

100%



PRODUCED  
RAN & OWNED

# Sendinblue API Data Model

*for use with Invantive SQL*

23.0



# Copyright

(C) Copyright 2004-2023 Invantive Software B.V., the Netherlands. All rights reserved.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Despite all the care taken in the compilation of this text, neither the author nor the publisher can accept liability for any damage, which might result from any error, which might appear in this publication.

This manual is a reference guide intended to clarify usage. If data in the sample images match data in your system, the similarity is coincidental.

## Important Safety and Usage Information

**Intended Use and Limitations:** This software, developed by Invantive, is designed to support a variety of business and information technology data processing functions, such as accounting, financial reporting and sales reporting. It is important to note that this software is not designed, tested, or approved for use in environments where malfunction or failure could lead to life-threatening situations or severe physical or environmental damage. This includes, but is not limited to:

- Nuclear facilities: The software should not be used for operations or functions related to the control, maintenance, or operation of nuclear facilities.
- Defense and Military Applications: This software is not suitable for use in defense-related applications, including but not limited to weaponry control, military strategy planning, or any other aspects of national defense.
- Aviation: The software is not intended for use in the operation, navigation, or communication systems of any aircraft or air traffic control environments.
- Healthcare and Medicine Production: This software should not be utilized for medical device operation, patient data analysis for critical health decisions, pharmaceutical production, or medical research where its failure or malfunction could impact patient health.
- Chemical and Hazardous Material Handling: This software is not intended for the management, control, or operational aspects of chemical plants or hazardous material handling facilities. Any malfunction in software used in these settings could result in dangerous chemical spills, explosions, or environmental disasters.
- Transportation and Traffic Control Systems: The software should not be used for the control, operation, or management of transportation systems, including railway signal controls, subway systems, or traffic light management. Malfunctions in such critical systems could lead to severe accidents and endanger public safety.
- Energy Grid and Utility Control Systems: This software is not designed for the control or operation of energy grid systems, including electrical substations, renewable energy control systems, or water utility control systems. The failure of software in these areas could lead to significant power outages, water supply disruptions, or other public utility failures, potentially endangering communities and causing extensive damage.
- Other High-Risk Environments: Any other critical infrastructure and environments where a failure of the software could result in significant harm to individuals or the environment.

**User Responsibility:** Users must ensure that they understand the intended use of the software and refrain from deploying it in any setting that falls outside of its designed purpose. It is the responsibility of the user to assess the suitability of the software for their intended application, especially in any scenarios that might pose a risk to life, health, or the environment.

**Disclaimer of Liability:** Invantive disclaims any responsibility for damage, injury, or legal consequences resulting from the use or misuse of this software in prohibited or unintended applications.

# Contents

<b>1</b>	<b>SQL Driver for Sendinblue API</b>	<b>1</b>
<b>2</b>	<b>SQL Driver Attributes for Sendinblue API</b>	<b>2</b>
<b>3</b>	<b>Catalog: Brevo</b>	<b>17</b>
<b>3.1</b>	<b>Schemas</b> .....	<b>17</b>
3.1.1	Schema: Account .....	17
3.1.2	Schema: Contacts .....	19
3.1.3	Schema: EmailCampaigns .....	55
3.1.4	Schema: Native .....	74
3.1.5	Schema: Process .....	75
3.1.6	Schema: Senders .....	77
3.1.7	Schema: SMSCampaigns .....	80
3.1.8	Schema: SMTP .....	86
3.1.9	Schema: TransactionalSMS .....	101
3.1.10	Schema: Views .....	105
3.1.11	Schema: Webhooks .....	108
3.1.12	Schema: WhatsAppCampaigns .....	111
<b>4</b>	<b>Catalog: dcr_metadata</b>	<b>0</b>
<b>4.1</b>	<b>Schemas</b> .....	<b>0</b>
4.1.1	dcr_metadata.get_partitions: Sendinblue Data container metadata package .....	0
<b>5</b>	<b>Catalog: sib_contacts</b>	<b>0</b>
<b>5.1</b>	<b>Schemas</b> .....	<b>0</b>
5.1.1	sib_contacts.addcontacttolist: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.2	sib_contacts.createattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.3	sib_contacts.createdocontact: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.4	sib_contacts.deleteattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.5	sib_contacts.importcontacts: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.6	sib_contacts.removecontactfromlist: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.7	sib_contacts.requestcontactexport: Sendinblue Package for Contacts operations that could not be mapped to a table.	
5.1.8	sib_contacts.updateattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.	
<b>6</b>	<b>Catalog: sib_email_campaigns</b>	<b>0</b>
<b>6.1</b>	<b>Schemas</b> .....	<b>0</b>
6.1.1	sib_email_campaigns.emailexportrecipients: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.	
6.1.2	sib_email_campaigns.sendemailcampaignnow: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.	
6.1.3	sib_email_campaigns.sendreport: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.	
6.1.4	sib_email_campaigns.sendtestemail: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.	
6.1.5	sib_email_campaigns.updatecampaignstatus: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.	
6.1.6	sib_email_campaigns.uploadimagetogallery: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.	
<b>7</b>	<b>Catalog: sib_sales</b>	<b>0</b>
<b>7.1</b>	<b>Schemas</b> .....	<b>0</b>
7.1.1	sib_sales.insert_note: Sendinblue Brevo sales .....	0
<b>8</b>	<b>Catalog: sib_sms_campaigns</b>	<b>0</b>
<b>8.1</b>	<b>Schemas</b> .....	<b>0</b>
8.1.1	sib_sms_campaigns.requestsmsrecipientexport: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.	

- 8.1.2 sib\_sms\_campaigns.sendsmscampaignnow: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.
- 8.1.3 sib\_sms\_campaigns.sendsmsreport: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.
- 8.1.4 sib\_sms\_campaigns.sendtestsms: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.
- 8.1.5 sib\_sms\_campaigns.updatecampaignstatus: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.

## 9 Catalog: sib\_smtp 0

### 9.1 Schemas ..... 0

- 9.1.1 sib\_smtp.deleteblockeddomain: Sendinblue Package for SMTP operations that could not be mapped to a table.
- 9.1.2 sib\_smtp.deletehardbounces: Sendinblue Package for SMTP operations that could not be mapped to a table.
- 9.1.3 sib\_smtp.sendtemplate: Sendinblue Package for SMTP operations that could not be mapped to a table. .... 0
- 9.1.4 sib\_smtp.sendtesttemplate: Sendinblue Package for SMTP operations that could not be mapped to a table. |
- 9.1.5 sib\_smtp.sendtransacemail: Sendinblue Package for SMTP operations that could not be mapped to a table.

## 10 Catalog: sib\_tracker 0

### 10.1 Schemas ..... 0

- 10.1.1 sib\_tracker.identify: Sendinblue Brevo tracker ..... 0
- 10.1.2 sib\_tracker.track\_event: Sendinblue Brevo tracker ..... 0
- 10.1.3 sib\_tracker.track\_link: Sendinblue Brevo tracker ..... 0
- 10.1.4 sib\_tracker.track\_page: Sendinblue Brevo tracker ..... 0

## 11 Catalog: sib\_transactional\_sms 0

### 11.1 Schemas ..... 0

- 11.1.1 sib\_transactional\_sms.sendtransacsms: Sendinblue Package for Transactional SMS operations that could not be mapped to a table.

## Index 117

# 1 SQL Driver for Sendinblue API

Invantive UniversalSQL is the fastest, easiest and most reliable way to exchange data with the Sendinblue 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](#). Invantive Support or other users will try to help you.

Brevo is cloud software to automate the marketing, sales and support process of prospects, leads and customers. Sendinblue includes email and SMS campaigns, sales management, website integration, chat, chat, tracking and advanced automatic workflows.

The Sendinblue driver covers 75 tables and 908 columns.

## Sendinblue API Clients

Invantive UniversalSQL is available on many user interfaces ("clients" in traditional server-client paradigm). All Invantive UniversalSQL 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 [Sendinblue Power BI connector](#) is based on the Invantive UniversalSQL driver for Sendinblue, 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 Sendinblue API into traditional databases such as SQL Server (on-premises and Azure), MySQL, PostgreSQL and Oracle is possible using [Invantive Data Replicator](#). Invantive Data Replicator automatically creates and maintains Sendinblue datawarehouses, possibly in combination with data from over 75 other (cloud) platforms. Invantive Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Sendinblue ADO.net provider.

Finally, online web apps can be build for Sendinblue using App Online of [Invantive Cloud](#).

## Monitor API Calls

When a query or DML-statement has been executed on Invantive UniversalSQL a developer can evaluate the actual calls made to the Sendinblue 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\NativeLog folder Invantive UniversalSQL will create log files per Sendinblue API request and response.

## Specifications

The SQL driver for Sendinblue does not support partitioning. Define one data container in a database for each company in Sendinblue to enable parallel access for data from multiple companies.

An introduction into the concepts of Invantive UniversalSQL such as databases, data containers and partitioning can be found in the [Invantive UniversalSQL grammar](#).

The configuration can be changed using various attributes from the database definition, on log on and during use. A full list of configuration options is listed in the [driver 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 Sendinblue the comparison of two texts is case sensitive by default.

Changes and bug fixes on the Sendinblue SQL driver can be found in the [release notes](#). Get access to the community through the [Sendinblue section](#) of the Invantive forums.

Driver code for use in settings.xml: `Sendinblue`

Alias: `sendinblue`

Recommended alias: `sib`

Driver code for use in settings.xml

The format of Brevo record IDs consists of a value from the view IDTypes, a dash as separator and a numeric value.

Updated 17-09-2024 15:16 using Invantive UniversalSQL version 24.1.8-BETA+4877.

## 2 SQL Driver Attributes for Sendinblue API

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

The Sendinblue 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.
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for Sendinblue can be found in the settings\*.xml file used for the database. The reference manuals contain instructions how to relocate the settings\*.xml files. Settings\*.xml files are typically located in the %USERPROFILE%\invantive folder in most deployment scenarios. 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 Sendinblue driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
add-odata-mandatory-filters	Whether to automatically add OData filters deemed necessary by the platform.	OData	False	✓	✓	✓	
analysis-enforce-row-uniqueness	Enforce rows to be unique for software analysis. A fingerprint is calculated from the whole row of data when the primary key column is unknown.	Shared	False	✓	✓	✓	
api-access-token	Access Token is a security token for multiple OAuth2 Flows. With an Access Token you can access protected resources. An Access Token must be stored securely since once compromised allows access to your protected resources.	OData		✓	✓	✓	✓
api-client-id	The client ID is a unique identifier of your application. It is generated by registering an application.	OData		✓		✓	✓
api-client-secret	The client secret is to be kept confidential. Such as a password for a login 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.	OData		✓		✓	✓
api-pre-expiry-refresh-sec	The number of seconds before the token expires to acquire a new token.	OData		✓	✓	✓	
api-redirect-url	The redirect URI is the website a browser session is redirected to after the OAuth2 authentication process has been completed.	OData		✓		✓	✓
api-refresh-token	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.	OData		✓		✓	✓

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
api-scope	The authorization scope(s) to request an OAuth token for.	OData		✓		✓	
api-token-url	The token URI is the OAuth2 endpoint to exchange tokens with.	OData		✓		✓	
api-url	URL of web service.	OData		✓		✓	
automation-key	The automation key for the tracking API. Available from <a href="https://automation.brevo.com/parameters">https://automation.brevo.com/parameters</a> .	Brevo		✓	✓	✓	✓
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	250	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when HTTP server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-408-request-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 408.		10	✓	✓	✓	
download-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 408.		10000	✓	✓	✓	
download-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 408.		300000	✓	✓	✓	
download-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 408.		2	✓	✓	✓	
download-error-422-bad-request-max-tries	Maximum number of tries when HTTP server reports unprocessable entity during retrieval of data.		30	✓	✓	✓	
download-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	



Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
dow nload-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
dow nload-error-429-too-many-requests-max-tries	Maximum number of tries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-initial-ms	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.		10000	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-max-ms	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.		300000	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-multiplicator	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.		2	✓	✓	✓	
dow nload-error-500-internal-server-error-max-tries	{res:itgen_pae_dow nload_error_500_internal_server_error_max_tries}		5	✓	✓	✓	
dow nload-error-500-internal-server-error-sleep-initial-ms	{res:itgen_pae_dow nload_error_500_internal_server_error_sleep_initial_ms}		1000	✓	✓	✓	
dow nload-error-500-internal-server-error-sleep-max-ms	{res:itgen_pae_dow nload_error_500_internal_server_error_sleep_max_ms}		5000	✓	✓	✓	
dow nload-error-500-internal-server-error-sleep-multiplicator	{res:itgen_pae_dow nload_error_500_internal_server_error_sleep_multiplicator}		2	✓	✓	✓	
dow nload-error-502-server-unavailable-max-tries	Maximum number of tries when HTTP server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
dow nload-error-502-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports a bad gateway during retrieval of data.		10000	✓	✓	✓	
dow nload-error-502-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that a bad gateway during retrieval of data.		300000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
dow nload-error-502-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports a bad gateway during retrieval of data.		2	✓	✓	✓	
dow nload-error-503-server-unavailable-max-tries	Maximum number of tries when HTTP server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
dow nload-error-503-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
dow nload-error-503-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		300000	✓	✓	✓	
dow nload-error-503-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
dow nload-error-504-gateway-timeout-max-tries	Maximum number of tries when the website reports a gateway timeout.		10	✓	✓	✓	
dow nload-error-504-gateway-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a gateway timeout.		10000	✓	✓	✓	
dow nload-error-504-gateway-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a gateway timeout.		300000	✓	✓	✓	
dow nload-error-504-gateway-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a gateway timeout.		2	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 590.		10	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 590.		10000	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 590.		300000	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 590.		2	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
download-error-599-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 599.		10	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 599.		10000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 599.		300000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 599.		2	✓	✓	✓	
download-error-argument-exception-max-tries	Maximum number of tries when an argument exception is returned when downloading a blob.		10	✓	✓	✓	
download-error-argument-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		10000	✓	✓	✓	
download-error-argument-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		300000	✓	✓	✓	
download-error-argument-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.		2	✓	✓	✓	
download-error-internet-download-max-tries	Maximum number of tries when the Internet connection seems down during retrieval of data.		10	✓	✓	✓	
download-error-internet-download-sleep-initial-ms	Initial sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		10000	✓	✓	✓	
download-error-internet-download-sleep-max-ms	Maximum sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		300000	✓	✓	✓	
download-error-internet-download-sleep-multiplicator	Multiplication factor for sleep between retries when the Internet connection seems down during retrieval of data.		2	✓	✓	✓	
download-error-io-exception-max-tries	Maximum number of tries when a network I/O connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-io-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		10000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
	tion failure occurs during retrieval of data.						
dow nload-error-io-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		300000	✓	✓	✓	
dow nload-error-io-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a network I/O connection failure occurs during retrieval of data.		2	✓	✓	✓	
dow nload-error-json-exception-max-tries	Maximum number of tries when an invalid JSON body is returned.		3	✓	✓	✓	
dow nload-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an invalid JSON body is returned.		1000	✓	✓	✓	
dow nload-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an invalid JSON body is returned.		10000	✓	✓	✓	
dow nload-error-json-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an invalid JSON body is returned.		2	✓	✓	✓	
dow nload-error-name-resolution-failure-max-tries	Maximum number of tries when the host name could not be resolved during retrieval of data.		5	✓	✓	✓	
dow nload-error-name-resolution-failure-sleep-initial-ms	Initial sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
dow nload-error-name-resolution-failure-sleep-max-ms	Maximum sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
dow nload-error-name-resolution-failure-sleep-multiplicator	{res:itgen_pae_dow nload_error_iname_resolution_failure_sleep_multiplicator}		1	✓	✓	✓	
dow nload-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
dow nload-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		10000	✓	✓	✓	
dow nload-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		300000	✓	✓	✓	
dow nload-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error occurs during retrieval of data.		2	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
download-error-socket-exception-max-tries	Maximum number of tries when the network connection is forcibly dropped during retrieval of data.		10	✓	✓	✓	
download-error-socket-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when the network connection is forcibly dropped during retrieval of data.		10000	✓	✓	✓	
download-error-socket-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when the network connection is forcibly dropped during retrieval of data.		300000	✓	✓	✓	
download-error-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcibly dropped during retrieval of data.		2	✓	✓	✓	
download-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-web-not-found-max-tries	{res:itgen_pae_download_error_web_not_found_max_tries}		1	✓	✓	✓	
download-error-web-not-found-sleep-initial-ms	{res:itgen_pae_download_error_web_not_found_sleep_initial_ms}		10000	✓	✓	✓	
download-error-web-not-found-sleep-max-ms	{res:itgen_pae_download_error_web_not_found_sleep_max_ms}		300000	✓	✓	✓	
download-error-web-not-found-sleep-multiplicator	{res:itgen_pae_download_error_web_not_found_sleep_multiplicator}		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-implemented-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports not implemented.		10000	✓	✓	✓	
download-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
dow nload-error-web-not-implemented-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports not implemented.		2	✓	✓	✓	
dow nload-error-web-timeout-max-tries	Maximum number of tries when the connection reports a timeout.		10	✓	✓	✓	
dow nload-error-web-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports a timeout.		1000	✓	✓	✓	
dow nload-error-web-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports a timeout.		30000	✓	✓	✓	
dow nload-error-web-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports a timeout.		2	✓	✓	✓	
dow nload-error-web-unauthorized-max-tries	Maximum number of tries when the connection reports an unauthorized error.		1	✓	✓	✓	
dow nload-error-web-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports an unauthorized error.		10000	✓	✓	✓	
dow nload-error-web-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports an unauthorized error.		300000	✓	✓	✓	
dow nload-error-web-unauthorized-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports an unauthorized error.		2	✓	✓	✓	
duplicate-row -mode	Configure behaviour when multiple rows are discovered during a limited search based upon a fingerprint generated from the primary key value, or all column values when no primary key is known. The following values can be used: "error" to raise error on duplicate fingerprint values in an API-page of rows, "keep" to ignore duplicate fingerprint values silently, "skip" like "keep" but with logging to trace of duplicate values or "null" for default behaviour (typically "error")..			✓	✓	✓	✓
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from: Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.	Shared	5	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
http-disk-cache-directory	Directory where HTTP cache is stored.	Shared	C:\Users\guido\In-vantive\Cache\http\guido\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.	Shared	False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.	Shared	2592000	✓	✓	✓	
http-get-timeout-max-ms	HTTP GET maximum timeout on retry (ms).		24000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		56000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.	OData	5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.	OData	14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		57000	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
	point.						
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-unknown-path-type	Whether to ignore path types not yet supported. An error will be generated when an unsupported type occurs.		True	✓	✓	✓	✓
ignore-values-unknown-path	Whether to ignore values outside of processed paths. An error will be generated when a value occurs outside a path otherwise.		True	✓	✓	✓	✓
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		1	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		1000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
invariantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invariantive-sql-correct-invalid-date	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.	SQL Engine V1	False	✓	✓	✓	
invariantive-sql-execution-profile-disk-path	{res:itgen_pae_invariantive_sql_execution_profile_disk_path}	SQL Engine V1	c:\temp\profiles	✓	✓	✓	
invariantive-sql-execution-profile-to-disk	{res:itgen_pae_invariantive_sql_execution_profile_to_disk}	SQL Engine V1	True	✓	✓	✓	
invariantive-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	



Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
max-odata-filters	Maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-odata-rewrite-in-count	{res:itgen_pae_max_odata_rewrite_in_count}	OData	500	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
maximum-discovered-column-count	Maximum number of discovered columns. An error will be generated when the column exceeds this value.		250	✓	✓	✓	✓
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
maximum-url-length	Maximum URL Length			✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.	OData	2	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	10000	✓	✓	✓	
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.	OData	2	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-504-errors-percentage	{res:itgen_pae_simulate_http_504_errors_percentage}		0	✓	✓	✓	
simulate-http-504-errors	{res:itgen_pae_simulate_http_504_errors}		False	✓	✓	✓	
simulate-http-522-errors-percentage	{res:itgen_pae_simulate_http_522_errors_percentage}		0	✓	✓	✓	
simulate-http-522-errors	{res:itgen_pae_simulate_http_522_errors}		False	✓	✓	✓	
simulate-http-524-errors-percentage	{res:itgen_pae_simulate_http_524_errors_percentage}		0	✓	✓	✓	
simulate-http-524-errors	{res:itgen_pae_simulate_http_524_errors}		False	✓	✓	✓	
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors when exchanging results with the HTTP endpoint..		False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	60000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
swagger-specification-download-tries	Specification Download Tries		5	✓	✓	✓	
swagger-specification-file	The Swagger file path, such as C:\temp\swagger.json.			✓	✓	✓	✓
swagger-specification-http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of Swagger specification data in the HTTP disk cache.		86400	✓	✓	✓	
swagger-specification-url	The Swagger URL such as https://example.org/rest/swagger.json.			✓	✓	✓	✓
totp-secret	Shared secret key to generate one-time password using TOTP RFC 6238. For improved security, manually enter the one-time password asked during login.	OData		✓		✓	✓
use-batch-insert	Whether to use batch insert.	OData	True	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.	Shared	False	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.	Shared	False	✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.	OData	True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.	OData	True	✓	✓	✓	

## 3 Catalog: Brevo

### 3.1 Schemas

#### 3.1.1 Schema: Account

##### 3.1.1.1 Tables

#### SubscriptionAccount: Sendinblue Subscription Account

Get your account information, plan and credits details

Catalog: Brevo

Schema: Account

Label: Subscription Account

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

### Table Columns

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

Name	Data Type	Label	Required	Documentation
address_city	string	Address City	<input type="checkbox"/>	City information
address_country	string	Address Country	<input type="checkbox"/>	Country information
address_street	string	Address Street	<input type="checkbox"/>	Street information
address_zipCode	string	Address Zip Code	<input type="checkbox"/>	Zip Code information
companyName	string	Company Name	<input type="checkbox"/>	Name of the company
email	string	Email Address	<input type="checkbox"/>	Login Email
firstName	string	First Name	<input type="checkbox"/>	First Name
lastName	string	Last Name	<input type="checkbox"/>	Last Name
marketingAutomation_enabled	boolean	Marketing Automation Enabled	<input type="checkbox"/>	Status of Marketing Automation Platform activation for your account (true=enabled, false=disabled)
marketingAutomation_key	string	Marketing Automation Key	<input type="checkbox"/>	Marketing Automation Tracker ID
relay_data_port	int64	Relay Data Port	<input type="checkbox"/>	Port used for SMTP Relay
relay_data_relay	string	Relay Data Relay	<input type="checkbox"/>	URL of the SMTP Relay
relay_data_userName	string	Relay Data Username	<input type="checkbox"/>	Email to use as login on transactional platform
relay_enabled	boolean	Relay Enabled	<input type="checkbox"/>	Status of your transactional email Account (true=Enabled, false=Disabled)

#### SubscriptionAccountPlans: Sendinblue Subscription Account Plans

Get your account information, plan and credits details

Catalog: Brevo

Schema: Account

Label: Subscription Account Plans

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

## Table Columns

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

Name	Data Type	Label	Required	Documentation
address_city	string	Address City	<input type="checkbox"/>	City information
address_country	string	Address Country	<input type="checkbox"/>	Country information
address_street	string	Address Street	<input type="checkbox"/>	Street information
address_zipCode	string	Address Zip Code	<input type="checkbox"/>	Zip Code information
companyName	string	Company Name	<input type="checkbox"/>	Name of the company
credits	float	Credits	<input type="checkbox"/>	Remaining credits of the user
creditsType	string	Credit Type	<input type="checkbox"/>	This is the type of the credit, "Send Limit" is one of the possible types of credit of a user. "Send Limit" implies the total number of emails you can send to the subscribers in your account.
email	string	Email Address	<input type="checkbox"/>	Login Email
endDate	datetime	End Date	<input type="checkbox"/>	Date of the period from which the plan will end (only available for "subscription" plan type)
firstName	string	First Name	<input type="checkbox"/>	First Name
lastName	string	Last Name	<input type="checkbox"/>	Last Name
marketingAutomation_enabled	boolean	Marketing Automation Enabled	<input type="checkbox"/>	Status of Marketing Automation Platform activation for your account (true=enabled, false=disabled)
marketingAutomation_key	string	Marketing Automation Key	<input type="checkbox"/>	Marketing Automation Tracker ID
relay_data_port	int64	Relay Data Port	<input type="checkbox"/>	Port used for SMTP Relay
relay_data_relay	string	Relay Data Relay	<input type="checkbox"/>	URL of the SMTP Relay
relay_data_userName	string	Relay Data Username	<input type="checkbox"/>	Email to use as login on transactional platform
relay_enabled	boolean	Relay Enabled	<input type="checkbox"/>	Status of your transactional email Account (true=Enabled, false=Disabled)
startDate	datetime	Start Date	<input type="checkbox"/>	Date of the period from which the plan will start (only available for "subscription" plan type)
type	string	Type	<input type="checkbox"/>	Displays the plan type of the user

### 3.1.2 Schema: Contacts

#### 3.1.2.1 Tables

##### Attributes: Sendinblue Attributes

List all attributes

Catalog: Brevo

Schema: Contacts

Label: Attributes

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

## Table Columns

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

Name	Data Type	Label	Required	Documentation
calculatedValue	string	Calculated Value	<input type="checkbox"/>	Calculated value formula
category	string	Category	<input type="checkbox"/>	Category of the attribute
name	string	Name	<input type="checkbox"/>	Name of the attribute
type	string	Type	<input type="checkbox"/>	Type of the attribute

##### AttributeValues: Sendinblue Attribute Values

List all attributes

Catalog: Brevo

Schema: Contacts

Label: Attribute Values

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

## Table Columns

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

Name	Data Type	Label	Required	Documentation
calculatedValue	string	Calculated Value	<input type="checkbox"/>	Calculated value formula
category	string	Category	<input type="checkbox"/>	Category of the attribute
label	string	Label	<input type="checkbox"/>	Label of the "category" type attribute
name	string	Name	<input type="checkbox"/>	Name of the attribute
type	string	Type	<input type="checkbox"/>	Type of the attribute
value	int64	Value	<input type="checkbox"/>	ID of Value of the "category" type attribute

### ContactAdminUnsubscriptionsByEmail: Sendinblue Contact Unsubscriptions by Email (Administrative Mode)

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Unsubscriptions by Email (Administrative Mode)

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
ip	string	IP	<input type="checkbox"/>	IP from which the user has been unsubscribed



## ContactClickStatisticsByEmail

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: {res:itgen\_sib\_contactclickstatisticsbyemail\_plural}

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event

### ContactComplaintStatisticsByEmail: Sendinblue Contact Complaint Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Complaint Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event

## ContactDetailAdminUnsubscriptionsByEmail: Sendinblue Contact Detail Unsubscriptions by Email (Administrative Mode)

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Unsubscriptions by Email (Administrative Mode)

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details

Name	Data Type	Label	Required	Documentation
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactDetailClickStatisticsByEmail

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: {res:itgen\_sib\_contactdetailclickstatisticsbyemail\_plural}

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign w hich generated the event
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
id	int64	ID	<input type="checkbox"/>	ID of the contact for w hich you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactDetailComplaintStatisticsByEmail: Sendinblue Contact Detail Complaint Statistics by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Complaint Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactDetailHardBounceStatisticsByEmail: Sendinblue Contact Detail Hard Bounce Statistics by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Hard Bounce Statistics by Email

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

## Parameters of Table Function

The following parameters can be used to control the behaviour of the table function `ContactDetailHardBounceStatisticsByEmail`. A value must be provided at all times for re-

quired parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign w hich generated the event
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
id	int64	ID	<input type="checkbox"/>	ID of the contact for w hich you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

## ContactDetailLinkClickStatisticsByEmail: Sendinblue Contact Detail Link Click Statistics by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Link Click Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign w hich generated the event
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
id	int64	ID	<input type="checkbox"/>	ID of the contact for w hich you requested the details



Name	Data Type	Label	Required	Documentation
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactDetailMessageSentStatisticsByEmail: Sendinblue Contact Detail Message Sent Statistics by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Message Sent Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactDetailOpenStatisticsByEmail: Sendinblue Contact Detail Open Statistics by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Open Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign w hich gener- ated the event
count	int64	Count	<input type="checkbox"/>	Number of openings for the cam- paign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM- DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email cam- paigns (true=blacklisted, false=not blacklisted)
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
id	int64	ID	<input type="checkbox"/>	ID of the contact for w hich you requested the details
ip	string	IP	<input type="checkbox"/>	IP from w hich the user has opened the email
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM- DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS cam- paigns (true=blacklisted, false=not blacklisted)

### ContactDetailSoftBounceStatisticsByEmail: Sendinblue Contact Detail Soft Bounce Statistics by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Soft Bounce Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign w hich generated the event
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
id	int64	ID	<input type="checkbox"/>	ID of the contact for w hich you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

## ContactDetailTransactionAttributesByEmail: Sendinblue Contact Detail Transaction Attributes by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Transaction Attributes by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactDetailUserUnsubscriptionsByEmail: Sendinblue Contact Detail User Unsubscriptions by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Detail User Unsubscriptions by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	<input type="checkbox"/>	Set of attributes of the contact
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-

Name	Data Type	Label	Required	Documentation
				DDTHH:mm:ss.SSSZ)
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

### ContactHardBounceStatisticsByEmail: Sendinblue Contact Hard Bounce Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Hard Bounce Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of

Name	Data Type	Required	Default Value	Documentation
				the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event

### ContactInfoByEmail: Sendinblue Contact Information by Email

Get a contact's details

Catalog: Brevo

Schema: Contacts

Label: Contact Information by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email (urlencoded) of the contact



Name	Data Type	Required	Default Value	Documentation
				OR its SMS attribute value

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF_TOT	string	{res:itgen_sib_abonnementen_actief_tot}	<input type="checkbox"/>	
ABONNEMENTEN_ACTIEF	string	{res:itgen_sib_abonnementen_actief}	<input type="checkbox"/>	
BLACKLIST	decimal	Blacklist	<input type="checkbox"/>	
BUBS_GBR_ID_MASTER	decimal	{res:itgen_sib_bubs_gbr_id_master}	<input type="checkbox"/>	
BUBS_GBR_ID	decimal	Estate Person ID	<input type="checkbox"/>	
CLICKERS	decimal	Clickers	<input type="checkbox"/>	
COMPANY_NAME	string	Company Name	<input type="checkbox"/>	
CONTACT_TIMEZONE	string	{res:itgen_sib_contact_timezone}	<input type="checkbox"/>	
CONTRACT_CODE	string	Contract Code	<input type="checkbox"/>	
COUNTRY	string	Country	<input type="checkbox"/>	
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	string	{res:itgen_sib_d31applications}	<input type="checkbox"/>	
D31BILLINGIDS	string	{res:itgen_sib_d31billingids}	<input type="checkbox"/>	
D31BUBSGBRIDS	string	{res:itgen_sib_d31bubsgbrids}	<input type="checkbox"/>	
D31DATACONTAINERCNT	decimal	{res:itgen_sib_d31datacontainercnt}	<input type="checkbox"/>	
D31MBIO	decimal	{res:itgen_sib_d31mbio}	<input type="checkbox"/>	
D31PARTITIONCNT	decimal	{res:itgen_sib_d31partitioncnt}	<input type="checkbox"/>	
D31PROVIDERCNT	decimal	{res:itgen_sib_d31providercnt}	<input type="checkbox"/>	
D31ROWCNT	decimal	{res:itgen_sib_d31rowcnt}	<input type="checkbox"/>	
D31TABLECNT	decimal	{res:itgen_sib_d31tablecnt}	<input type="checkbox"/>	
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted,

Name	Data Type	Label	Required	Documentation
				false=not blacklisted)
EXT_ID	string	{res:itgen_sib_ext_id}	<input type="checkbox"/>	
FIRSTNAME	string	First Name	<input type="checkbox"/>	
FORUMS_LOGIN	string	Forums Login	<input type="checkbox"/>	
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
JOBTITLE	string	Job Title	<input type="checkbox"/>	
LANDLINE_NUMBER	string	{res:itgen_sib_landline_number}	<input type="checkbox"/>	
LANGUAGE_CODE	string	Language Code	<input type="checkbox"/>	
LASTNAME	string	Last Name	<input type="checkbox"/>	
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
MOBILE_NUMBER	string	Mobile Number	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
PHONE_NUMBER	string	Phone Number	<input type="checkbox"/>	
PRIMARY_AUDIENCE	string	Primary Audience	<input type="checkbox"/>	
READERS	decimal	Readers	<input type="checkbox"/>	
SMS	string	SMS	<input type="checkbox"/>	
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	string	WhatsApp	<input type="checkbox"/>	

### ContactLinkClicksByEmail: Sendinblue Contact Link Clicks by Email

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Link Clicks by Email

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

## Parameters of Table Function

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

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select \* from table(value1, value2, value3)` on a table with four para-

meters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
count	int64	Count	<input type="checkbox"/>	Number of clicks on this link for the campaign
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
ip	string	IP	<input type="checkbox"/>	IP from which the user has clicked on the link
url	string	URL	<input type="checkbox"/>	URL of the clicked link

### **ContactMessageSentStatisticsByEmail: Sendinblue Contact Message Sent Statistics by Email**

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Message Sent Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event

### ContactOpenStatisticsByEmail: Sendinblue Contact Open Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Open Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
count	int64	Count	<input type="checkbox"/>	Number of openings of the campaign
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
ip	string	IP	<input type="checkbox"/>	IP from which the user has opened the campaign

### Contacts: Sendinblue Contacts

Get all the contacts

Catalog: Brevo

Schema: Contacts

Primary Keys: email

Label: Contacts

Can retrieve data and change data using insert, update and delete.

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
createdSince	datetime	<input type="checkbox"/>		Filter (urlencoded) the contacts created after a given UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ). Prefer to pass your timezone in date-time format for accurate result.
modifiedSince	datetime	<input type="checkbox"/>		Filter (urlencoded) the contacts modified after a given UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ). Prefer to pass your timezone in date-time format for accurate result.

## Columns of Table Function

The columns of the table function Contacts 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 and update.

Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF_TOT	string	{res:itgen_sib_abonnementen_actief_tot}	<input type="checkbox"/>	
ABONNEMENTEN_ACTIEF	string	{res:itgen_sib_abonnementen_actief}	<input type="checkbox"/>	
BLACKLIST	decimal	Blacklist	<input type="checkbox"/>	
BUBS_GBR_ID_MASTER	decimal	{res:itgen_sib_bubs_gbr_id_master}	<input type="checkbox"/>	
BUBS_GBR_ID	decimal	Estate Person ID	<input type="checkbox"/>	
CLICKERS	decimal	Clickers	<input type="checkbox"/>	
COMPANY_NAME	string	Company Name	<input type="checkbox"/>	
CONTACT_TIMEZONE	string	{res:itgen_sib_contact_timezone}	<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
CONTRACT_CODE	string	Contract Code	<input type="checkbox"/>	
COUNTRY	string	Country	<input type="checkbox"/>	
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	string	{res:itgen_sib_d31applications}	<input type="checkbox"/>	
D31BILLINGIDS	string	{res:itgen_sib_d31billingids}	<input type="checkbox"/>	
D31BUBSGBRIDS	string	{res:itgen_sib_d31bubsgbrids}	<input type="checkbox"/>	
D31DATACONTAINERCNT	decimal	{res:itgen_sib_d31datacontainercnt}	<input type="checkbox"/>	
D31MBIO	decimal	{res:itgen_sib_d31mbio}	<input type="checkbox"/>	
D31PARTITIONCNT	decimal	{res:itgen_sib_d31partitioncnt}	<input type="checkbox"/>	
D31PROVIDERCNT	decimal	{res:itgen_sib_d31providercnt}	<input type="checkbox"/>	
D31ROWCNT	decimal	{res:itgen_sib_d31rowcnt}	<input type="checkbox"/>	
D31TABLECNT	decimal	{res:itgen_sib_d31tablecnt}	<input type="checkbox"/>	
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
EXT_ID	string	{res:itgen_sib_ext_id}	<input type="checkbox"/>	
FIRSTNAME	string	First Name	<input type="checkbox"/>	
FORUMS_LOGIN	string	Forums Login	<input type="checkbox"/>	
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
JOBTITLE	string	Job Title	<input type="checkbox"/>	
LANDLINE_NUMBER	string	{res:itgen_sib_landline_number}	<input type="checkbox"/>	
LANGUAGE_CODE	string	Language Code	<input type="checkbox"/>	
LASTNAME	string	Last Name	<input type="checkbox"/>	
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
MOBILE_NUMBER	string	Mobile Number	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
PHONE_NUMBER	string	Phone Number	<input type="checkbox"/>	
PRIMARY_AUDIENCE	string	Primary Audience	<input type="checkbox"/>	
READERS	decimal	Readers	<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
SMS	string	SMS	<input type="checkbox"/>	
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	string	WhatsApp	<input type="checkbox"/>	

### ContactsByListId: Sendinblue Contacts by List ID

Get contacts in a list

Catalog: Brevo

Schema: Contacts

Label: Contacts by List ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
listId	int64	<input checked="" type="checkbox"/>		Id of the list
modifiedSince	datetime	<input type="checkbox"/>		Filter (urlencoded) the contacts modified after a given UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ). Prefer to pass your timezone in date-time format for accurate result.

## Columns of Table Function

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



Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF_TOT	string	{res:itgen_sib_abonnementen_actief_tot}	<input type="checkbox"/>	
ABONNEMENTEN_ACTIEF	string	{res:itgen_sib_abonnementen_actief}	<input type="checkbox"/>	
BLACKLIST	decimal	Blacklist	<input type="checkbox"/>	
BUBS_GBR_ID_MASTER	decimal	{res:itgen_sib_bubs_gbr_id_master}	<input type="checkbox"/>	
BUBS_GBR_ID	decimal	Estate Person ID	<input type="checkbox"/>	
CLICKERS	decimal	Clickers	<input type="checkbox"/>	
COMPANY_NAME	string	Company Name	<input type="checkbox"/>	
CONTACT_TIMEZONE	string	{res:itgen_sib_contact_timezone}	<input type="checkbox"/>	
CONTRACT_CODE	string	Contract Code	<input type="checkbox"/>	
COUNTRY	string	Country	<input type="checkbox"/>	
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	string	{res:itgen_sib_d31applications}	<input type="checkbox"/>	
D31BILLINGIDS	string	{res:itgen_sib_d31billingids}	<input type="checkbox"/>	
D31BUBSGBRIDS	string	{res:itgen_sib_d31bubsgbrids}	<input type="checkbox"/>	
D31DATACONTAINERCNT	decimal	{res:itgen_sib_d31datacontainercnt}	<input type="checkbox"/>	
D31MBIO	decimal	{res:itgen_sib_d31mbio}	<input type="checkbox"/>	
D31PARTITIONCNT	decimal	{res:itgen_sib_d31partitioncnt}	<input type="checkbox"/>	
D31PROVIDERCNT	decimal	{res:itgen_sib_d31providercnt}	<input type="checkbox"/>	
D31ROWCNT	decimal	{res:itgen_sib_d31rowcnt}	<input type="checkbox"/>	
D31TABLECNT	decimal	{res:itgen_sib_d31tablecnt}	<input type="checkbox"/>	
email	string	Email Address	<input type="checkbox"/>	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	<input type="checkbox"/>	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
EXT_ID	string	{res:itgen_sib_ext_id}	<input type="checkbox"/>	
FIRSTNAME	string	First Name	<input type="checkbox"/>	
FORUMS_LOGIN	string	Forums Login	<input type="checkbox"/>	
id	int64	ID	<input type="checkbox"/>	ID of the contact for which you requested the details
JOBTITLE	string	Job Title	<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
LANDLINE_NUMBER	string	{res:itgen_sib_landline_number}	<input type="checkbox"/>	
LANGUAGE_CODE	string	Language Code	<input type="checkbox"/>	
LASTNAME	string	Last Name	<input type="checkbox"/>	
listIds	string	List IDs	<input type="checkbox"/>	
listUnsubscribed	string	Unsubscribed List IDs	<input type="checkbox"/>	
MOBILE_NUMBER	string	Mobile Number	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
PHONE_NUMBER	string	Phone Number	<input type="checkbox"/>	
PRIMARY_AUDIENCE	string	Primary Audience	<input type="checkbox"/>	
READERS	decimal	Readers	<input type="checkbox"/>	
SMS	string	SMS	<input type="checkbox"/>	
smsBlacklisted	boolean	SMS is Blacklisted	<input type="checkbox"/>	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	string	WhatsApp	<input type="checkbox"/>	

### ContactSoftBounceStatisticsByEmail: Sendinblue Contact Soft Bounce Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Soft Bounce Statistics by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event

### **ContactTransactionAttributesByEmail: Sendinblue Contact Transaction Attributes by Email**

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Transaction Attributes by Email

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
orderDate	datetime	Order Date	<input type="checkbox"/>	Date of the order
orderId	int64	Order ID	<input type="checkbox"/>	ID of the order
orderPrice	float	Order Price	<input type="checkbox"/>	Price of the order

### ContactUserUnsubscriptionsByEmail: Sendinblue Contact Unsubscriptions by Email (User Mode)

Get email campaigns' statistics for a contact

Catalog: Brevo

Schema: Contacts

Label: Contact Unsubscriptions by Email (User Mode)

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input checked="" type="checkbox"/>		Email address (urlencoded) of the contact
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign which generated the event
eventTime	datetime	Event Time	<input type="checkbox"/>	UTC date-time of the event
ip	string	IP	<input type="checkbox"/>	IP from which the user has unsubscribed

### FolderById: Sendinblue Folder by ID

Returns a folder's details

Catalog: Brevo

Schema: Contacts

Label: Folder by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
folderId	int64	<input checked="" type="checkbox"/>		id of the folder

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
id	int64	ID	<input type="checkbox"/>	ID of the folder
name	string	Name	<input type="checkbox"/>	Name of the folder
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the folder
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the folder
uniqueSubscribers	int64	#Unique Subscribers	<input type="checkbox"/>	Number of unique contacts in the folder

### Folders: Sendinblue Folders

Get all folders

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Folders

Can retrieve data and change data using insert, update and delete.

## Table Columns

The columns of the table Folders 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 and update.

Name	Data Type	Label	Required	Documentation
id	int64	ID	<input type="checkbox"/>	ID of the folder
name	string	Name	<input type="checkbox"/>	Name of the folder
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the folder
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the folder
uniqueSubscribers	int64	#Unique Subscribers	<input type="checkbox"/>	Number of unique contacts in the folder

**ListById: Sendinblue List by ID**

Get a list's details

Catalog: Brevo

Schema: Contacts

Label: List by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
listId	int64	<input checked="" type="checkbox"/>		Id of the list

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	boolean	Dynamic List	<input type="checkbox"/>	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)
folderId	int64	Folder ID	<input type="checkbox"/>	ID of the folder
id	int64	ID	<input type="checkbox"/>	ID of the list
name	string	Name	<input type="checkbox"/>	Name of the list
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the list

**ListCampaignStatisticsByListId: Sendinblue List Campaign Statistics by List ID**

Get a list's details

Catalog: Brevo

Schema: Contacts

Label: List Campaign Statistics by List ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
listId	int64	<input checked="" type="checkbox"/>		Id of the list

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	boolean	Dynamic List	<input type="checkbox"/>	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)
folderId	int64	Folder ID	<input type="checkbox"/>	ID of the folder
id	int64	ID	<input type="checkbox"/>	ID of the list
name	string	Name	<input type="checkbox"/>	Name of the list
stats_clickers	int64	#Clickers	<input type="checkbox"/>	Number of total clicks for the campaign
stats_complaints	int64	#Complaints	<input type="checkbox"/>	Number of complaints (Spam reports) for the campaign



Name	Data Type	Label	Required	Documentation
stats_deferred	int64	#Deferred	<input type="checkbox"/>	Number of deferred emails for the campaign
stats_delivered	int64	#Delivered	<input type="checkbox"/>	Number of delivered emails for the campaign
stats_hardBounces	int64	#Hard-bounces	<input type="checkbox"/>	Number of harbounce for the campaign
stats_listId	int64	List ID	<input type="checkbox"/>	List Id of email campaign (only in case of get email campaign(s) (not for global stats))
stats_returnBounce	int64	#Return-bounces	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.
stats_sent	int64	#Sent	<input type="checkbox"/>	Number of sent emails for the campaign
stats_softBounces	int64	#Soft-bounces	<input type="checkbox"/>	Number of softbounce for the campaign
stats_trackableViews	int64	#Trackable Views	<input type="checkbox"/>	Recipients without any privacy protection option enabled in their email client
stats_uniqueClicks	int64	#Unique Clicks	<input type="checkbox"/>	Number of unique clicks for the campaign
stats_uniqueViews	int64	#Unique Views	<input type="checkbox"/>	Number of unique openings for the campaign
stats_unsubscriptions	int64	#Unsubscriptions	<input type="checkbox"/>	Number of unsubscription for the campaign
stats_viewed	int64	#Viewed	<input type="checkbox"/>	Number of openings for the campaign
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the list

### Lists: Sendinblue Lists

Get all the lists

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Lists

Can retrieve data and change data using insert, update and delete.

### Table Columns

The columns of the table Lists 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 and update.

Name	Data Type	Label	Required	Documentation
folderId	int64	Folder ID	<input type="checkbox"/>	ID of the folder

Name	Data Type	Label	Required	Documentation
id	int64	ID	<input type="checkbox"/>	ID of the list
name	string	Name	<input type="checkbox"/>	Name of the list
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the list

### ListsByFolderId: Sendinblue Lists by Folder ID

Get lists in a folder

Catalog: Brevo

Schema: Contacts

Label: Lists by Folder ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
folderId	int64	<input checked="" type="checkbox"/>		Id of the folder

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
id	int64	ID	<input type="checkbox"/>	ID of the list
name	string	Name	<input type="checkbox"/>	Name of the list
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the list

Name	Data Type	Label	Required	Documentation
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the list

### Segments: Sendinblue Segments

Get all the segments

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Segments

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

## Table Columns

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

Name	Data Type	Label	Required	Documentation
categoryName	string	{res:itgen_sib_categoryname}	<input type="checkbox"/>	Name of the Segment Category
id	int64	ID	<input type="checkbox"/>	ID of the list
segmentName	string	{res:itgen_sib_segmentname}	<input type="checkbox"/>	Name of the Segment
updatedAt	string	{res:itgen_sib_updatedat}	<input type="checkbox"/>	Update UTC date-time of the segment (YYYY-MM-DDTHH:mm:ss.SSSZ)

### 3.1.3 Schema: EmailCampaigns

#### 3.1.3.1 Tables

##### AbTestCampaignResultById: Sendinblue A/B Test Campaign Result by ID

Get an A/B test email campaign results Obtain winning version of an A/B test email campaign

Catalog: Brevo

Schema: EmailCampaigns

Label: A/B Test Campaign Result by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
campaignId	int64	<input checked="" type="checkbox"/>		Id of the A/B test campaign

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
clickRate	string	Clickrate	<input type="checkbox"/>	Click rate for current winning version
openRate	string	Open Rate	<input type="checkbox"/>	Open rate for current winning version
statistics_clicks_Version_A	string	{res:itgen_sib_statistics_clicks_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_clicks_Version_B	string	{res:itgen_sib_statistics_clicks_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_complaints_Version_A	string	{res:itgen_sib_statistics_complaints_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_complaints_Version_B	string	{res:itgen_sib_statistics_complaints_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_hardBounces_Version_A	string	{res:itgen_sib_statistics_hardbounces_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_hardBounces_Version_B	string	{res:itgen_sib_statistics_hardbounces_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_openers_Version_A	string	{res:itgen_sib_statistics_openers_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_openers_Version_B	string	{res:itgen_sib_statistics_openers_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_softBounces_Version_A	string	{res:itgen_sib_statistics_softbounces_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_softBounces_Version_B	string	{res:itgen_sib_statistics_softbounces_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_unsubscribed_Version_A	string	{res:itgen_sib_statistics_unsub-	<input type="checkbox"/>	percentage of an event for version A

Name	Data Type	Label	Required	Documentation
		scribed_version_a}		
statistics_unsubscribed_Version_B	string	{res:itgen_sib_statistics_unsubscribed_version_b}	<input type="checkbox"/>	percentage of an event for version B
winningCriteria	string	Winning Criteria	<input type="checkbox"/>	Criteria chosen for winning version (Open/Click)
winningSubjectLine	string	Winning Subject Line	<input type="checkbox"/>	Subject Line of current winning version
winningVersion	string	Winning Version	<input type="checkbox"/>	Winning Campaign Info. pending = Campaign has been picked for sending and winning version is yet to be decided, tie = A tie happened between both the versions, notAvailable = Campaign has not yet been picked for sending.
winningVersionRate	string	Winning Version Rate	<input type="checkbox"/>	Open/Click rate for the winner version

### EmailCampaignById: Sendinblue Email Campaign by ID

Get an email campaign report

Catalog: Brevo

Schema: EmailCampaigns

Label: Email Campaign by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
campaignId	int64	<input checked="" type="checkbox"/>		Id of the campaign
statistics	string	<input type="checkbox"/>		Filter on type of the statistics required. Example <code>**globalStats**</code>

Name	Data Type	Required	Default Value	Documentation
				value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain, statsByDevice, statsByBrowser)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	<input type="checkbox"/>	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
footer	string	Footer	<input type="checkbox"/>	Footer of the campaign
header	string	Header	<input type="checkbox"/>	Header of the campaign
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the campaign
id	int64	ID	<input type="checkbox"/>	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	<input type="checkbox"/>	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true means image can be embedded, in the email.
mirrorActive	boolean	Mirror Active	<input type="checkbox"/>	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the campaign
recipients	string	Recipients	<input type="checkbox"/>	
recurring	boolean	Recurring	<input type="checkbox"/>	FOR TRIGGER ONLY ! Type of trigger campaign. recurring = false means contact can receive the same Trigger campaign only once, & recurring = true means contact can receive the same Trigger campaign several times
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.

Name	Data Type	Label	Required	Documentation
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	<input type="checkbox"/>	It is true if you have chosen to send your campaign at best time, otherwise it is false
sender_email	string	Sender Email Address	<input type="checkbox"/>	Sender email of the campaign
sender_id	int64	Sender ID	<input type="checkbox"/>	Sender id of the campaign
sender_name	string	Sender Name	<input type="checkbox"/>	Sender name of the campaign
sentDate	datetime	Sent Date	<input type="checkbox"/>	Sent UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	<input type="checkbox"/>	Link to share the campaign on social medias
splitRule	int64	Split Rule	<input type="checkbox"/>	The size of your ab-test groups. Only available if 'abTesting' flag of the campaign is 'true'
statistics	string	Statistics	<input type="checkbox"/>	
status	string	Status	<input type="checkbox"/>	Status of the campaign
subject	string	Subject	<input type="checkbox"/>	Subject of the campaign. Only available if 'abTesting' flag of the campaign is 'false'
subjectA	string	Subject A	<input type="checkbox"/>	Subject A of the ab-test campaign. Only available if 'abTesting' flag of the campaign is 'true'
subjectB	string	Subject B	<input type="checkbox"/>	Subject B of the ab-test campaign. Only available if 'abTesting' flag of the campaign is 'true'
tag	string	Tag	<input type="checkbox"/>	Tag of the campaign
testSent	boolean	Test Sent	<input type="checkbox"/>	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field of the campaign
type	string	Type	<input type="checkbox"/>	Type of campaign
winnerCriteria	string	Winner Criteria	<input type="checkbox"/>	Criteria for the winning version. Only available if 'abTesting' flag of the campaign is 'true'
winnerDelay	int64	Winner Delay	<input type="checkbox"/>	The duration of the test in hours at the end of which the winning version will be sent. Only available if 'abTesting' flag of the campaign is 'true'

## EmailCampaignRecipientExclusionLists: Sendinblue Email Campaign Recipient Exclusion Lists

Return all your created email campaigns

Catalog: Brevo

Schema: EmailCampaigns

Label: Email Campaign Recipient Exclusion Lists

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
excludeHtmlContent	boolean	<input type="checkbox"/>		Use this flag to exclude htmlContent from the response body. If set to <b>true</b> , htmlContent field will be returned as empty string in the response body (Waarden: true, false)
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )



Name	Data Type	Required	Default Value	Documentation
statistics	string	<input type="checkbox"/>		Filter on type of the statistics required. Example <b>**globalStats**</b> value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain)
status	string	<input type="checkbox"/>		Filter on the status of the campaign (Waarden: suspended, archive, sent, queued, draft, in-Process)
type	string	<input type="checkbox"/>		Filter on the type of the campaigns (Waarden: classic, trigger)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	<input type="checkbox"/>	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
footer	string	Footer	<input type="checkbox"/>	Footer of the campaign
header	string	Header	<input type="checkbox"/>	Header of the campaign
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the campaign
id	int64	ID	<input type="checkbox"/>	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	<input type="checkbox"/>	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true means image can be embedded, in the email.
lists	string	{res:itgen_sib_lists}	<input type="checkbox"/>	
mirrorActive	boolean	Mirror Active	<input type="checkbox"/>	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the campaign
recurring	boolean	Recurring	<input type="checkbox"/>	FOR TRIGGER ONLY ! Type of trigger campaign. recurring = false means contact can receive the same Trigger campaign only

Name	Data Type	Label	Required	Documentation
				once, & recurring = true means contact can receive the same Trigger campaign several times
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	<input type="checkbox"/>	It is true if you have chosen to send your campaign at best time, otherwise it is false
sender_email	string	Sender Email Address	<input type="checkbox"/>	Sender email of the campaign
sender_id	int64	Sender ID	<input type="checkbox"/>	Sender id of the campaign
sender_name	string	Sender Name	<input type="checkbox"/>	Sender name of the campaign
sentDate	datetime	Sent Date	<input type="checkbox"/>	Sent UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	<input type="checkbox"/>	Link to share the campaign on social medias
splitRule	int64	Split Rule	<input type="checkbox"/>	The size of your ab-test groups. Only available if 'abTesting' flag of the campaign is 'true'
statistics_globalStats	string	Global Statistics	<input type="checkbox"/>	Overall statistics of the campaign
statistics_linksStats	string	Link Statistics	<input type="checkbox"/>	Statistics about the number of clicks for the links
statistics_mirrorClick	int64	#Mirror Clicks	<input type="checkbox"/>	Number of clicks on mirror link
statistics_remaining	int64	#Remaining	<input type="checkbox"/>	Number of remaining emails to send
statistics_statsByBrowser	string	Statistics by Browser	<input type="checkbox"/>	Statistics about the campaign on the basis of various browsers
statistics_statsByDomain	string	Statistics by Domain	<input type="checkbox"/>	
status	string	Status	<input type="checkbox"/>	Status of the campaign
subject	string	Subject	<input type="checkbox"/>	Subject of the campaign. Only available if 'abTesting' flag of the campaign is 'false'
subjectA	string	Subject A	<input type="checkbox"/>	Subject A of the ab-test campaign. Only available if 'abTesting' flag of the campaign is 'true'
subjectB	string	Subject B	<input type="checkbox"/>	Subject B of the ab-test campaign. Only available if 'abTesting' flag of the campaign is 'true'
tag	string	Tag	<input type="checkbox"/>	Tag of the campaign
testSent	boolean	Test Sent	<input type="checkbox"/>	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)

Name	Data Type	Label	Required	Documentation
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field of the campaign
type	string	Type	<input type="checkbox"/>	Type of campaign
winnerCriteria	string	Winner Criteria	<input type="checkbox"/>	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`
winnerDelay	int64	Winner Delay	<input type="checkbox"/>	The duration of the test in hours at the end of which the winning version will be sent. Only available if `abTesting` flag of the campaign is `true`

### EmailCampaignRecipientLists: Sendinblue Email Campaign Recipient Lists

Return all your created email campaigns

Catalog: Brevo

Schema: EmailCampaigns

Label: Email Campaign Recipient Lists

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

### Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )

Name	Data Type	Required	Default Value	Documentation
excludeHtmlContent	boolean	<input type="checkbox"/>		Use this flag to exclude htmlContent from the response body. If set to <b>true</b> , htmlContent field will be returned as empty string in the response body (Waarden: true, false)
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
statistics	string	<input type="checkbox"/>		Filter on type of the statistics required. Example <b>globalStats</b> value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain)
status	string	<input type="checkbox"/>		Filter on the status of the campaign (Waarden: suspended, archive, sent, queued, draft, inProcess)
type	string	<input type="checkbox"/>		Filter on the type of the campaigns (Waarden: classic, trigger)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	<input type="checkbox"/>	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
exclusionLists	string	{res:itgen_sib_exclusionlists}	<input type="checkbox"/>	
footer	string	Footer	<input type="checkbox"/>	Footer of the campaign
header	string	Header	<input type="checkbox"/>	Header of the campaign
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the campaign
id	int64	ID	<input type="checkbox"/>	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	<input type="checkbox"/>	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true

Name	Data Type	Label	Required	Documentation
				means image can be embedded, in the email.
mirrorActive	boolean	Mirror Active	<input type="checkbox"/>	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the campaign
recurring	boolean	Recurring	<input type="checkbox"/>	FOR TRIGGER ONLY ! Type of trigger campaign. recurring = false means contact can receive the same Trigger campaign only once, & recurring = true means contact can receive the same Trigger campaign several times
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	<input type="checkbox"/>	It is true if you have chosen to send your campaign at best time, otherwise it is false
sender_email	string	Sender Email Address	<input type="checkbox"/>	Sender email of the campaign
sender_id	int64	Sender ID	<input type="checkbox"/>	Sender id of the campaign
sender_name	string	Sender Name	<input type="checkbox"/>	Sender name of the campaign
sentDate	datetime	Sent Date	<input type="checkbox"/>	Sent UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	<input type="checkbox"/>	Link to share the campaign on social medias
splitRule	int64	Split Rule	<input type="checkbox"/>	The size of your ab-test groups. Only available if 'abTesting' flag of the campaign is 'true'
statistics_globalStats	string	Global Statistics	<input type="checkbox"/>	Overall statistics of the campaign
statistics_linksStats	string	Link Statistics	<input type="checkbox"/>	Statistics about the number of clicks for the links
statistics_mirrorClick	int64	#Mirror Clicks	<input type="checkbox"/>	Number of clicks on mirror link
statistics_remaining	int64	#Remaining	<input type="checkbox"/>	Number of remaining emails to send
statistics_statsByBrowser	string	Statistics by Browser	<input type="checkbox"/>	Statistics about the campaign on the basis of various browsers
statistics_statsByDomain	string	Statistics by Domain	<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
status	string	Status	<input type="checkbox"/>	Status of the campaign
subject	string	Subject	<input type="checkbox"/>	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
subjectA	string	Subject A	<input type="checkbox"/>	Subject A of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
subjectB	string	Subject B	<input type="checkbox"/>	Subject B of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
tag	string	Tag	<input type="checkbox"/>	Tag of the campaign
testSent	boolean	Test Sent	<input type="checkbox"/>	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field of the campaign
type	string	Type	<input type="checkbox"/>	Type of campaign
winnerCriteria	string	Winner Criteria	<input type="checkbox"/>	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`
winnerDelay	int64	Winner Delay	<input type="checkbox"/>	The duration of the test in hours at the end of which the winning version will be sent. Only available if `abTesting` flag of the campaign is `true`

### EmailCampaigns: Sendinblue Email Campaigns

Return all your created email campaigns

Catalog: Brevo

Schema: EmailCampaigns

Primary Keys: id

Label: Email Campaigns

Can retrieve data and change data using insert, update and delete.

### Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
excludeHtmlContent	boolean	<input type="checkbox"/>		Use this flag to exclude htmlContent from the response body. If set to <b>true</b> , htmlContent field will be returned as empty string in the response body (Waarden: true, false)
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
statistics	string	<input type="checkbox"/>		Filter on type of the statistics required. Example <b>globalStats</b> value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain)
status	string	<input type="checkbox"/>		Filter on the status of the campaign (Waarden: suspended, archive, sent, queued, draft, in-Process)
type	string	<input type="checkbox"/>		Filter on the type of the campaigns (Waarden: classic, trigger)

## Columns of Table Function

The columns of the table function EmailCampaigns 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 and update.

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	<input type="checkbox"/>	Status of A/B Test for the campaign. abTesting = false means it

Name	Data Type	Label	Required	Documentation
				is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
footer	string	Footer	<input type="checkbox"/>	Footer of the campaign
header	string	Header	<input type="checkbox"/>	Header of the campaign
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the campaign
id	int64	ID	<input type="checkbox"/>	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	<input type="checkbox"/>	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true means image can be embedded, in the email.
mirrorActive	boolean	Mirror Active	<input type="checkbox"/>	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the campaign
recipients_exclusionLists	string	{res:itgen_sib_recipients_exclusionlists}	<input type="checkbox"/>	
recipients_lists	string	{res:itgen_sib_recipients_lists}	<input type="checkbox"/>	
recurring	boolean	Recurring	<input type="checkbox"/>	FOR TRIGGER ONLY ! Type of trigger campaign.recurring = false means contact can receive the same Trigger campaign only once, & recurring = true means contact can receive the same Trigger campaign several times
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	<input type="checkbox"/>	It is true if you have chosen to send your campaign at best time, otherwise it is false
sender_email	string	Sender Email Address	<input type="checkbox"/>	Sender email of the campaign
sender_id	int64	Sender ID	<input type="checkbox"/>	Sender id of the campaign
sender_name	string	Sender Name	<input type="checkbox"/>	Sender name of the campaign
sentDate	datetime	Sent Date	<input type="checkbox"/>	Sent UTC date-time of the campaign (YYYY-MM-



Name	Data Type	Label	Required	Documentation
				DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	<input type="checkbox"/>	Link to share the campaign on social medias
splitRule	int64	Split Rule	<input type="checkbox"/>	The size of your ab-test groups. Only available if `abTesting` flag of the campaign is `true`
statistics_globalStats	string	Global Statistics	<input type="checkbox"/>	Overall statistics of the campaign
statistics_linksStats	string	Link Statistics	<input type="checkbox"/>	Statistics about the number of clicks for the links
statistics_mirrorClick	int64	#Mirror Clicks	<input type="checkbox"/>	Number of clicks on mirror link
statistics_remaining	int64	#Remaining	<input type="checkbox"/>	Number of remaining emails to send
statistics_statsByBrowser	string	Statistics by Browser	<input type="checkbox"/>	Statistics about the campaign on the basis of various browsers
statistics_statsByDomain	string	Statistics by Domain	<input type="checkbox"/>	
status	string	Status	<input type="checkbox"/>	Status of the campaign
subject	string	Subject	<input type="checkbox"/>	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
subjectA	string	Subject A	<input type="checkbox"/>	Subject A of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
subjectB	string	Subject B	<input type="checkbox"/>	Subject B of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
tag	string	Tag	<input type="checkbox"/>	Tag of the campaign
testSent	boolean	Test Sent	<input type="checkbox"/>	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field of the campaign
type	string	Type	<input type="checkbox"/>	Type of campaign
winnerCriteria	string	Winner Criteria	<input type="checkbox"/>	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`
winnerDelay	int64	Winner Delay	<input type="checkbox"/>	The duration of the test in hours at the end of which the winning version will be sent. Only available if `abTesting` flag of the campaign is `true`

### EmailCampaignStatistics: Sendinblue Email Campaign Statistics

Return all your created email campaigns

Catalog: Brevo

Schema: EmailCampaigns

Label: Email Campaign Statistics

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
excludeHtmlContent	boolean	<input type="checkbox"/>		Use this flag to exclude htmlContent from the response body. If set to <b>true</b> , htmlContent field will be returned as empty string in the response body (Waarden: true, false)
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
statistics	string	<input type="checkbox"/>		Filter on type of the statistics required. Example <b>globalStats</b> value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain)

Name	Data Type	Required	Default Value	Documentation
status	string	<input type="checkbox"/>		Filter on the status of the campaign (Waarden: suspended, archive, sent, queued, draft, in-Process)
type	string	<input type="checkbox"/>		Filter on the type of the campaigns (Waarden: classic, trigger)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	<input type="checkbox"/>	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
footer	string	Footer	<input type="checkbox"/>	Footer of the campaign
globalStats	string	Global Statistics	<input type="checkbox"/>	Overall statistics of the campaign
header	string	Header	<input type="checkbox"/>	Header of the campaign
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the campaign
id	int64	ID	<input type="checkbox"/>	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	<input type="checkbox"/>	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true means image can be embedded, in the email.
linksStats	string	Link Statistics	<input type="checkbox"/>	Statistics about the number of clicks for the links
mirrorActive	boolean	Mirror Active	<input type="checkbox"/>	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
mirrorClick	int64	Mirror Click	<input type="checkbox"/>	Number of clicks on mirror link
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the campaign
recipients_exclusionLists	string	{res:itgen_sib_recipients_exclusionlists}	<input type="checkbox"/>	
recipients_lists	string	{res:itgen_sib_recipients_lists}	<input type="checkbox"/>	
recurring	boolean	Recurring	<input type="checkbox"/>	FOR TRIGGER ONLY ! Type of trigger campaign.recurring =

Name	Data Type	Label	Required	Documentation
				false means contact can receive the same Trigger campaign only once, & recurring = true means contact can receive the same Trigger campaign several times
remaining	int64	Remaining	<input type="checkbox"/>	Number of remaining emails to send
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	<input type="checkbox"/>	It is true if you have chosen to send your campaign at best time, otherwise it is false
sender_email	string	Sender Email Address	<input type="checkbox"/>	Sender email of the campaign
sender_id	int64	Sender ID	<input type="checkbox"/>	Sender id of the campaign
sender_name	string	Sender Name	<input type="checkbox"/>	Sender name of the campaign
sentDate	datetime	Sent Date	<input type="checkbox"/>	Sent UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	<input type="checkbox"/>	Link to share the campaign on social medias
splitRule	int64	Split Rule	<input type="checkbox"/>	The size of your ab-test groups. Only available if 'abTesting' flag of the campaign is 'true'
statsByBrowser	string	Statistics by Browser	<input type="checkbox"/>	Statistics about the campaign on the basis of various browsers
statsByDomain	string	Statistics by Domain	<input type="checkbox"/>	
status	string	Status	<input type="checkbox"/>	Status of the campaign
subject	string	Subject	<input type="checkbox"/>	Subject of the campaign. Only available if 'abTesting' flag of the campaign is 'false'
subjectA	string	Subject A	<input type="checkbox"/>	Subject A of the ab-test campaign. Only available if 'abTesting' flag of the campaign is 'true'
subjectB	string	Subject B	<input type="checkbox"/>	Subject B of the ab-test campaign. Only available if 'abTesting' flag of the campaign is 'true'
tag	string	Tag	<input type="checkbox"/>	Tag of the campaign
testSent	boolean	Test Sent	<input type="checkbox"/>	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field of the campaign
type	string	Type	<input type="checkbox"/>	Type of campaign

Name	Data Type	Label	Required	Documentation
w innerCriteria	string	Winner Criteria	<input type="checkbox"/>	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`
w innerDelay	int64	Winner Delay	<input type="checkbox"/>	The duration of the test in hours at the end of which the winning version will be sent. Only available if `abTesting` flag of the campaign is `true`

### SharedTemplateUrlByCampaignId: Sendinblue Shared URL Template by Campaign ID

Get a shared template urlGet a unique URL to share & import an email template from one Sendinblue account to another.

Catalog: Brevo

Schema: EmailCampaigns

Label: Shared URL Template by Campaign ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
campaignId	int64	<input checked="" type="checkbox"/>		Id of the campaign or template

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
sharedUrl	string	Shared URL	<input type="checkbox"/>	A unique URL for the email campaign or transactional template.

Name	Data Type	Label	Required	Documentation
				This URL can be shared with other Sendinblue users.

### 3.1.4 Schema: Native

#### 3.1.4.1 Tables

#### NATIVEPLATFORMSCALARREQUESTS: Sendinblue Native Platform Scalar Requests

{res:itgen\_native\_platform\_scalar\_requests\_desc}

Catalog: Brevo

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 Brevo 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 Brevo 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.

Name	Data Type	Label	Required	Documentation
BLOB_PREFERRED	boolean	BLOB Preferred	<input checked="" type="checkbox"/>	Indicator whether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	<input type="checkbox"/>	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	<input type="checkbox"/>	
DATE_ENDED	datetime	End Date	<input checked="" type="checkbox"/>	
DATE_STARTED	datetime	Start Date	<input checked="" type="checkbox"/>	
DRY_RUN	boolean	Run without Actions	<input checked="" type="checkbox"/>	
DURATION_MS	int64	Duration (ms)	<input checked="" type="checkbox"/>	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	<input type="checkbox"/>	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	<input type="checkbox"/>	
FAIL_ON_ERROR	boolean	Fail on Error	<input checked="" type="checkbox"/>	Whether to raise an exception when processing the native re-

Name	Data Type	Label	Required	Documentation
				quest triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	<input type="checkbox"/>	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	<input type="checkbox"/>	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	<input type="checkbox"/>	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	<input type="checkbox"/>	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	<input type="checkbox"/>	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	<input type="checkbox"/>	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	string(30)	HTTP Method	<input type="checkbox"/>	
HTTP_STATUS_CODE	int32	HTTP Status Code	<input type="checkbox"/>	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	<input type="checkbox"/>	
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	<input type="checkbox"/>	
PAYLOAD_TEXT	string	Payload	<input type="checkbox"/>	
RESULT_BLOB	byte[]	Result BLOB	<input type="checkbox"/>	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	<input type="checkbox"/>	
RESULT_NUMBER	decimal	Result Number	<input type="checkbox"/>	
RESULT_TEXT	string	Result Text	<input type="checkbox"/>	
SUCCESSFUL	boolean	Successful	<input checked="" type="checkbox"/>	
TIMEOUT_SEC	int32	Timeout (sec)	<input type="checkbox"/>	Timeout in seconds.
TRANSACTION_ID	int64	Transaction ID	<input checked="" type="checkbox"/>	Incrementing ID of the transaction.
URL	string(4000)	URL	<input type="checkbox"/>	

### 3.1.5 Schema: Process

#### 3.1.5.1 Tables

##### BackgroundProcessById: Sendinblue Background Process by ID

Return the informations for a process

Catalog: Brevo

Schema: Process

Label: Background Process by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
processId	int64	<input checked="" type="checkbox"/>		Id of the process

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
export_url	string	Export URL	<input type="checkbox"/>	URL on which send export the of contacts once the process is completed
id	int64	ID	<input type="checkbox"/>	Id of the process
name	string	Name	<input type="checkbox"/>	Process name
status	string	Status	<input type="checkbox"/>	Status of the process

### BackgroundProcesses: Sendinblue Background Processes

Return all the processes for your account

Catalog: Brevo

Schema: Process

Primary Keys: id

Label: Background Processes

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

## Table Columns

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



Name	Data Type	Label	Required	Documentation
export_url	string	Export URL	<input type="checkbox"/>	URL on which send export the of contacts once the process is completed
id	int64	ID	<input type="checkbox"/>	Id of the process
name	string	Name	<input type="checkbox"/>	Process name
status	string	Status	<input type="checkbox"/>	Status of the process

### 3.1.6 Schema: Senders

#### 3.1.6.1 Tables

##### IpAddresses: Sendinblue IP Addresses

Get all the dedicated IPs for your account

Catalog: Brevo

Schema: Senders

Primary Keys: id

Label: IP Addresses

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

## Table Columns

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

Name	Data Type	Label	Required	Documentation
active	boolean	Active	<input type="checkbox"/>	Status of the IP (true=active, false=inactive)
domain	string	Domain	<input type="checkbox"/>	Domain associated to the IP
id	int64	ID	<input type="checkbox"/>	ID of the dedicated IP
ip	string	IP	<input type="checkbox"/>	Dedicated IP

##### IpAddressesBySenderId: Sendinblue IP Addresses by Send ID

Get all the dedicated IPs for a sender

Catalog: Brevo

Schema: Senders

Label: IP Addresses by Send ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
senderId	int64	<input checked="" type="checkbox"/>		Id of the sender

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
domain	string	Domain	<input type="checkbox"/>	Domain associated to the IP
id	int64	ID	<input type="checkbox"/>	ID of the dedicated IP
ip	string	IP	<input type="checkbox"/>	Dedicated IP
weight	int64	Weight	<input type="checkbox"/>	Weight of the IP

### SenderIdAddresses: Sendinblue Sender IP Addresses

Get the list of all your senders

Catalog: Brevo

Schema: Senders

Label: Sender IP Addresses

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

## Parameters of Table Function

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

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four para-

meters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
domain	string	<input type="checkbox"/>		Filter your senders for a specific domain
ip	string	<input type="checkbox"/>		Filter your senders for a specific ip (available for dedicated IP usage only)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
active	boolean	Active	<input type="checkbox"/>	Status of sender (true=activated, false=deactivated)
domain	string	Domain	<input type="checkbox"/>	Domain of the IP
email	string	Email Address	<input type="checkbox"/>	From Email associated to the sender
id	int64	ID	<input type="checkbox"/>	Id of the sender
ip	string	IP	<input type="checkbox"/>	Dedicated IP available in your account
name	string	Name	<input type="checkbox"/>	From Name associated to the sender
weight	int64	Weight	<input type="checkbox"/>	Weight of the IP for this sender

### Senders: Sendinblue Senders

Get the list of all your senders

Catalog: Brevo

Schema: Senders

Primary Keys: id

Label: Senders

Can retrieve data and change data using insert, update and delete.

## Parameters of Table Function

The following parameters can be used to control the behaviour of the table function `Senders`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour.

Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
domain	string	<input type="checkbox"/>		Filter your senders for a specific domain
ip	string	<input type="checkbox"/>		Filter your senders for a specific ip (available for dedicated IP usage only)

## Columns of Table Function

The columns of the table function Senders 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 and update.

Name	Data Type	Label	Required	Documentation
active	boolean	Active	<input type="checkbox"/>	Status of sender (true=activated, false=deactivated)
email	string	Email Address	<input type="checkbox"/>	From Email associated to the sender
id	int64	ID	<input type="checkbox"/>	Id of the sender
name	string	Name	<input type="checkbox"/>	From Name associated to the sender

### 3.1.7 Schema: SMSCampaigns

#### 3.1.7.1 Tables

##### SmsCampaignById: Sendinblue SMS Campaign by ID

Get an SMS campaign

Catalog: Brevo

Schema: SMSCampaigns

Label: SMS Campaign by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
campaignId	int64	<input checked="" type="checkbox"/>		id of the SMS campaign

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	string	Content	<input type="checkbox"/>	Content of the SMS Campaign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	<input type="checkbox"/>	ID of the SMS Campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the SMS Campaign
recipients	string	Recipients	<input type="checkbox"/>	
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
sender	string	Sender	<input type="checkbox"/>	Sender of the SMS Campaign
statistics	string	Statistics	<input type="checkbox"/>	
status	string	Status	<input type="checkbox"/>	Status of the SMS Campaign

### **SmsCampaignRecipientExclusionLists: Sendinblue SMS Campaign Recipient Exclusion Lists**

Returns the information for all your created SMS campaigns

Catalog: Brevo

Schema: SMSCampaigns

Label: SMS Campaign Recipient Exclusion Lists

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
status	string	<input type="checkbox"/>		Status of campaign. (Waarden: suspended, archive, sent, queued, draft, inProcess)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	string	Content	<input type="checkbox"/>	Content of the SMS Campaign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the SMS campaign (YYYY-MM-

Name	Data Type	Label	Required	Documentation
				DDTHH:mm:ss.SSSZ)
id	int64	ID	<input type="checkbox"/>	ID of the SMS Campaign
lists	string	{res:itgen_sib_lists}	<input type="checkbox"/>	
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the SMS Campaign
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
sender	string	Sender	<input type="checkbox"/>	Sender of the SMS Campaign
statistics_answered	int64	#Answered	<input type="checkbox"/>	Number of replies to the SMS
statistics_delivered	int64	#Delivered	<input type="checkbox"/>	Number of delivered SMS
statistics_hardBounces	int64	#Hard-bounces	<input type="checkbox"/>	Number of hardbounced SMS
statistics_processing	int64	#Processing	<input type="checkbox"/>	Number of processing SMS
statistics_sent	int64	#Sent	<input type="checkbox"/>	Number of sent SMS
statistics_softBounces	int64	#Soft-bounces	<input type="checkbox"/>	Number of softbounced SMS
statistics_unsubcriptions	int64	#Unsubcriptions	<input type="checkbox"/>	Number of unsubscription SMS
status	string	Status	<input type="checkbox"/>	Status of the SMS Campaign

### SmsCampaignRecipientLists: Sendinblue SMS Campaign Recipient Lists

Returns the information for all your created SMS campaigns

Catalog: Brevo

Schema: SMSCampaigns

Label: SMS Campaign Recipient Lists

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
status	string	<input type="checkbox"/>		Status of campaign. (Waarden: suspended, archive, sent, queued, draft, inProcess)

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	string	Content	<input type="checkbox"/>	Content of the SMS Campaign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
exclusionLists	string	{res:itgen_sib_exclusionlists}	<input type="checkbox"/>	
id	int64	ID	<input type="checkbox"/>	ID of the SMS Campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the SMS Campaign
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
sender	string	Sender	<input type="checkbox"/>	Sender of the SMS Campaign
statistics_answered	int64	#Answered	<input type="checkbox"/>	Number of replies to the SMS
statistics_delivered	int64	#Delivered	<input type="checkbox"/>	Number of delivered SMS
statistics_hardBounces	int64	#Hard-bounces	<input type="checkbox"/>	Number of hardbounced SMS



Name	Data Type	Label	Required	Documentation
statistics_processing	int64	#Processing	<input type="checkbox"/>	Number of processing SMS
statistics_sent	int64	#Sent	<input type="checkbox"/>	Number of sent SMS
statistics_softBounces	int64	#Soft-bounces	<input type="checkbox"/>	Number of softbounced SMS
statistics_unsubscriptions	int64	#Unsubscriptions	<input type="checkbox"/>	Number of unsubscription SMS
status	string	Status	<input type="checkbox"/>	Status of the SMS Campaign

### SmsCampaigns: Sendinblue SMS Campaigns

Returns the information for all your created SMS campaigns

Catalog: Brevo

Schema: SMSCampaigns

Primary Keys: id

Label: SMS Campaigns

Can retrieve data and change data using insert, update and delete.

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result ( only available if either 'status' not passed and if passed is set to 'sent' )
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result ( only

Name	Data Type	Required	Default Value	Documentation
				available if either 'status' not passed and if passed is set to 'sent' )
status	string	<input type="checkbox"/>		Status of campaign. (Waarden: suspended, archive, sent, queued, draft, inProcess)

## Columns of Table Function

The columns of the table function SmsCampaigns 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 and update.

Name	Data Type	Label	Required	Documentation
content	string	Content	<input type="checkbox"/>	Content of the SMS Campaign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	<input type="checkbox"/>	ID of the SMS Campaign
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the SMS Campaign
recipients_exclusionLists	string	{res:itgen_sib_recipients_exclusionlists}	<input type="checkbox"/>	
recipients_lists	string	{res:itgen_sib_recipients_lists}	<input type="checkbox"/>	
scheduledAt	datetime	Schedule Date	<input type="checkbox"/>	UTC date-time on w hich SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
sender	string	Sender	<input type="checkbox"/>	Sender of the SMS Campaign
statistics_answered	int64	#Answered	<input type="checkbox"/>	Number of replies to the SMS
statistics_delivered	int64	#Delivered	<input type="checkbox"/>	Number of delivered SMS
statistics_hardBounces	int64	#Hard-bounces	<input type="checkbox"/>	Number of hardbounced SMS
statistics_processing	int64	#Processing	<input type="checkbox"/>	Number of processing SMS
statistics_sent	int64	#Sent	<input type="checkbox"/>	Number of sent SMS
statistics_softBounces	int64	#Soft-bounces	<input type="checkbox"/>	Number of softbounced SMS
statistics_unsubscriptions	int64	#Unsubscriptions	<input type="checkbox"/>	Number of unsubscription SMS
status	string	Status	<input type="checkbox"/>	Status of the SMS Campaign

### 3.1.8 Schema: SMTP

#### 3.1.8.1 Tables

##### BlockedDomains: Sendinblue Blocked Domains

Get the list of blocked domainsGet the list of blocked domains

Catalog: Brevo

Schema: SMTP

Label: Blocked Domains

Retrieve: true

Insert: true

Update: false

Delete: false

## Table Columns

The columns of the table BlockedDomains 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.

Name	Data Type	Label	Required	Documentation
TEXT	string	Text	<input type="checkbox"/>	name of blocked domain

### EmailEventReport: Sendinblue Email Event Report

Get all your transactional email activity (unaggregated events)

Catalog: Brevo

Schema: SMTP

Label: Email Event Report

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
days	int64	<input type="checkbox"/>		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'

Name	Data Type	Required	Default Value	Documentation
email	string	<input type="checkbox"/>		Filter the report for a specific email addresses
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
event	string	<input type="checkbox"/>		Filter the report for a specific event type (Waarden: bounces, hardBounces, softBounces, delivered, spam, requests, opened, clicks, invalid, deferred, blocked, unsubscribed)
messageId	string	<input type="checkbox"/>		Filter on a specific message id
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date of the report (YYYY-MM-DD). Must be lower than equal to endDate
tags	string	<input type="checkbox"/>		Filter the report for tags (serialized and urlencoded array)
templateId	int64	<input type="checkbox"/>		Filter on a specific template id

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	<input type="checkbox"/>	UTC date-time on which the event has been generated
email	string	Email Address	<input type="checkbox"/>	Email address which generates the event
event	string	Event	<input type="checkbox"/>	Event which occurred
from	string	From	<input type="checkbox"/>	Sender email from which the emails are sent
ip	string	IP	<input type="checkbox"/>	IP from which the user has opened the email or clicked on the link (only available if the event is opened or clicks)
link	string	Link	<input type="checkbox"/>	The link which is sent to the user (only available if the event is requests or opened or clicks)
messageId	string	Message ID	<input type="checkbox"/>	Message ID which generated the event
reason	string	Reason	<input type="checkbox"/>	Reason of bounce (only available if the event is hardbounce or softbounce)
subject	string	Subject	<input type="checkbox"/>	Subject of the event
tag	string	Tag	<input type="checkbox"/>	Tag of the email which generated the event

**SmtpReport: Sendinblue SMTP Report**

Get your transactional email activity aggregated per day

Catalog: Brevo

Schema: SMTP

Label: SMTP Report

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
days	int64	<input type="checkbox"/>		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date of the report (YYYY-MM-DD)
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date of the report (YYYY-MM-DD)
tag	string	<input type="checkbox"/>		Tag of the emails

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
blocked	int64	Blocked	<input type="checkbox"/>	Number of blocked emails for the date
clicks	int64	Clicks	<input type="checkbox"/>	Number of clicks for the date
date	datetime	Date	<input type="checkbox"/>	Date of the statistics

Name	Data Type	Label	Required	Documentation
delivered	int64	Delivered	<input type="checkbox"/>	Number of delivered emails for the date
hardBounces	int64	Hard-bounces	<input type="checkbox"/>	Number of hardbounces for the date
invalid	int64	Invalid	<input type="checkbox"/>	Number of invalid emails for the date
opens	int64	Opens	<input type="checkbox"/>	Number of openings for the date
requests	int64	Requests	<input type="checkbox"/>	Number of requests for the date
softBounces	int64	Soft-bounces	<input type="checkbox"/>	Number of softbounces for the date
spamReports	int64	Spam Reports	<input type="checkbox"/>	Number of complaints (spam reports) for the date
uniqueClicks	int64	#Unique Clicks	<input type="checkbox"/>	Number of unique clicks for the date
uniqueOpens	int64	#Unique Opens	<input type="checkbox"/>	Number of unique openings for the date
unsubscribed	int64	Unsubscribed	<input type="checkbox"/>	Number of unsubscribed emails for the date

### Smtprports: Sendinblue SMTP Reports

Get your transactional email activity aggregated over a period of time

Catalog: Brevo

Schema: SMTP

Label: SMTP Reports

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
days	int64	<input type="checkbox"/>		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date of the report (YYYY-MM-DD). Must be lower than equal to endDate
tag	string	<input type="checkbox"/>		Tag of the emails

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
blocked	int64	Blocked	<input type="checkbox"/>	Number of blocked contact emails for the timeframe
clicks	int64	Clicks	<input type="checkbox"/>	Number of clicks for the timeframe
delivered	int64	Delivered	<input type="checkbox"/>	Number of delivered emails for the timeframe
hardBounces	int64	Hard-bounces	<input type="checkbox"/>	Number of hardbounces for the timeframe
invalid	int64	Invalid	<input type="checkbox"/>	Number of invalid emails for the timeframe
opens	int64	Opens	<input type="checkbox"/>	Number of openings for the timeframe
range	string	Range	<input type="checkbox"/>	Time frame of the report
requests	int64	Requests	<input type="checkbox"/>	Number of requests for the timeframe
softBounces	int64	Soft-bounces	<input type="checkbox"/>	Number of softbounces for the timeframe
spamReports	int64	Spam Reports	<input type="checkbox"/>	Number of complaint (spam report) for the timeframe
uniqueClicks	int64	#Unique Clicks	<input type="checkbox"/>	Number of unique clicks for the timeframe
uniqueOpens	int64	#Unique Opens	<input type="checkbox"/>	Number of unique openings for the timeframe
unsubscribed	int64	Unsubscribed	<input type="checkbox"/>	Number of unsubscribed emails for the timeframe

**Smtptemplatebyid: Sendinblue SMTP Template by ID**

Returns the template information

Catalog: Brevo

Schema: SMTP

Label: SMTP Template by ID

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
templateid	int64	<input checked="" type="checkbox"/>		id of the template

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
doiTemplate	boolean	DOI Template	<input type="checkbox"/>	It is true if template is a valid Double opt-in (DOI) template, otherwise it is false. This field will be available only in case of single template detail call.
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the template
id	int64	ID	<input type="checkbox"/>	ID of the template
isActive	boolean	Is Active	<input type="checkbox"/>	Status of template (true=active, false=inactive)
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)



Name	Data Type	Label	Required	Documentation
name	string	Name	<input type="checkbox"/>	Name of the template
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" for the template
sender_email	string	Sender Email Address	<input type="checkbox"/>	From email for the template
sender_id	string	Sender ID	<input type="checkbox"/>	Sender id of the template
sender_name	string	Sender Name	<input type="checkbox"/>	From email for the template
subject	string	Subject	<input type="checkbox"/>	Subject of the template
tag	string	Tag	<input type="checkbox"/>	Tag of the template
testSent	boolean	Test Sent	<input type="checkbox"/>	Status of test sending for the template (true=test email has been sent, false=test email has not been sent)
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field for the template

### Smtptemplates: Sendinblue SMTP Templates

Get the list of email templates

Catalog: Brevo

Schema: SMTP

Primary Keys: id

Label: SMTP Templates

Can retrieve data and change data using insert, update and delete.

### Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
templateStatus	boolean	<input type="checkbox"/>		Filter on the status of the template. Active = true, inactive = false

## Columns of Table Function

The columns of the table function Smtptemplates 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 and update.

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
doiTemplate	boolean	DOI Template	<input type="checkbox"/>	It is true if template is a valid Double opt-in (DOI) template, otherwise it is false. This field will be available only in case of single template detail call.
htmlContent	string	HTML Content	<input type="checkbox"/>	HTML content of the template
id	int64	ID	<input type="checkbox"/>	ID of the template
isActive	boolean	Is Active	<input type="checkbox"/>	Status of template (true=active, false=inactive)
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the template
replyTo	string	Reply-to	<input type="checkbox"/>	Email defined as the "Reply to" for the template
sender_email	string	Sender Email Address	<input type="checkbox"/>	From email for the template
sender_id	string	Sender ID	<input type="checkbox"/>	Sender id of the template
sender_name	string	Sender Name	<input type="checkbox"/>	From email for the template
subject	string	Subject	<input type="checkbox"/>	Subject of the template
tag	string	Tag	<input type="checkbox"/>	Tag of the template
testSent	boolean	Test Sent	<input type="checkbox"/>	Status of test sending for the template (true=test email has been sent, false=test email has not been sent)
toField	string	To Field	<input type="checkbox"/>	Customisation of the "to" field for the template

### TransacBlockedContacts: Sendinblue Transaction Blocked Contacts

Get the list of blocked or unsubscribed transactional contacts

Catalog: Brevo

Schema: SMTP

Primary Keys: email

Label: Transaction Blocked Contacts

Retrieve: true

Insert: false

Update: false

Delete: true

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) till w hich you w ant to fetch the blocked or unsubscribed contacts
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) from w hich you w ant to fetch the blocked or unsubscribed contacts

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
blockedAt	datetime	Blocked at	<input type="checkbox"/>	Date w hen the contact w as blocked or unsubscribed on
email	string	Email Address	<input type="checkbox"/>	Email address of the blocked or unsubscribed contact
reason_code	string	Reason Code	<input type="checkbox"/>	Reason code for blocking / unsubscribing (This code is safe for comparison)
reason_message	string	Reason Message	<input type="checkbox"/>	Reason for blocking / unsubscribing (This string is not safe for comparison)
senderEmail	string	Sender Email	<input type="checkbox"/>	Sender email address of the blocked or unsubscribed contact

### TransacEmailContent\_Events

Get the personalized content of a sent transactional email

Catalog: Brevo

Schema: SMTP

Label: {res:itgen\_sib\_transacemailcontent\_events\_plural}

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
uuid	string	<input checked="" type="checkbox"/>		Unique id of the transactional email that has been sent to a particular contact

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attachmentCount	int64	Attachment Count	<input type="checkbox"/>	Count of the attachments that were sent in the email
body	string	Body	<input type="checkbox"/>	Actual content of the transactional email that has been sent
date	datetime	Date	<input type="checkbox"/>	Date on which transactional email was sent
email	string	Email Address	<input type="checkbox"/>	Email address to which transactional email has been sent
name	string	Name	<input type="checkbox"/>	Name of the event that occurred on the sent email
subject	string	Subject	<input type="checkbox"/>	Subject of the sent email
templateId	int64	Template ID	<input type="checkbox"/>	Id of the template

Name	Data Type	Label	Required	Documentation
time	datetime	Time	<input type="checkbox"/>	Time at which the event occurred

### TransacEmailContent: Sendinblue Transaction Email Contents

Get the personalized content of a sent transactional email

Catalog: Brevo

Schema: SMTP

Label: Transaction Email Contents

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
uuid	string	<input checked="" type="checkbox"/>		Unique id of the transactional email that has been sent to a particular contact

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attachmentCount	int64	Attachment Count	<input type="checkbox"/>	Count of the attachments that were sent in the email
body	string	Body	<input type="checkbox"/>	Actual content of the transactional email that has been sent
date	datetime	Date	<input type="checkbox"/>	Date on which transactional email was sent

Name	Data Type	Label	Required	Documentation
email	string	Email Address	<input type="checkbox"/>	Email address to which transactional email has been sent
subject	string	Subject	<input type="checkbox"/>	Subject of the sent email
templateId	int64	Template ID	<input type="checkbox"/>	Id of the template

### TransacEmailsList\_TransactionalEmailsTags

Get the list of transactional emails on the basis of allowed filters. This endpoint will show the list of emails for past 30 days by default. To retrieve emails before that time, please pass startDate and endDate in query filters.

Catalog: Brevo

Schema: SMTP

Label: {res:itgen\_sib\_transacemailslist\_transactionalemailstags\_plural}

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
email	string	<input type="checkbox"/>		Mandatory if templateId and messageId are not passed in query filters. Email address to which transactional email has been sent.
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) till which you want to fetch the list. Maximum time period that can be selected is one month.
messageId	string	<input type="checkbox"/>		Mandatory if templateId and email are not passed in query filters. Message ID of the transactional email sent.

Name	Data Type	Required	Default Value	Documentation
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) from which you want to fetch the list. Maximum time period that can be selected is one month.
templateId	int64	<input type="checkbox"/>		Mandatory if email and messageId are not passed in query filters. Id of the template that was used to compose transactional email.

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	<input type="checkbox"/>	Date on which transactional email was sent
email	string	Email Address	<input type="checkbox"/>	Email address to which transactional email has been sent
from	string	From	<input type="checkbox"/>	Email address of the sender from which the email was sent
messageId	string	Message ID	<input type="checkbox"/>	Message Id of the sent email
subject	string	Subject	<input type="checkbox"/>	Subject of the sent email
templateId	int64	Template ID	<input type="checkbox"/>	Id of the template
TEXT	string	Text	<input type="checkbox"/>	
uuid	string	UUID	<input type="checkbox"/>	Unique id of the email sent to a particular contact

### TransacEmailsList: Sendinblue Transaction Email Lists

Get the list of transactional emails on the basis of allowed filters. This endpoint will show the list of emails for past 30 days by default. To retrieve emails before that time, please pass startDate and endDate in query filters.

Catalog: Brevo

Schema: SMTP

Label: Transaction Email Lists

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

## Parameters of Table Function

The following parameters can be used to control the behaviour of the table function `TransacEmailsList`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a

pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
email	string	<input type="checkbox"/>		Mandatory if templateId and messageId are not passed in query filters. Email address to which transactional email has been sent.
endDate	datetime	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) till which you want to fetch the list. Maximum time period that can be selected is one month.
messageId	string	<input type="checkbox"/>		Mandatory if templateId and email are not passed in query filters. Message ID of the transactional email sent.
startDate	datetime	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) from which you want to fetch the list. Maximum time period that can be selected is one month.
templateId	int64	<input type="checkbox"/>		Mandatory if email and messageId are not passed in query filters. Id of the template that was used to compose transactional email.

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	<input type="checkbox"/>	Date on which transactional email was sent
email	string	Email Address	<input type="checkbox"/>	Email address to which transactional email has been sent
from	string	From	<input type="checkbox"/>	Email address of the sender from which the email was sent
messageId	string	Message ID	<input type="checkbox"/>	Message Id of the sent email
subject	string	Subject	<input type="checkbox"/>	Subject of the sent email
templateId	int64	Template ID	<input type="checkbox"/>	Id of the template



Name	Data Type	Label	Required	Documentation
uuid	string	UUID	<input type="checkbox"/>	Unique id of the email sent to a particular contact

### 3.1.9 Schema: TransactionalSMS

#### 3.1.9.1 Tables

##### SmsEvents: Sendinblue SMS Events

Get all your SMS activity (unaggregated events)

Catalog: Brevo

Schema: TransactionalSMS

Label: SMS Events

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
days	int64	<input type="checkbox"/>		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the report
event	string	<input type="checkbox"/>		Filter the report for specific events (Waarden: bounces, hardBounces, softBounces, delivered, sent, accepted, unsubscribed, replies, blocked, rejected)
phoneNumber	string	<input type="checkbox"/>		Filter the report for a specific phone number
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of

Name	Data Type	Required	Default Value	Documentation
				the report
tags	string	<input type="checkbox"/>		Filter the report for specific tags passed as a serialized url-encoded array

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	<input type="checkbox"/>	UTC date-time on which the event has been generated
event	string	Event	<input type="checkbox"/>	Event which occurred
messageId	string	Message ID	<input type="checkbox"/>	Message ID which generated the event
phoneNumber	string	Phone Number	<input type="checkbox"/>	Phone number which has generated the event
reason	string	Reason	<input type="checkbox"/>	Reason of bounce (only available if the event is hardbounce or softbounce)
reply	string	Reply	<input type="checkbox"/>	
tag	string	Tag	<input type="checkbox"/>	Tag of the SMS which generated the event

### TransacSmsReport: Sendinblue SMS Transactions Report

Get your SMS activity aggregated per day

Catalog: Brevo

Schema: TransactionalSMS

Label: SMS Transactions Report

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` 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
days	int64	<input type="checkbox"/>		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the report
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the report
tag	string	<input type="checkbox"/>		Filter on a tag

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
accepted	int64	Accepted	<input type="checkbox"/>	Number of accepted for the date
blocked	int64	Blocked	<input type="checkbox"/>	Number of blocked contact for the date
date	datetime	Date	<input type="checkbox"/>	Date for which statistics are retrieved
delivered	int64	Delivered	<input type="checkbox"/>	Number of delivered SMS for the date
hardBounces	int64	Hard-bounces	<input type="checkbox"/>	Number of hardbounces for the date
rejected	int64	Rejected	<input type="checkbox"/>	Number of rejected for the date
replied	int64	Replied	<input type="checkbox"/>	Number of answered SMS for the date
requests	int64	Requests	<input type="checkbox"/>	Number of requests for the date
softBounces	int64	Soft-bounces	<input type="checkbox"/>	Number of softbounces for the date
unsubscribed	int64	Unsubscribed	<input type="checkbox"/>	Number of unsubscription for the date

### TransactionAggregatedSmsReports: Sendinblue SMS Aggregated Transactions Report

Get your SMS activity aggregated over a period of time

Catalog: Brevo

Schema: TransactionalSMS

Label: SMS Aggregated Transactions Report

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
days	int64	<input type="checkbox"/>		Number of days in the past including today (positive integer). Not compatible with startDate and endDate
endDate	string	<input type="checkbox"/>		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the report
startDate	string	<input type="checkbox"/>		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the report
tag	string	<input type="checkbox"/>		Filter on a tag

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
accepted	int64	Accepted	<input type="checkbox"/>	Number of accepted for the timeframe
blocked	int64	Blocked	<input type="checkbox"/>	Number of blocked contact for the timeframe
delivered	int64	Delivered	<input type="checkbox"/>	Number of delivered SMS for the timeframe
hardBounces	int64	Hard-bounces	<input type="checkbox"/>	Number of hardbounces for the timeframe
range	string	Range	<input type="checkbox"/>	Time frame of the report
rejected	int64	Rejected	<input type="checkbox"/>	Number of rejected for the timeframe

Name	Data Type	Label	Required	Documentation
replied	int64	Replied	<input type="checkbox"/>	Number of answered SMS for the timeframe
requests	int64	Requests	<input type="checkbox"/>	Number of requests for the timeframe
softBounces	int64	Soft-bounces	<input type="checkbox"/>	Number of softbounces for the timeframe
unsubscribed	int64	Unsubscribed	<input type="checkbox"/>	Number of unsubscription for the timeframe

### 3.1.10 Schema: Views

#### 3.1.10.1 Views

##### AbTestCampaignResults: Sendinblue A/B Test Campaign Results

Catalog: Brevo

Schema: Views

Label: A/B Test Campaign Results

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

## View Columns

The columns of the view AbTestCampaignResults are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
CampaignId	int64	ID	<input type="checkbox"/>	ID of the campaign
CampaignName	string	Name	<input type="checkbox"/>	Name of the campaign
CampaignSubject	string	Subject	<input type="checkbox"/>	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
clickRate	string	Clickrate	<input type="checkbox"/>	Click rate for current winning version
openRate	string	Open Rate	<input type="checkbox"/>	Open rate for current winning version
statistics_clicks_Version_A	string	{res:itgen_sib_statistics_clicks_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_clicks_Version_B	string	{res:itgen_sib_statistics_clicks_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_complaints_Version_A	string	{res:itgen_sib_statistics_complaints_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_complaints_Version_B	string	{res:itgen_sib_statistics_complaints_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_hardBounces_Version_A	string	{res:itgen_sib_statistics_hardbounces_version_a}	<input type="checkbox"/>	percentage of an event for version A

Name	Data Type	Label	Required	Documentation
statistics_hardBounces_Version_B	string	{res:itgen_sib_statistics_hardbounces_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_openers_Version_A	string	{res:itgen_sib_statistics_openers_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_openers_Version_B	string	{res:itgen_sib_statistics_openers_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_softBounces_Version_A	string	{res:itgen_sib_statistics_softbounces_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_softBounces_Version_B	string	{res:itgen_sib_statistics_softbounces_version_b}	<input type="checkbox"/>	percentage of an event for version B
statistics_unsubscribed_Version_A	string	{res:itgen_sib_statistics_unsubscribed_version_a}	<input type="checkbox"/>	percentage of an event for version A
statistics_unsubscribed_Version_B	string	{res:itgen_sib_statistics_unsubscribed_version_b}	<input type="checkbox"/>	percentage of an event for version B
winningCriteria	string	Winning Criteria	<input type="checkbox"/>	Criteria chosen for winning version (Open/Click)
winningSubjectLine	string	Winning Subject Line	<input type="checkbox"/>	Subject Line of current winning version
winningVersion	string	Winning Version	<input type="checkbox"/>	Winning Campaign Info. pending = Campaign has been picked for sending and winning version is yet to be decided, tie = A tie happened between both the versions, notAvailable = Campaign has not yet been picked for sending.
winningVersionRate	string	Winning Version Rate	<input type="checkbox"/>	Open/Click rate for the winner version

### FolderLists: Sendinblue Folder Lists

Catalog: Brevo

Schema: Views

Label: Folder Lists

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

## View Columns

The columns of the view FolderLists are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
Folderid	int64	ID	<input type="checkbox"/>	ID of the folder

Name	Data Type	Label	Required	Documentation
Foldername	string	Name	<input type="checkbox"/>	Name of the folder
FoldertotalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the folder
FoldertotalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the folder
FolderuniqueSubscribers	int64	#Unique Subscribers	<input type="checkbox"/>	Number of unique contacts in the folder
id	int64	ID	<input type="checkbox"/>	ID of the list
name	string	Name	<input type="checkbox"/>	Name of the list
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the list

### ListCampaignStatistics: Sendinblue List Campaign Statistics

Catalog: Brevo

Schema: Views

Label: List Campaign Statistics

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

## View Columns

The columns of the view ListCampaignStatistics are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	<input type="checkbox"/>	ID of the campaign
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	boolean	Dynamic List	<input type="checkbox"/>	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)
folderId	int64	Folder ID	<input type="checkbox"/>	ID of the folder
id	int64	ID	<input type="checkbox"/>	ID of the list
name	string	Name	<input type="checkbox"/>	Name of the list
stats_clickers	int64	#Clickers	<input type="checkbox"/>	Number of total clicks for the campaign
stats_complaints	int64	#Complaints	<input type="checkbox"/>	Number of complaints (Spam reports) for the campaign
stats_deferred	int64	#Deferred	<input type="checkbox"/>	Number of deferred emails for the campaign
stats_delivered	int64	#Delivered	<input type="checkbox"/>	Number of delivered emails for the campaign
stats_hardBounces	int64	#Hard-bounces	<input type="checkbox"/>	Number of harbounce for the campaign

Name	Data Type	Label	Required	Documentation
stats_listId	int64	List ID	<input type="checkbox"/>	List Id of email campaign (only in case of get email campaign(s) (not for global stats))
stats_returnBounce	int64	#Return-bounces	<input type="checkbox"/>	Total number of non-delivered campaigns for a particular campaign id.
stats_sent	int64	#Sent	<input type="checkbox"/>	Number of sent emails for the campaign
stats_softBounces	int64	#Soft-bounces	<input type="checkbox"/>	Number of softbounce for the campaign
stats_trackableView s	int64	#Trackable View s	<input type="checkbox"/>	Recipients without any privacy protection option enabled in their email client
stats_uniqueClicks	int64	#Unique Clicks	<input type="checkbox"/>	Number of unique clicks for the campaign
stats_uniqueView s	int64	#Unique View s	<input type="checkbox"/>	Number of unique openings for the campaign
stats_unsubscriptions	int64	#Unsubscriptions	<input type="checkbox"/>	Number of unsubscription for the campaign
stats_view ed	int64	#View ed	<input type="checkbox"/>	Number of openings for the campaign
totalBlacklisted	int64	Total Blacklisted	<input type="checkbox"/>	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	<input type="checkbox"/>	Number of contacts in the list

### 3.1.11 Schema: Webhooks

#### 3.1.11.1 Tables

##### WebhookById: Sendinblue Webhook by ID

Get a webhook details

Catalog: Brevo

Schema: Webhooks

Label: Webhook by ID

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

## Parameters of Table Function

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

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



Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
w ebhookId	int64	<input checked="" type="checkbox"/>		Id of the w ebhook

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the w ebhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	string	Description	<input type="checkbox"/>	Description of the w ebhook
id	int64	ID	<input type="checkbox"/>	ID of the w ebhook
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the w ebhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
type	string	Type	<input type="checkbox"/>	Type of w ebhook (marketing or transac)
url	string	URL	<input type="checkbox"/>	URL of the w ebhook

### WebhookEventsByWebhookId

Get a webhook details

Catalog: Brevo

Schema: Webhooks

Label: {res:itgen\_sib\_webhookeventsbywebhookid\_plural}

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
w ebhookld	int64	<input checked="" type="checkbox"/>		Id of the w ebhook

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the w ebhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	string	Description	<input type="checkbox"/>	Description of the w ebhook
id	int64	ID	<input type="checkbox"/>	ID of the w ebhook
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the w ebhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
TEXT	string	Text	<input type="checkbox"/>	Events w hich w ill trigger the w ebhook w hen they occure
type	string	Type	<input type="checkbox"/>	Type of w ebhook (marketing or transac)
url	string	URL	<input type="checkbox"/>	URL of the w ebhook

### Webhooks: Sendinblue Webhooks

Get all webhooks

Catalog: Brevo

Schema: Webhooks

Primary Keys: id

Label: Webhooks

Can retrieve data and change data using insert, update and delete.

## Parameters of Table Function

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

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four para-

meters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
type	string	<input type="checkbox"/>	transactional	Filter on webhook type (Waarden: marketing, transactional)

## Columns of Table Function

The columns of the table function Webhooks 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 and update.

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	string	Description	<input type="checkbox"/>	Description of the webhook
id	int64	ID	<input type="checkbox"/>	ID of the webhook
modifiedAt	datetime	Modified Date	<input type="checkbox"/>	Last modification UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
type	string	Type	<input type="checkbox"/>	Type of webhook (marketing or transac)
url	string	URL	<input type="checkbox"/>	URL of the webhook

### 3.1.12 Schema: WhatsAppCampaigns

#### 3.1.12.1 Tables

##### WhatsAppCampaignById: Sendinblue WhatsApp Campaign by ID

Get a WhatsApp campaign

Catalog: Brevo

Schema: WhatsAppCampaigns

Label: WhatsApp Campaign by ID

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

## Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WhatsAppCampaignById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a

pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
campaignId	int64	<input checked="" type="checkbox"/>		Id of the campaign

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignName	string	{res:itgen_sib_campaignname}	<input type="checkbox"/>	Name of the WhatsApp Campaign
campaignStatus	string	{res:itgen_sib_campaignstatus}	<input type="checkbox"/>	Status of the WhatsApp Campaign
createdAt	string	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	<input type="checkbox"/>	ID of the WhatsApp Campaign
modifiedAt	string	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
scheduledAt	string	Schedule Date	<input type="checkbox"/>	UTC date-time on which WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
senderNumber	string	{res:itgen_sib_sendernumber}	<input type="checkbox"/>	Sender of the WhatsApp Campaign
stats_delivered	int64	#Delivered	<input type="checkbox"/>	
stats_notSent	int64	{res:itgen_sib_stats_notsent}	<input type="checkbox"/>	
stats_read	int64	{res:itgen_sib_stats_read}	<input type="checkbox"/>	
stats_sent	int64	#Sent	<input type="checkbox"/>	
stats_unsubscribe	int64	{res:itgen_sib_stats_unsubscribe}	<input type="checkbox"/>	
template_button_type	string	{res:itgen_sib_template_button_type}	<input type="checkbox"/>	
template_category	string	{res:itgen_sib_template_category}	<input type="checkbox"/>	description of the template

Name	Data Type	Label	Required	Documentation
template_contains_button	boolean	{res:itgen_sib_template_contains_button}	<input type="checkbox"/>	
template_display_header	boolean	{res:itgen_sib_template_display_header}	<input type="checkbox"/>	
template_header_type	string	{res:itgen_sib_template_header_type}	<input type="checkbox"/>	type of header
template_hide_footer	boolean	{res:itgen_sib_template_hide_footer}	<input type="checkbox"/>	
template_language	string	{res:itgen_sib_template_language}	<input type="checkbox"/>	language of the template
template_name	string	{res:itgen_sib_template_name}	<input type="checkbox"/>	name of the template

### WhatsAppCampaigns: Sendinblue WhatsApp Campaigns

Return all your created WhatsApp campaigns

Catalog: Brevo

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp Campaigns

Retrieve: true

Insert: false

Update: false

Delete: true

### Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	string	<input type="checkbox"/>		**Mandatory if startDate is used**. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the

Name	Data Type	Required	Default Value	Documentation
				campaigns created.**Prefer to pass your timezone in date-time format for accurate result**
startDate	string	<input type="checkbox"/>		**Mandatory if endDate is used**. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the campaigns created.**Prefer to pass your timezone in date-time format for accurate result**

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignName	string	{res:itgen_sib_campaignname}	<input type="checkbox"/>	Name of the WhatsApp Campaign
campaignStatus	string	{res:itgen_sib_campaignstatus}	<input type="checkbox"/>	Status of the WhatsApp Campaign
createdAt	string	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
errorReason	string	{res:itgen_sib_errorreason}	<input type="checkbox"/>	Error reason in the campaign creation
id	int64	ID	<input type="checkbox"/>	ID of the WhatsApp Campaign
invalidatedContacts	int64	{res:itgen_sib_invalidatedcontacts}	<input type="checkbox"/>	Count of invalidated contacts
modifiedAt	string	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the WhatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
readPercentage	float	{res:itgen_sib_readpercentage}	<input type="checkbox"/>	Read percentage of the the WhatsApp campaign created
scheduledAt	string	Schedule Date	<input type="checkbox"/>	UTC date-time on which WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
stats_delivered	int64	#Delivered	<input type="checkbox"/>	
stats_notSent	int64	{res:itgen_sib_stats_notsent}	<input type="checkbox"/>	
stats_read	int64	{res:itgen_sib_stats_read}	<input type="checkbox"/>	
stats_sent	int64	#Sent	<input type="checkbox"/>	
stats_unsubscribe	int64	{res:itgen_sib_stats_unsubscribe}	<input type="checkbox"/>	
templateId	string	Template ID	<input type="checkbox"/>	Id of the WhatsApp template

## WhatsAppTemplates: Sendinblue WhatsApp Templates

Return all your created WhatsApp templates

Catalog: Brevo

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp Templates

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

## Parameters of Table Function

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

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

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` 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
endDate	string	<input type="checkbox"/>		<b>**Mandatory if startDate is used**</b> . Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the templates created. <b>**Prefer to pass your timezone in date-time format for accurate result**</b>
startDate	string	<input type="checkbox"/>		<b>**Mandatory if endDate is used**</b> . Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the templates created. <b>**Prefer to pass your timezone in date-time format for accurate result**</b>

## Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
category	string	Category	<input type="checkbox"/>	category of the template

Name	Data Type	Label	Required	Documentation
createdAt	string	Creation Date	<input type="checkbox"/>	Creation UTC date-time of the WhatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
errorReason	string	{res:itgen_sib_error-reason}	<input type="checkbox"/>	Error reason in the template creation
id	string	ID	<input type="checkbox"/>	id of the template
language	string	{res:itgen_sib_language}	<input type="checkbox"/>	Language in which template exists
modifiedAt	string	Modified Date	<input type="checkbox"/>	UTC date-time of last modification of the WhatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	<input type="checkbox"/>	Name of the WhatsApp template
status	string	Status	<input type="checkbox"/>	Status of the WhatsApp template



# Index

## - # -

#Answered 81, 83, 85  
 #Clickers 52, 107  
 #Complaints 52, 107  
 #Deferred 52, 107  
 #Delivered 52, 81, 83, 85, 107, 111, 113  
 #Hard-bounces 52, 81, 83, 85, 107  
 #Mirror Clicks 60, 63, 66  
 #Processing 81, 83, 85  
 #Remaining 60, 63, 66  
 #Return-bounces 52, 107  
 #Sent 52, 81, 83, 85, 107, 111, 113  
 #Soft-bounces 52, 81, 83, 85, 107  
 #Trackable Views 52, 107  
 #Unique Clicks 52, 89, 90, 107  
 #Unique Opens 89, 90  
 #Unique Subscribers 49, 50, 106  
 #Unique Views 52, 107  
 #Unsubscriptions 52, 81, 83, 85, 107  
 #Viewed 52, 107

## - A -

A/B Test Campaign Result by ID 55  
 A/B Test Campaign Results 105  
 A/B-Testing 57, 60, 63, 66, 69  
 ABONNEMENTEN\_ACTIEF 36, 41, 44  
 ABONNEMENTEN\_ACTIEF\_TOT 36, 41, 44  
 AbTestCampaignResultById 55  
 AbTestCampaignResults 105  
 abTesting 57, 60, 63, 66, 69  
 Accepted 102, 103  
 Active 77, 78, 79  
 add-odata-mandatory-filters 2  
 Address City 17  
 Address Country 17  
 Address Street 17  
 Address Zip Code 17  
 address\_city 17  
 address\_country 17  
 address\_street 17  
 address\_zipCode 17  
 analysis-enforce-row-uniqueness 2  
 api-access-token 2  
 api-client-id 2

api-client-secret 2  
 api-pre-expiry-refresh-sec 2  
 api-redirect-url 2  
 api-refresh-token 2  
 api-scope 2  
 api-token-url 2  
 api-url 2  
 Attachment Count 96, 97  
 attachmentCount 96, 97  
 Attribute Values 19  
 Attributes 19, 23, 24, 25, 26, 28, 29, 30, 31, 33, 34  
 AttributeValues 19  
 automation-key 2

## - B -

Background Process by ID 75  
 Background Processes 76  
 BackgroundProcessById 75  
 BackgroundProcesses 76  
 BLACKLIST 36, 41, 44  
 BLOB Preferred 74  
 BLOB\_PREFERRED 74  
 Blocked 89, 90, 102, 103  
 Blocked at 94  
 Blocked Domains 86  
 blockedAt 94  
 BlockedDomains 86  
 Body 96, 97  
 BOL\_RESPONSE\_CACHE\_MAX\_AGE\_SEC 74  
 BUBS\_GBR\_ID 36, 41, 44  
 BUBS\_GBR\_ID\_MASTER 36, 41, 44  
 bulk-delete-page-size-rows 2  
 bulk-insert-page-size-bytes 2  
 bulk-insert-page-size-rows 2

## - C -

Calculated Value 19  
 calculatedValue 19  
 Campaign ID 21, 22, 24, 25, 26, 28, 29, 30, 31, 35, 38, 39, 40, 46, 48, 52, 107  
 campaignId 21, 22, 24, 25, 26, 28, 29, 30, 31, 35, 38, 39, 40, 46, 48, 52, 55, 57, 73, 80, 105, 107, 111  
 campaignName 105, 111, 113  
 campaignStatus 111, 113  
 CampaignSubject 105  
 Category 19, 115  
 categoryName 55  
 CLICKERS 36, 41, 44  
 Clickrate 55, 105

Clicks	89, 90	Content Type	74
Company Name	17, 36, 41, 44	CONTENT_TYPE	74
COMPANY_NAME	36, 41, 44	Contract Code	36, 41, 44
companyName	17	CONTRACT_CODE	36, 41, 44
Contact Complaint Statistics by Email	22	Count	30, 38, 40
Contact Detail Complaint Statistics by Email	25	COUNTRY	36, 41, 44
Contact Detail Hard Bounce Statistics by Email	26	createdAt	23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44, 51, 52, 57, 60, 63, 66, 69, 80, 81, 83, 85, 92, 93, 107, 108, 109, 110, 111, 113, 115
Contact Detail Link Click Statistics by Email	28	createdSince	41
Contact Detail Message Sent Statistics by Email	29	Creation Date	23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44, 51, 52, 57, 60, 63, 66, 69, 80, 81, 83, 85, 92, 93, 107, 108, 109, 110, 111, 113, 115
Contact Detail Open Statistics by Email	30	Credit Type	17
Contact Detail Soft Bounce Statistics by Email	31	Credits	17
Contact Detail Transaction Attributes by Email	33	creditsType	17
Contact Detail Unsubscriptions by Email (Administrative Mode)	23		
Contact Detail User Unsubscriptions by Email	34		
Contact Hard Bounce Statistics by Email	35		
Contact Information by Email	36		
Contact Link Clicks by Email	38		
Contact Message Sent Statistics by Email	39		
Contact Open Statistics by Email	40		
Contact Soft Bounce Statistics by Email	46		
Contact Transaction Attributes by Email	47		
Contact Unsubscriptions by Email (Administrative Mode)	20		
Contact Unsubscriptions by Email (User Mode)	48		
CONTACT_TIMEZONE	36, 41, 44		
ContactAdminUnsubscriptionsByEmail	20		
ContactClickStatisticsByEmail	21		
ContactComplaintStatisticsByEmail	22		
ContactDetailAdminUnsubscriptionsByEmail	23		
ContactDetailClickStatisticsByEmail	24		
ContactDetailComplaintStatisticsByEmail	25		
ContactDetailHardBounceStatisticsByEmail	26		
ContactDetailLinkClickStatisticsByEmail	28		
ContactDetailMessageSentStatisticsByEmail	29		
ContactDetailOpenStatisticsByEmail	30		
ContactDetailSoftBounceStatisticsByEmail	31		
ContactDetailTransactionAttributesByEmail	33		
ContactDetailUserUnsubscriptionsByEmail	34		
ContactHardBounceStatisticsByEmail	35		
ContactInfoByEmail	36		
ContactLinkClicksByEmail	38		
ContactMessageSentStatisticsByEmail	39		
ContactOpenStatisticsByEmail	40		
Contacts	41		
Contacts by List ID	44		
ContactsByListId	44		
ContactSoftBounceStatisticsByEmail	46		
ContactTransactionAttributesByEmail	47		
ContactUserUnsubscriptionsByEmail	48		
Content	80, 81, 83, 85		

## - D -

D31APPLICATIONS	36, 41, 44
D31BILLINGIDS	36, 41, 44
D31BUBSGBRIDS	36, 41, 44
D31DATACONTAINERCNT	36, 41, 44
D31MBIO	36, 41, 44
D31PARTITIONCNT	36, 41, 44
D31PROVIDERCNT	36, 41, 44
D31ROWCNT	36, 41, 44
D31TABLECNT	36, 41, 44
Database Driver	1
Date	87, 89, 96, 97, 98, 99, 101, 102
DATE_ENDED	74
DATE_STARTED	74
days	87, 89, 90, 101, 102, 103
Delivered	89, 90, 102, 103
DOI Template	92, 93
doiTemplate	92, 93
Domain	77, 78
download-error-400-bad-request-max-tries	2
download-error-400-bad-request-sleep-initial-ms	2
download-error-400-bad-request-sleep-max-ms	2
download-error-400-bad-request-sleep-multiplier	2
download-error-408-request-timeout-max-tries	2
download-error-408-request-timeout-sleep-initial-ms	2
download-error-408-request-timeout-sleep-max-ms	2
download-error-408-request-timeout-sleep-multiplier	2
download-error-422-bad-request-max-tries	2
download-error-422-bad-request-sleep-initial-ms	2
download-error-422-bad-request-sleep-max-ms	2

download-error-422-bad-request-sleep-multiplicator 2	download-error-argument-exception-sleep-initial-ms 2
download-error-429-too-many-requests-max-tries 2	download-error-argument-exception-sleep-max-ms 2
download-error-429-too-many-requests-sleep-initial-ms 2	download-error-argument-exception-sleep-multiplicator 2
download-error-429-too-many-requests-sleep-max-ms 2	download-error-internet-down-max-tries 2
download-error-429-too-many-requests-sleep-multiplicator 2	download-error-internet-down-sleep-initial-ms 2
download-error-500-internal-server-error-max-tries 2	download-error-internet-down-sleep-max-ms 2
download-error-500-internal-server-error-sleep-initial-ms 2	download-error-internet-down-sleep-multiplicator 2
download-error-500-internal-server-error-sleep-max-ms 2	download-error-io-exception-max-tries 2
download-error-500-internal-server-error-sleep-multiplicator 2	download-error-io-exception-sleep-initial-ms 2
download-error-502-server-unavailable-max-tries 2	download-error-io-exception-sleep-max-ms 2
download-error-502-server-unavailable-sleep-initial-ms 2	download-error-io-exception-sleep-multiplicator 2
download-error-502-server-unavailable-sleep-max-ms 2	download-error-json-exception-max-tries 2
download-error-502-server-unavailable-sleep-multiplicator 2	download-error-json-exception-sleep-initial-ms 2
download-error-503-server-unavailable-max-tries 2	download-error-json-exception-sleep-max-ms 2
download-error-503-server-unavailable-sleep-initial-ms 2	download-error-json-exception-sleep-multiplicator 2
download-error-503-server-unavailable-sleep-max-ms 2	download-error-name-resolution-failure-max-tries 2
download-error-503-server-unavailable-sleep-multiplicator 2	download-error-name-resolution-failure-sleep-initial-ms 2
download-error-504-gateway-timeout-max-tries 2	download-error-name-resolution-failure-sleep-max-ms 2
download-error-504-gateway-timeout-sleep-initial-ms 2	download-error-name-resolution-failure-sleep-multiplicator 2
download-error-504-gateway-timeout-sleep-max-ms 2	download-error-name-resolution-failure-sleep-multiplicator 2
download-error-504-gateway-timeout-sleep-multiplicator 2	download-error-socket-exception-max-tries 2
download-error-590-network-connect-timeout-max-tries 2	download-error-socket-exception-sleep-initial-ms 2
download-error-590-network-connect-timeout-sleep-initial-ms 2	download-error-socket-exception-sleep-max-ms 2
download-error-590-network-connect-timeout-sleep-max-ms 2	download-error-socket-exception-sleep-multiplicator 2
download-error-590-network-connect-timeout-sleep-multiplicator 2	download-error-web-exception-max-tries 2
download-error-599-network-connect-timeout-max-tries 2	download-error-web-exception-sleep-initial-ms 2
download-error-599-network-connect-timeout-sleep-initial-ms 2	download-error-web-exception-sleep-max-ms 2
download-error-599-network-connect-timeout-sleep-max-ms 2	download-error-web-exception-sleep-multiplicator 2
download-error-599-network-connect-timeout-sleep-multiplicator 2	download-error-web-not-found-max-tries 2
	download-error-web-not-found-sleep-initial-ms 2
	download-error-web-not-found-sleep-max-ms 2
	download-error-web-not-found-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

DRY\_RUN 74

duplicate-row-mode 2

Duration (ms) 74

DURATION\_MS 74

Dynamic List 51, 52, 107

dynamicList 51, 52, 107

## - E -

email 17, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 35, 36, 38, 39, 40, 41, 44, 46, 47, 48, 78, 79, 87, 94, 96, 97, 98, 99

Email Address 17, 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44, 78, 79, 87, 94, 96, 97, 98, 99

Email Campaign by ID 57

Email Campaign Recipient Exclusion Lists 60

Email Campaign Recipient Lists 63

Email Campaign Statistics 69

Email Campaigns 66

Email Event Report 87

Email is Blacklisted 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44

emailBlacklisted 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44

EmailCampaignByld 57

EmailCampaignRecipientExclusionLists 60

EmailCampaignRecipientLists 63

EmailCampaigns 66

EmailCampaignStatistics 69

EmailEventReport 87

End Date 17, 74

endDate 17, 20, 21, 22, 35, 38, 39, 40, 46, 47, 48, 60, 63, 66, 69, 81, 83, 85, 87, 89, 90, 94, 98, 99, 101, 102, 103, 113, 115

endDate 17, 20, 21, 22, 35, 38, 39, 40, 46, 47, 48, 60, 63, 66, 69, 81, 83, 85, 87, 89, 90, 94, 98, 99, 101, 102, 103, 113, 115

Error Message Code 74

Error Message Text 74

ERROR\_MESSAGE\_CODE 74

ERROR\_MESSAGE\_TEXT 74

errorReason 113, 115

Estate Person ID 36, 41, 44

Event 87, 101

Event Time 20, 22, 25, 26, 29, 30, 31, 35, 38, 39, 40, 46, 48

eventTime 20, 22, 25, 26, 29, 30, 31, 35, 38, 39, 40, 46, 48

excludeHtmlContent 60, 63, 66, 69

exclusionLists 63, 83

Export URL 75, 76

export\_url 75, 76

EXT\_ID 36, 41, 44

## - F -

Fail on Error 74

FAIL\_ON\_ERROR 74

First Name 17, 36, 41, 44

FIRSTNAME 17, 36, 41, 44

Folder by ID 49

Folder ID 51, 52, 53, 107

Folder Lists 106

FolderByld 49

folderId 49, 51, 52, 53, 54, 106, 107

FolderLists 106

Foldername 106

Folders 50

FoldertotalBlacklisted 106

FoldertotalSubscribers 106

FolderuniqueSubscribers 106

Footer 57, 60, 63, 66, 69

force-case-sensitive-identifiers 2

forced-casing-identifiers 2

Forums Login 36, 41, 44

FORUMS\_LOGIN 36, 41, 44

From 87, 98, 99

## - G -

Global Statistics 60, 63, 66, 69

globalStats 69

## - H -

HardBounces 89, 90, 102, 103

Hard-bounces 89, 90, 102, 103

Header 57, 60, 63, 66, 69

HTML Content 57, 60, 63, 66, 69, 92, 93

htmlContent 57, 60, 63, 66, 69, 92, 93

HTTP Disk Cache Maximum Age (sec) 74

HTTP Memory Cache Maximum Age (sec) 74

HTTP Method 74

HTTP Status Code 74

HTTP\_DISK\_CACHE\_MAX\_AGE\_SEC 74

HTTP\_DISK\_CACHE\_SAVE 74

HTTP\_DISK\_CACHE\_USE 74

HTTP\_MEMORY\_CACHE\_MAX\_AGE\_SEC 74

HTTP\_MEMORY\_CACHE\_SAVE 74

HTTP\_MEMORY\_CACHE\_USE 74

HTTP\_METHOD 74  
 HTTP\_STATUS\_CODE 74  
 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-max-ms 2  
 http-get-timeout-ms 2  
 http-memory-cache-compression-level 2  
 http-memory-cache-max-age-sec 2  
 http-post-timeout-max-ms 2  
 http-post-timeout-ms 2

## - I -

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-http-503-errors 2  
 ignore-unknown-path-type 2  
 ignore-values-unknown-path 2  
 Inline Image Activation 57, 60, 63, 66, 69  
 inlinelImageActivation 57, 60, 63, 66, 69  
 Invalid 89, 90  
 invalidatedContacts 113  
 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-compress-sparse-arrays 2  
 invantive-sql-correct-invalid-date 2  
 invantive-sql-execution-profile-disk-path 2  
 invantive-sql-execution-profile-to-disk 2  
 invantive-sql-forward-filters-to-data-containers 2  
 invantive-sql-share-byte-arrays 2  
 invantive-sql-share-strings 2  
 invantive-sql-shuffle-fetch-results-data-containers 2  
 invantive-use-cache 2  
 IP 20, 30, 38, 40, 48, 77, 78, 79, 87  
 IP Addresses 77  
 IP Addresses by Send ID 77

IpAddresses 77  
 IpAddressesBySenderId 77  
 Is Active 92, 93  
 isActive 92, 93

## - J -

Job Title 36, 41, 44  
 JOBTITLE 36, 41, 44  
 join-set-points-per-request 2

## - L -

Label 19  
 LANDLINE\_NUMBER 36, 41, 44  
 language 115  
 Language Code 36, 41, 44  
 LANGUAGE\_CODE 36, 41, 44  
 Last Name 17, 36, 41, 44  
 LASTNAME 17, 36, 41, 44  
 limit-partition-calls-left 2  
 Link 87  
 Link Statistics 60, 63, 66, 69  
 linksStats 69  
 List by ID 51  
 List Campaign Statistics 107  
 List Campaign Statistics by List ID 52  
 List ID 52, 107  
 List IDs 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44  
 ListById 51  
 ListCampaignStatistics 107  
 ListCampaignStatisticsByListId 52  
 listId 44, 51, 52  
 listIds 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44  
 lists 53, 60, 81  
 Lists by Folder ID 54  
 ListsByFolderId 54  
 listUnsubscribed 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44  
 log-native-calls-to-disk-max-events 2  
 log-native-calls-to-disk-max-seconds 2  
 log-native-calls-to-disk-on-error 2  
 log-native-calls-to-disk-on-success 2  
 log-native-calls-to-trace 2

## - M -

Marketing Automation Enabled 17

Marketing Automation Key 17  
 marketingAutomation\_enabled 17  
 marketingAutomation\_key 17  
 maximum-discovered-column-count 2  
 maximum-length-identifiers 2  
 maximum-url-length 2  
 max-odata-filters 2  
 max-odata-rewrite-in-count 2  
 max-url-length-accepted 2  
 max-url-length-desired 2  
 Message ID 87, 98, 99, 101  
 messageld 87, 98, 99, 101  
 metadata-cache-max-age-sec 2  
 Mirror Active 57, 60, 63, 66, 69  
 Mirror Click 69  
 mirrorActive 57, 60, 63, 66, 69  
 mirrorClick 69  
 Mobile Number 36, 41, 44  
 MOBILE\_NUMBER 36, 41, 44  
 Modified Date 23, 24, 25, 26, 28, 29, 30, 31, 33, 34,  
 36, 41, 44, 57, 60, 63, 66, 69, 80, 81, 83, 85, 92, 93,  
 108, 109, 110, 111, 113, 115  
 modifiedAt 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, Range 90, 103  
 36, 41, 44, 57, 60, 63, 66, 69, 80, 81, 83, 85, 92, 93  
 108, 109, 110, 111, 113, 115  
 modifiedSince 41, 44

## - N -

Name 19, 49, 50, 51, 52, 53, 54, 57, 60, 63, 66, 69,  
 75, 76, 78, 79, 80, 81, 83, 85, 92, 93, 96, 105, 106,  
 107, 115  
 Native Platform Scalar Requests 74  
 NATIVEPLATFORMSCALARREQUESTS 74  
 npt 74

## - O -

oauth-unauthorized-max-tries 2  
 oauth-unauthorized-sleep-initial-ms 2  
 oauth-unauthorized-sleep-max-ms 2  
 oauth-unauthorized-sleep-multiplier 2  
 Open Rate 55, 105  
 openRate 55, 105  
 Opens 89, 90  
 Order Date 47  
 Order ID 47  
 Order Price 47  
 orderDate 47  
 orderId 47  
 orderPrice 47

ORIG\_SYSTEM\_GROUP 74  
 ORIG\_SYSTEM\_REFERENCE 74  
 Original System Group 74  
 Original System Reference 74

## - P -

partition-slot-based-rate-limit-length-ms 2  
 partition-slot-based-rate-limit-slots 2  
 Payload 74  
 PAYLOAD\_TEXT 74  
 Phone Number 36, 41, 44, 101  
 PHONE\_NUMBER 36, 41, 44  
 phoneNumber 101  
 pre-request-delay-ms 2  
 Primary Audience 36, 41, 44  
 PRIMARY\_AUDIENCE 36, 41, 44  
 processId 75

## - R -

Range 90, 103  
 READERS 36, 41, 44  
 readPercentage 113  
 Reason 87, 101  
 Reason Code 94  
 Reason Message 94  
 reason\_code 94  
 reason\_message 94  
 Recipients 57, 80  
 recipients\_exclusionLists 66, 69, 85  
 recipients\_lists 66, 69, 85  
 Recurring 57, 60, 63, 66, 69  
 Rejected 102, 103  
 Relay Data Port 17  
 Relay Data Relay 17  
 Relay Data Username 17  
 Relay Enabled 17  
 relay\_data\_port 17  
 relay\_data\_relay 17  
 relay\_data\_userName 17  
 relay\_enabled 17  
 Remaining 69  
 Replied 102, 103  
 Reply 101  
 replyTo 57, 60, 63, 66, 69, 92, 93  
 Reply-to 57, 60, 63, 66, 69, 92, 93  
 requested-page-size 2  
 Requests 89, 90, 102, 103  
 requests-parallel-max 2

Response Cache Maximum Age (sec)	74	shareLink	57, 60, 63, 66, 69
Result BLOB	74	simulate-http-400-errors	2
Result Date Time	74	simulate-http-400-errors-percentage	2
Result Number	74	simulate-http-401-errors	2
Result Text	74	simulate-http-401-errors-percentage	2
RESULT_BLOB	74	simulate-http-403-errors	2
RESULT_DATE_TIME_UTC	74	simulate-http-403-errors-percentage	2
RESULT_NUMBER	74	simulate-http-408-errors	2
RESULT_TEXT	74	simulate-http-408-errors-percentage	2
Return Bounce	57, 60, 63, 66, 69	simulate-http-429-errors	2
returnBounce	57, 60, 63, 66, 69	simulate-http-429-errors-percentage	2
Run without Actions	74	simulate-http-500-errors	2
		simulate-http-500-errors-percentage	2
		simulate-http-502-errors	2
		simulate-http-502-errors-percentage	2
		simulate-http-503-errors	2
		simulate-http-503-errors-percentage	2
		simulate-http-504-errors	2
		simulate-http-504-errors-percentage	2
		simulate-http-522-errors	2
		simulate-http-522-errors-percentage	2
		simulate-http-524-errors	2
		simulate-http-524-errors-percentage	2
		simulate-http-protocol-errors	2
		simulate-http-protocol-errors-percentage	2
		simulate-http-timeout-errors	2
		simulate-http-timeout-errors-percentage	2
		slot-based-rate-limit-length-ms	2
		slot-based-rate-limit-slots	2
		SMS	36, 41, 44
		SMS Aggregated Transactions Report	103
		SMS Campaign by ID	80
		SMS Campaign Recipient Exclusion Lists	81
		SMS Campaign Recipient Lists	83
		SMS Campaigns	85
		SMS Events	101
		SMS is Blacklisted	23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44
		SMS Transactions Report	102
		smsBlacklisted	23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44
		SmsCampaignById	80
		SmsCampaignRecipientExclusionLists	81
		SmsCampaignRecipientLists	83
		SmsCampaigns	85
		SmsEvents	101
		SMTP Report	89
		SMTP Reports	90
		SMTP Template by ID	92
		SMTP Templates	93
		SmtReport	89

SmtReports 90  
 SmtTemplateById 92  
 SmtTemplates 93  
 softBounces 89, 90, 102, 103  
 Soft-bounces 89, 90, 102, 103  
 Spam Reports 89, 90  
 spamReports 89, 90  
 Split Rule 57, 60, 63, 66, 69  
 splitRule 57, 60, 63, 66, 69  
 standardize-identifiers 2  
 standardize-identifiers-casing 2  
 Start Date 17, 74  
 startDate 17, 20, 21, 22, 35, 38, 39, 40, 46, 47, 48, 15  
 60, 63, 66, 69, 81, 83, 85, 87, 89, 90, 94, 98, 99, 101, 102, 103, 113, 115  
 Statistics 57, 60, 63, 66, 69, 80  
 Statistics by Browser 60, 63, 66, 69  
 Statistics by Domain 60, 63, 66, 69  
 statistics\_answered 81, 83, 85  
 statistics\_clicks\_Version\_A 55, 105  
 statistics\_clicks\_Version\_B 55, 105  
 statistics\_complaints\_Version\_A 55, 105  
 statistics\_complaints\_Version\_B 55, 105  
 statistics\_delivered 81, 83, 85  
 statistics\_globalStats 60, 63, 66  
 statistics\_hardBounces 81, 83, 85  
 statistics\_hardBounces\_Version\_A 55, 105  
 statistics\_hardBounces\_Version\_B 55, 105  
 statistics\_linksStats 60, 63, 66  
 statistics\_mirrorClick 60, 63, 66  
 statistics\_openers\_Version\_A 55, 105  
 statistics\_openers\_Version\_B 55, 105  
 statistics\_processing 81, 83, 85  
 statistics\_remaining 60, 63, 66  
 statistics\_sent 81, 83, 85  
 statistics\_softBounces 81, 83, 85  
 statistics\_softBounces\_Version\_A 55, 105  
 statistics\_softBounces\_Version\_B 55, 105  
 statistics\_statsByBrowser 60, 63, 66  
 statistics\_statsByDomain 60, 63, 66  
 statistics\_unsubscribed\_Version\_A 55, 105  
 statistics\_unsubscribed\_Version\_B 55, 105  
 statistics\_unsubscriptions 81, 83, 85  
 stats\_clickers 52, 107  
 stats\_complaints 52, 107  
 stats\_deferred 52, 107  
 stats\_delivered 52, 107, 111, 113  
 stats\_hardBounces 52, 107  
 stats\_listId 52, 107  
 stats\_notSent 111, 113  
 stats\_read 111, 113  
 stats\_returnBounce 52, 107  
 stats\_sent 52, 107, 111, 113  
 stats\_softBounces 52, 107  
 stats\_trackableViews 52, 107  
 stats\_uniqueClicks 52, 107  
 stats\_uniqueViews 52, 107  
 stats\_unsubscribe 111, 113  
 stats\_unsubscriptions 52, 107  
 stats\_viewed 52, 107  
 statsByBrowser 69  
 statsByDomain 69  
 Status 57, 60, 63, 66, 69, 75, 76, 80, 81, 83, 85,  
 99, 105  
 Subject 57, 60, 63, 66, 69, 87, 92, 93, 96, 97, 98,  
 99, 105  
 Subject A 57, 60, 63, 66, 69  
 Subject B 57, 60, 63, 66, 69  
 subjectA 57, 60, 63, 66, 69  
 subjectB 57, 60, 63, 66, 69  
 Subscription Account 17  
 Subscription Account Plans 17  
 SubscriptionAccount 17  
 SubscriptionAccountPlans 17  
 Succesful 74  
 SUCCESSFUL 74  
 swagger-specification-download-tries 2  
 swagger-specification-file 2  
 swagger-specification-http-disk-cache-max-age-sec  
 2  
 swagger-specification-url 2  
  
**- T -**  
 tag 57, 60, 63, 66, 69, 87, 89, 90, 92, 93, 101, 102,  
 103  
 tags 87, 101  
 Template ID 96, 97, 98, 99, 113  
 template\_button\_type 111  
 template\_category 111  
 template\_contains\_button 111  
 template\_display\_header 111  
 template\_header\_type 111  
 template\_hide\_footer 111  
 template\_language 111  
 template\_name 111  
 templateId 87, 92, 96, 97, 98, 99, 113  
 templateStatus 93  
 Test Sent 57, 60, 63, 66, 69, 92, 93  
 testSent 57, 60, 63, 66, 69, 92, 93  
 TEXT 86, 98, 109  
 Time 96



Timeout (sec) 74  
 TIMEOUT\_SEC 74  
 To Field 57, 60, 63, 66, 69, 92, 93  
 toField 57, 60, 63, 66, 69, 92, 93  
 Total Blacklisted 49, 50, 51, 52, 53, 54, 106, 107  
 Total Subscribers 49, 50, 51, 52, 53, 54, 106, 107  
 totalBlacklisted 49, 50, 51, 52, 53, 54, 106, 107  
 totalSubscribers 49, 50, 51, 52, 53, 54, 106, 107  
 totp-secret 2  
 TransacBlockedContacts 94  
 TransacEmailContent 97  
 TransacEmailContent\_Events 96  
 TransacEmailsList 99  
 TransacEmailsList\_TransactionalEmailsTags 98  
 TransacSmsReport 102  
 Transaction Blocked Contacts 94  
 Transaction Email Contents 97  
 Transaction Email Lists 99  
 Transaction ID 74  
 TRANSACTION\_ID 74  
 TransactionAggregatedSmsReports 103  
 Type 17, 19, 57, 60, 63, 66, 69, 108, 109, 110  
 WebhookEventsByWebhookId 109  
 webhookId 108, 109  
 Webhooks 110  
 Weight 77, 78  
 WHATSAPP 36, 41, 44  
 WhatsApp Campaign by ID 111  
 WhatsApp Campaigns 113  
 WhatsApp Templates 115  
 WhatsAppCampaignById 111  
 WhatsAppCampaigns 113  
 WhatsAppTemplates 115  
 Winner Criteria 57, 60, 63, 66, 69  
 Winner Delay 57, 60, 63, 66, 69  
 winnerCriteria 57, 60, 63, 66, 69  
 winnerDelay 57, 60, 63, 66, 69  
 Winning Criteria 55, 105  
 Winning Subject Line 55, 105  
 Winning Version 55, 105  
 Winning Version Rate 55, 105  
 winningCriteria 55, 105  
 winningSubjectLine 55, 105  
 winningVersion 55, 105  
 winningVersionRate 55, 105

## - U -

uniqueClicks 89, 90  
 uniqueOpens 89, 90  
 uniqueSubscribers 49, 50  
 Unsubscribed 89, 90, 102, 103  
 Unsubscribed List IDs 23, 24, 25, 26, 28, 29, 30, 31, 33, 34, 36, 41, 44  
 updatedAt 55  
 URL 38, 74, 108, 109, 110  
 Use HTTP Disk Cache 74  
 Use HTTP Memory Cache 74  
 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  
 UUID 96, 97, 98, 99

## - V -

Value 19

## - W -

Webhook by ID 108  
 WebhookById 108



# *invantive* the **SQL** company

Invantive B.V.  
Biesteweg 11  
3849 RD Hierden  
the Netherlands

Tel: +31 88 00 26 500  
Fax: +31 84 22 58 178  
info@invantive.com  
invantive.com

IBAN NL25 BUNQ 2098 2586 07  
Chamber of Industry and Commerce  
13031406  
VAT NL812602377B01  
RSIN 8122602377  
Managing Director: Guido Leenders  
Registered office: Roermond