdminSiteSettingsFixedCategoryPositionsOnCreate 163
putAdminSiteSettingsFlagSockpuppets 514
putAdminSiteSettingsForumPasswordStrict 307
putAdminSiteSettingsFullNameRequired 620
putAdminSiteSettingsGaDomainName 164
putAdminSiteSettingsGaTrackingCode 165
putAdminSiteSettingsGaUniversalDomainName 166
putAdminSiteSettingsGaUniversalTrackingCode 167
putAdminSiteSettingsGithubClientId 308
putAdminSiteSettingsGithubClientSecret 309
putAdminSiteSettingsGlobalNotice 353
putAdminSiteSettingsGoogleOauth2ClientId 310
putAdminSiteSettingsGoogleOauth2ClientSecret 311
putAdminSiteSettingsGtmContainerId 168
putAdminSiteSettingsHardBounceScore 226
putAdminSiteSettingsHideUserProfilesFromPublic 621
putAdminSiteSettingsHighlightedLanguages 419
putAdminSiteSettingsHistoryHoursHigh 354
putAdminSiteSettingsHistoryHoursLow 355
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>putAdminSiteSettingsHistoryHoursMedium</td>
<td>356</td>
</tr>
<tr>
<td>putAdminSiteSettingsIgnoreByTitle</td>
<td>227</td>
</tr>
<tr>
<td>putAdminSiteSettingsIncomingEmailPreferredHtml</td>
<td>313</td>
</tr>
<tr>
<td>putAdminSiteSettingsInstagramConsumerKey</td>
<td>312</td>
</tr>
<tr>
<td>putAdminSiteSettingsInstagramConsumerSecret</td>
<td>227</td>
</tr>
<tr>
<td>putAdminSiteSettingsInviteExpiryDays</td>
<td>622</td>
</tr>
<tr>
<td>putAdminSiteSettingsInviteOnly</td>
<td>314</td>
</tr>
<tr>
<td>putAdminSiteSettingsInvitePassThroughHours</td>
<td>623</td>
</tr>
<tr>
<td>putAdminSiteSettingsInvitesPerPage</td>
<td>624</td>
</tr>
<tr>
<td>putAdminSiteSettingsLevenshteinDistanceSpammerMails</td>
<td>516</td>
</tr>
<tr>
<td>putAdminSiteSettingsLimitSuggestedToCategory</td>
<td>169</td>
</tr>
<tr>
<td>putAdminSiteSettingsLoginRequired</td>
<td>315</td>
</tr>
<tr>
<td>putAdminSiteSettingsLogMailProcessingFailures</td>
<td>229</td>
</tr>
<tr>
<td>putAdminSiteSettingsLogoSmallUrl</td>
<td>488</td>
</tr>
<tr>
<td>putAdminSiteSettingsLogoUrl</td>
<td>489</td>
</tr>
<tr>
<td>putAdminSiteSettingsLogoutRedirect</td>
<td>625</td>
</tr>
<tr>
<td>putAdminSiteSettingsLongPollingBaseUrl</td>
<td>185</td>
</tr>
<tr>
<td>putAdminSiteSettingsLongPollingInterval</td>
<td>186</td>
</tr>
<tr>
<td>putAdminSiteSettingsMailgunApiKey</td>
<td>230</td>
</tr>
<tr>
<td>putAdminSiteSettingsManualPollingEnabled</td>
<td>231</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxAgeUnmatchedEmails</td>
<td>517</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxApiKeysPerUser</td>
<td>609</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxAttachmentSizeKb</td>
<td>275</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxBookmarksPerPage</td>
<td>458</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxEditsPerDay</td>
<td>459</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxEmailsPerPagePerUser</td>
<td>232</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxFlagsPerDay</td>
<td>460</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxImageHeight</td>
<td>276</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxImageMegapixels</td>
<td>277</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxImageSizeKb</td>
<td>278</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxImageWidth</td>
<td>279</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxMaximumBackups</td>
<td>149</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxMaximumSessionAge</td>
<td>317</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxMaximumStagedUsersPerEmail</td>
<td>233</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxInvitesPerPage</td>
<td>461</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxLikesPerPage</td>
<td>462</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxMentionsPerPage</td>
<td>420</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxNewAccountsPerRegistrationNlp</td>
<td>519</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxOneboxesPerPage</td>
<td>340</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxPostLength</td>
<td>421</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxPrintsPerHourPerUser</td>
<td>463</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxPrivateMessagesPerDay</td>
<td>464</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxRepliesInFirstDay</td>
<td>465</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxReplyHistory</td>
<td>422</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxSimilarResults</td>
<td>357</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTagLength</td>
<td>532</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTagSearchResults</td>
<td>533</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTagsInFilterList</td>
<td>534</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTagsPerTopic</td>
<td>535</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTopicInvitationsPerDay</td>
<td>566</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTopicsInFirstDay</td>
<td>467</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTopicsPerDay</td>
<td>468</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxTopicTitleLength</td>
<td>423</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxUserApiRequestsPerDay</td>
<td>581</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxUserApiRequestsPerMinute</td>
<td>582</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxUsernameLength</td>
<td>626</td>
</tr>
<tr>
<td>putAdminSiteSettingsMaxUsersNotifiedPerGroupMention</td>
<td>424</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinAdminPasswordLength</td>
<td>627</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinBanEntriesForRollUp</td>
<td>520</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinBodyTitleLength</td>
<td>425</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinFirstPostLength</td>
<td>426</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinFirstPostTypingTime</td>
<td>521</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinimumTopicsSimilar</td>
<td>358</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinPasswordLength</td>
<td>628</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinPostLength</td>
<td>427</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinPrivateMessagePostLength</td>
<td>428</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinPrivateMessageTitleLength</td>
<td>429</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinSearchTermLength</td>
<td>496</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTitleSimilarLength</td>
<td>430</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTopicTitleLength</td>
<td>431</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustLevelForUserApiKey</td>
<td>583</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustLevelToTagTopics</td>
<td>536</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustToAllowSelfWiki</td>
<td>547</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustToCreateTag</td>
<td>537</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustToCreateTopic</td>
<td>548</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustToEditPost</td>
<td>549</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustToEditWikiPost</td>
<td>550</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinTrustToSendMessages</td>
<td>551</td>
</tr>
<tr>
<td>putAdminSiteSettingsMinUsernameLength</td>
<td>629</td>
</tr>
<tr>
<td>putAdminSiteSettingsMobileLogoUrl</td>
<td>490</td>
</tr>
<tr>
<td>putAdminSiteSettingsMustApproveUsers</td>
<td>318</td>
</tr>
<tr>
<td>putAdminSiteSettingsNativeAppInstallBanner</td>
<td>359</td>
</tr>
<tr>
<td>putAdminSiteSettingsNewuserMaxAttachments</td>
<td>432</td>
</tr>
</tbody>
</table>
putAdminSiteSettingsNewuserMaxImages     433
putAdminSiteSettingsNewuserMaxLinks     434
putAdminSiteSettingsNewuserMaxMentionsPerPost     435
putAdminSiteSettingsNewuserMaxRepliesPerTopic     436
putAdminSiteSettingsNewuserSpamHostThreshold     522
putAdminSiteSettingsNewVersionEmails     360
putAdminSiteSettingsNotificationEmail     491
putAdminSiteSettingsNotifyAboutFlagsAfter     361
putAdminSiteSettingsNotifyAboutQueuedPostsAfter     437
putAdminSiteSettingsNotifyModsWhenUserBlocked     523
putAdminSiteSettingsNumFlaggersToCloseTopic     524
putAdminSiteSettingsNumFlagsToCloseTopic     525
putAdminSiteSettingsNumSpamFlagsToBlockNewUser     526
putAdminSiteSettingsNumTl3FlagsToCloseTopic     527
putAdminSiteSettingsNumTl3UsersToBlockNewUser     528
putAdminSiteSettingsNumUsersToBlockNewUser     529
putAdminSiteSettingsOneboxDomainsBlacklist     341
putAdminSiteSettingsPendingUsersReminderDelay     319
putAdminSiteSettingsPermalinkNormalizations     362
putAdminSiteSettingsPollEditWindowMins     389
putAdminSiteSettingsPollEnabled     390
putAdminSiteSettingsPollingInterval     187
putAdminSiteSettingsPollMaximumOptions     391
putAdminSiteSettingsPop3PollingEnabled     234
putAdminSiteSettingsPop3PollingHost     235
putAdminSiteSettingsPop3PollingOpensslVerify     236
putAdminSiteSettingsPop3PollingPassword     237
putAdminSiteSettingsPop3PollingPeriodMins     238
putAdminSiteSettingsPop3PollingPort     239
putAdminSiteSettingsPop3PollingSsl     240
putAdminSiteSettingsPop3PollingUsername     241
putAdminSiteSettingsPop3PollingPort     188
putAdminSiteSettingsPostEditTimeLimit     438
putAdminSiteSettingsPostExcerptMaxLength     439
putAdminSiteSettingsPostMenu     170
putAdminSiteSettingsPostMenuHiddenItems     171
putAdminSiteSettingsPostOneboxMaxLength     342
putAdminSiteSettingsPostUndoActionWindowMins     440
putAdminSiteSettingsPreventAnonsFromDownloadingFiles     280
putAdminSiteSettingsPreviousVisitTimeoutHours     363
putAdminSiteSettingsPrioritizeUsernameInUx     630
putAdminSiteSettingsPrivacyPolicyUrl     289
putAdminSiteSettingsPrivateEmailTimeWindowSeconds     242
putAdminSiteSettingsPublicUserCustomFields     631
putAdminSiteSettingsPurgeDeletedUploadsGracePeriodDays     281
putAdminSiteSettingsPurgeUnactivatedUsersGracePeriodDays     632
putAdminSiteSettingsQueueJobs     189
putAdminSiteSettingsRateLimitCreatePost     469
putAdminSiteSettingsRateLimitCreateTopic     470
putAdminSiteSettingsRateLimitNewUserCreatePost     471
putAdminSiteSettingsRateLimitNewUserCreateTopic     472
putAdminSiteSettingsReadOnlyModeDuringBackup     150
putAdminSiteSettingsReadTimeWordCount     364
putAdminSiteSettingsRebakeOldPostsCount     190
putAdminSiteSettingsRedirectUsersToTopPage     633
putAdminSiteSettingsRelativeDateDuration     172
putAdminSiteSettingsRemoveMutedTagsFromLatest     538
putAdminSiteSettingsReplyByEmailAddress     243
putAdminSiteSettingsReplyByEmailEnabled     244
putAdminSiteSettingsReservedUsernames     634
putAdminSiteSettingsResetBounceScoreAfterDays     245
putAdminSiteSettingsS3AccessKeyId     282
putAdminSiteSettingsS3BackupBucket     151
putAdminSiteSettingsS3CdnUrl     283
putAdminSiteSettingsS3DisableCleanup     152
putAdminSiteSettingsS3Region     284
putAdminSiteSettingsS3SecretAccessKey     285
putAdminSiteSettingsS3UploadBucket     286
putAdminSiteSettingsS3UseIamProfile     287
putAdminSiteSettingsSearchPreferRecentPosts     497
putAdminSiteSettingsSearchRecentPostsSize     498
putAdminSiteSettingsSearchTokenizeChineseJapaneseKorean     499
putAdminSiteSettingsSendWelcomeMessage     365
putAdminSiteSettingsSequentialRepliesThreshold     366
putAdminSiteSettingsSetLocaleFromAcceptLanguageHeader     173
putAdminSiteSettingsShortEmailLength     246
putAdminSiteSettingsShortProgressTextThreshold     41
putAdminSiteSettingsShowCreateTopicsNotice     367
Discourse Data Model

putAdminSiteSettingsShowEmailOnProfile 635
putAdminSiteSettingsShowFilterByTag 539
putAdminSiteSettingsShowPinnedExcerptDesktop 442
putAdminSiteSettingsShowPinnedExcerptMobile 443
putAdminSiteSettingsShowSubcategoryList 174
putAdminSiteSettingsShowTimeGapDays 444
putAdminSiteSettingsShowTopicFeaturedLinkInDigest 247
putAdminSiteSettingsSiteContactUsername 492
putAdminSiteSettingsSiteDescription 493
putAdminSiteSettingsSlugGenerationMethod 368
putAdminSiteSettingsSoftBounceScore 248
putAdminSiteSettingsSsoAllowsAllReturnPaths 320
putAdminSiteSettingsSsoNotApprovedUrl 321
putAdminSiteSettingsSsoOverridesAvatar 322
putAdminSiteSettingsSsoOverridesBio 323
putAdminSiteSettingsSsoOverridesEmail 324
putAdminSiteSettingsSsoOverridesName 325
putAdminSiteSettingsSsoOverridesUsername 326
putAdminSiteSettingsSsoSecret 327
putAdminSiteSettingsSsoUrl 328
putAdminSiteSettingsStaffLikeWeight 369
putAdminSiteSettingsStaffTags 540
putAdminSiteSettingsStaffUsersCustomFields 636
putAdminSiteSettingsStripImagesFromShortEmails 249
putAdminSiteSettingsSuggestedTopics 175
putAdminSiteSettingsSuggestedTopicsMaxDaysOld 176
putAdminSiteSettingsSummaryLikesRequired 370
putAdminSiteSettingsSummaryMaxResults 371
putAdminSiteSettingsSummaryPercentFilter 372
putAdminSiteSettingsSummaryPostsRequired 373
putAdminSiteSettingsSummaryScoreThreshold 374
putAdminSiteSettingsSuppressOverlappingTagsInList 541
putAdminSiteSettingsSuppressReplyDirectlyAbove 445
putAdminSiteSettingsSuppressReplyDirectlyBelow 446
putAdminSiteSettingsSuppressReplyWhenQuoting 447
putAdminSiteSettingsSuppressUncategorizedBadge 375
putAdminSiteSettingsTaggingEnabled 542
putAdminSiteSettingsTagsSortAlphabetically 543
putAdminSiteSettingsTagStyle 544
putAdminSiteSettingsTitle 494
putAdminSiteSettingsTitleFancyEntities 449
putAdminSiteSettingsTitleMaxWordLength 450
putAdminSiteSettingsTitleMinEntropy 451
putAdminSiteSettingsTitlePrettify 452
putAdminSiteSettingsTl1RequiresReadPosts 552
putAdminSiteSettingsTl1RequiresTimeSpentMins 553
putAdminSiteSettingsTl1RequiresTopicsEntered 554
putAdminSiteSettingsTl2AdditionalLikesPerDayMultiplier 473
putAdminSiteSettingsTl2RequiresDaysVisited 555
putAdminSiteSettingsTl2RequiresLikesGiven 556
putAdminSiteSettingsTl2RequiresLikesReceived 557
putAdminSiteSettingsTl2RequiresReadPosts 558
putAdminSiteSettingsTl2RequiresTimeSpentMins 559
putAdminSiteSettingsTl2RequiresTopicReplyCount 560
putAdminSiteSettingsTl2RequiresTopicsEntered 561
putAdminSiteSettingsTl3AdditionalLikesPerDayMultiplier 474
putAdminSiteSettingsTl3LinksNoFollow 562
putAdminSiteSettingsTl3PromotionMinDuration 563
putAdminSiteSettingsTl3RequiresDaysVisited 564
putAdminSiteSettingsTl3RequiresLikesGiven 565
putAdminSiteSettingsTl3RequiresLikesReceived 566
putAdminSiteSettingsTl3RequiresMaxFlagged 567
putAdminSiteSettingsTl3RequiresPostsRead 568
putAdminSiteSettingsTl3RequiresPostsReadAllTime 569
putAdminSiteSettingsTl3RequiresPostsReadCap 570
putAdminSiteSettingsTl3RequiresReadCap 571
putAdminSiteSettingsTl3RequiresTopicsReplyedTo 572
putAdminSiteSettingsTl3RequiresTopicsViewed 573
putAdminSiteSettingsTl3RequiresTopicsViewedAllTime 574
putAdminSiteSettingsTl3RequiresTopicsViewedCap 575
putAdminSiteSettingsTl3TimePeriod 576
putAdminSiteSettingsTl4AdditionalLikesPerDayMultiplier 475
putAdminSiteSettingsTopicFeaturedLinkEnabled 453
putAdminSiteSettingsTopicPageTitleIncludesCategory 376
putAdminSiteSettingsTopicPostLikeHeatHigh 377
putAdminSiteSettingsTopicPostLikeHeatLow 378
putAdminSiteSettingsTopicPostLikeHeatMedium 379
putAdminSiteSettingsTopicViewDurationHours 380
putAdminSiteSettingsTopicViewsHeatHigh 381

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
putAdminSiteSettingsTopicViewsHeatLow 382 putBySlugId 656
putAdminSiteSettingsTopicViewsHeatMedium 383 putUsersByUsernamePreferencesAvatarPick 701
putAdminSiteSettingsTopMenu 177 putUsersByUsernamePreferencesEmail 702
putAdminSiteSettingsTopTopicsFormulaFirstPostLike 191
putAdminSiteSettingsTopTopicsFormulaLeastLikesPerPostMultiplier 192
putAdminSiteSettingsTopTopicsFormulaLogViewsMultiplier 193
putAdminSiteSettingsTosUrl 290 quote_count 106, 108, 109, 111, 665, 668, 670, 673
putAdminSiteSettingsTrackExternalRightClicks 178
putAdminSiteSettingsTraditionalMarkdownLinebreaks 454
putAdminSiteSettingsTwitterConsumerKey 329
putAdminSiteSettingsTwitterConsumerSecret 330
putAdminSiteSettingsTwitterSummaryLargeImageUrl 495
putAdminSiteSettingsUniquePostsMin 477
putAdminSiteSettingsUnsubscribeViaEmail 250
putAdminSiteSettingsUnsubscribeViaEmailFooter 251
putAdminSiteSettingsUseAdminIpWhitelist 506
putAdminSiteSettingsUserProfileViewDurationHours 384
putAdminSiteSettingsVacuumDbDays 385
putAdminSiteSettingsVerboseDbLocalization 195
putAdminSiteSettingsVerboseSsoLogging 331
putAdminSiteSettingsVersionChecks 386
putAdminSiteSettingsWarnRevivingOldTopicAge 455
putAdminSiteSettingsWhiteListedSpamHostDomains 531
putAdminUsersByIdActivate 32
putAdminUsersByIdAnonymize 33
putAdminUsersByIdSilence 34
putAdminUsersByIdSuspend 35
putAdminUsersByIdUnsilence 36
putAdminUsersByIdUnsuspend 37
putCategories_CategoryAvailable_groupsById 75
putCategories_CategoryGroup_permissionsById 76
putCategoriesById 78
putGroupsById_UsernamesMembers 38
putGroupsByIdMembers 39
putGroupsById 88
putNotificationsByMarkRead 98
putPolls_VoteVote 102
putPollsVote 103
putPostsByIdLocked 113
putByIdBookmark 653
putByIdChangeTimestamp 654
putByIdStatus 655
Discourse Data Model


RESULT_BLOB 94
RESULT_TEX T 94
Run without Actions 94

Save HTTP Disk Cache 94
Save HTTP Memory Cache 94

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Start Date 94, 128, 130
start_date 128, 130
Static Doc 106, 109
static_doc 106, 109
Status 673
subject 41
Successful 94
Success 33, 38, 39, 75, 76, 78, 652, 655, 700
SUCCESSFUL 94
Suppress From Homepage 75, 76
suppress_from_homepage 75, 76
Suspended 17, 20
Suspended at 17, 20
Suspended by Avatar Template 17, 20
Suspended by ID 17, 20
Suspended by Username 17, 20
Suspended till 17, 20
suspended_at 17, 20
suspended_by_avatar_template 17, 20
suspended_by_id 17, 20
suspended_by_username 17, 20
suspended_till 17, 20
swagger-specification-http-disk-cache-max-age-sec 2
swagger-specification-url 2
System 46, 50
System Information 14
SystemInformation 14

t_Actions_summaryByI d 657
t_DetailsParticipantsByI d 659
t_DetailsSuggested_topicsByI d 661
t_DetailsSuggested_topicsPostersByI d 663
t_Post_streamPostsActions_summaryByI d 665
t_Post_streamPostsByI d 668
t_Post_streamPostsPolls_votesPollByI d 670
t_Post_streamPostsPollsPollOptionsByI d 673
t_Post_streamStreamByI d 675
t_Timeline_lookup0ByI d 677
t_Timeline_lookupByI d 678
tag 644
tag_name 639
Tag Groups 638
Tag Groups 638
TagGroups_Tag_groupParent_tag_nameByI d 639
TagGroups_Tag_groupsTag_names 640
TagGroups_Tag_groupTag_namesByI d 641
TagGroupsById 642
Tags 643
TagsByTag 644
Target Name 709
Target Posts 50
Target Tag 643
Target UserId 709
Target UserName 709
target_name 709
target_posts 50
target_tag 643
target_user_id 709
target_username 709
tById 680
tById_Post_streamPostsPosts 682
_topic_1 13, 661, 663
to_Address 79
to_address 79
Top 684
Top Users 688
Top_Topic_listTopics 685
top_Topic_listTopicsPosters 686
Top_Users 688
top_ByFlag 689
Topic HTML Title 106, 109
Topic ID 97, 106, 108, 109, 111, 116, 121, 122, 124, 128, 130, 137, 139, 140, 141, 665, 668, 670, 673, 709
Topic List Can Create Topic 70, 73, 126, 127, 645, 649, 684, 688
Topic List Draft 70, 73
Topic List Draft Key 70, 73, 126, 127, 645, 649, 684, 688
Topic List Draft Sequence 70, 73, 126, 127, 684, 688
Topic List for Period 684, 688
Topic List Per Page 70, 73, 126, 127, 645, 649, 684, 688
Topic Slug 106, 108, 109, 111, 665, 668, 670, 673
Topic Template 71, 75, 76
Topic Title 106, 109
Topic URL 71, 75, 76
topic_count 17, 20, 71, 75, 76, 696
topic_html_title 106, 109
_topic_list_can_create_topic 70, 73, 126, 127, 645, 649, 684, 688
topic_list_draft 70, 73
topic_list_draft_key 70, 73, 126, 127, 645, 649, 684, 688
topic_list_draft_sequence 70, 73, 126, 127, 684, 688
_topic_list_for_period 684, 688
topic_list_per_page 70, 73, 126, 127, 645, 649, 684, 688
topic_slug 106, 108, 109, 111, 665, 668, 670, 673
topic_template 71, 75, 76
topic_title 106, 109
topic_url 71, 75, 76
Topics All Time 71
Topics Day 71
Topics Entered 17, 20, 693, 696
Topics Month 71
Topics Week 71
Topics Year 71
topics_all_time 71
topic_day 71
topic_entered 17, 20, 693, 696
topic_month 71
topic_week 71
topic_year 71
Topics_PrivateMessages_Topic_listTopicsByUsername 114
Topics_PrivateMessages_Topic_listTopicsParticipantsByUsername 116
Topics_PrivateMessages_Topic_listTopicsPostersByUsername 118
Total 128, 130
Total Rows Directory Items 695, 696
Total Rows Notifications 96, 97
Total_rows_directory_items 695, 696

(C) Copyright 2004-2020 Invantive Software B.V., the Netherlands. All rights reserved.
Index

- U -

uByExternal_User_badgesByExternalI d 703
uByExternal_UserFeatured_user_badge_idsByExternalI d 704
uByExternal_UserGroupsByExternalI d 706
uByExternalByExternalI d 708
Unpinned 67, 68
Unread 114, 116, 118, 121, 122, 124, 685, 686
Unseen 67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 661, 663, 685, 686
Updated at 106, 108, 109, 111, 665, 668, 670, 672, 690
updated_at 106, 108, 109, 111, 665, 668, 670, 672, 690
URL 94, 101, 690
Use HTTP Disk Cache 94
Use HTTP Memory Cache 94
use-batch-insert 2
use-http-disk-cache-read 2
use-http-disk-cache-write 2
use-http-memory-cache-read 2
use-http-memory-cache-write 2
User Admin 703, 708, 711, 715
User Avatar Template 28, 663, 696, 703, 708, 711, 715
User Badge Count 703, 708, 711, 715
User Can Edit 703, 708, 711, 715
User Can Edit Email 703, 708, 711, 715
User Can Edit Name 703, 708, 711, 715
User Can Edit Username 703, 708, 711, 715
User Can Send Private Message to User 703, 708, 711, 715
User Can Send Private Messages 703, 708, 711, 715
User Created at 703, 708, 711, 715
User Deleted 106, 108, 109, 111, 665, 668, 670, 673
User Found 99
User History Action ID 92
User History Action Name 92
User ID 1 663, 665, 668, 670, 673
User Last Posted at 703, 708, 711, 715
User Last Seen at 703, 708, 711, 715
User Moderator 703, 708, 711, 715
User Pending Count 703, 708, 711, 715
User Profile View Count 703, 708, 711, 715
User Trust Level 703, 708, 711, 715
user_admin 703, 708, 711, 715
user_avatar_template 28, 663, 696, 703, 708, 711, 715
user_badge_count 703, 708, 711, 715
user_can_edit 703, 708, 711, 715
user_can_edit_email 703, 708, 711, 715
user_can_edit_name 703, 708, 711, 715
user_can_edit_username 703, 708, 711, 715
user_count 17, 29, 85, 706, 714
user_created_at 703, 708, 711, 715
user_deleted 106, 108, 109, 111, 665, 668, 670, 673
user_found 99
user_history_actions_id 92
user_history_actions_name 92
<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id_1</td>
<td>663, 665, 668, 670, 673</td>
</tr>
<tr>
<td>user_last_posted_at</td>
<td>703, 708, 711, 715</td>
</tr>
<tr>
<td>user_last_seen_at</td>
<td>703, 708, 711, 715</td>
</tr>
<tr>
<td>user_moderator</td>
<td>703, 708, 711, 715</td>
</tr>
<tr>
<td>user_name</td>
<td>696</td>
</tr>
<tr>
<td>user_pending_count</td>
<td>703, 708, 711, 715</td>
</tr>
<tr>
<td>user_profile_view_count</td>
<td>703, 708, 711, 715</td>
</tr>
<tr>
<td>user_trust_level</td>
<td>703, 708, 711, 715</td>
</tr>
<tr>
<td>user_username</td>
<td>28, 663, 696, 703, 708, 711, 715</td>
</tr>
<tr>
<td>UserActions</td>
<td>709</td>
</tr>
<tr>
<td>UserBadges_Badge_typesByUsername</td>
<td>60</td>
</tr>
<tr>
<td>UserBadges_BadgesByUsername</td>
<td>61</td>
</tr>
<tr>
<td>UserBadges_Granted_biesByUsername</td>
<td>62</td>
</tr>
<tr>
<td>UserBadges_TopicsByUsername</td>
<td>63</td>
</tr>
<tr>
<td>UserBadges_User_badgesByUsername</td>
<td>64</td>
</tr>
<tr>
<td>UserBadges_UsersByUsername</td>
<td>65</td>
</tr>
<tr>
<td>UserBadgesByUsername</td>
<td>66</td>
</tr>
<tr>
<td>Username Lowercase</td>
<td>17, 20, 693</td>
</tr>
<tr>
<td>username_lower</td>
<td>17, 20, 693</td>
</tr>
<tr>
<td>Users_User_badgesByUsername</td>
<td>711</td>
</tr>
<tr>
<td>Users_UserFeatured_user_badge_idsByUsername</td>
<td>712</td>
</tr>
<tr>
<td>Users_UserGroupsByUsername</td>
<td>714</td>
</tr>
<tr>
<td>UsersByUsername</td>
<td>715</td>
</tr>
<tr>
<td>Votes</td>
<td>673</td>
</tr>
<tr>
<td>Value</td>
<td>15, 16, 17, 40</td>
</tr>
<tr>
<td>Value 1</td>
<td>17</td>
</tr>
<tr>
<td>value_1</td>
<td>17</td>
</tr>
<tr>
<td>Verify Certificate</td>
<td>717, 718, 719, 720</td>
</tr>
<tr>
<td>verify_certificate</td>
<td>717, 718, 719, 720</td>
</tr>
<tr>
<td>Views</td>
<td>67, 68, 114, 116, 118, 121, 122, 124, 137, 139, 646, 648, 657, 659, 661, 663, 665, 668, 670, 673, 675, 677, 678, 680, 685, 686</td>
</tr>
<tr>
<td>Views 1</td>
<td>661, 663</td>
</tr>
<tr>
<td>views_1</td>
<td>661, 663</td>
</tr>
<tr>
<td>Visibility Level</td>
<td>17, 85</td>
</tr>
<tr>
<td>visibility_level</td>
<td>17, 85</td>
</tr>
<tr>
<td>Visible 1</td>
<td>661, 663</td>
</tr>
<tr>
<td>visible_1</td>
<td>661, 663</td>
</tr>
<tr>
<td>Voters</td>
<td>673</td>
</tr>
<tr>
<td>Warnings Received Count</td>
<td>17, 20</td>
</tr>
<tr>
<td>warnings_received_count</td>
<td>17, 20</td>
</tr>
<tr>
<td>Width</td>
<td>690</td>
</tr>
<tr>
<td>Wiki</td>
<td>106, 108, 109, 111, 665, 668, 670, 673</td>
</tr>
<tr>
<td>wildcard_web_hook</td>
<td>717, 718, 719, 720</td>
</tr>
<tr>
<td>X</td>
<td>130</td>
</tr>
<tr>
<td>xaxis</td>
<td>128, 130</td>
</tr>
<tr>
<td>X-Axis</td>
<td>128, 130</td>
</tr>
<tr>
<td>Y</td>
<td>130</td>
</tr>
<tr>
<td>yaxis</td>
<td>128, 130</td>
</tr>
<tr>
<td>Y-Axis</td>
<td>128, 130</td>
</tr>
<tr>
<td>Yours</td>
<td>106, 108, 109, 111, 665, 668, 670, 673</td>
</tr>
</tbody>
</table>
The JasperReports License, Version 1.0
Copyright (C) 2001-2004 Teodor Daniciu (teodord@users.sourceforge.net).
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by Teodor Daniciu (http://jasperreports.sourceforge.net)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The name "JasperReports" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact teodord@users.sourceforge.net.

5. Products derived from this software may not be called "JasperReports", nor may "JasperReports" appear in their name, without prior written permission of Teodor Daniciu.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.