

Knack API Data Model

for use with Invantive SQL



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

1	SQL Driver for Knack API	1
2	SQL Driver Attributes for Knack API	2
3	Schema: DataDictionary	14
3.1	Tables	14
3.1.1	ApplicationAccounts	14
3.1.2	ApplicationAccountsByApplicationId	16
3.1.3	ApplicationById: Knack Application by ID	18
3.1.4	ApplicationDesigns	20
3.1.5	ApplicationDesignsByApplicationId	20
3.1.6	Applications: Knack Applications	22
3.1.7	ApplicationSettings: Knack Application Settings	23
3.1.8	ApplicationSettingsByApplicationId	24
3.1.9	ObjectFields	26
3.1.10	ObjectFieldsByApplicationId	27
3.1.11	Objects	28
3.1.12	ObjectsByApplicationId	29
3.1.13	Scenes	30
3.1.14	ScenesByApplicationId	31
3.1.15	SceneView s	32
3.1.16	SceneView sByApplicationId	33
4	Schema: Journal_Details	35
4.1	Tables	35
4.1.1	Journal_Details	35
5	Schema: Journals	36
5.1	Tables	36
5.1.1	Journals: Knack Table	36
6	Schema: Native	37
6.1	Tables	37
6.1.1	NATIVEPLATFORMSCALARREQUESTS: Knack Native Platform Scalar Requests	37
7	Schema: Objects	38
7.1	Tables	38
7.1.1	A_B_C: Knack A B Cs	38
7.1.2	Accounts: Knack Accounts	39
7.1.3	Companies: Knack Companies	40
7.1.4	CompaniesExpanded: Knack Companies	41
7.1.5	Contacts: Knack Contacts	42
7.1.6	ContactsExpanded: Knack Contacts	43
7.1.7	DateTest: Knack DateTests	44
7.1.8	Journals: Knack Journals	44
7.1.9	JournalsMultiple_Choice: Knack Multiple choices for Journals	45
7.1.10	OrdersObject: Knack OrdersObjects	46
7.1.11	Payments: Knack Payments	47
7.1.12	Products: Knack Products	47
7.1.13	ProductsExpanded: Knack Products	48

7.1.14	Test: Knack Tests	49
7.1.15	TestcaseUD: Knack TestcaseUDs	51
7.1.16	TestcaseUDEExpanded: Knack TestcaseUDs	52
7.1.17	TestExpanded: Knack Tests	53
7.1.18	testpho: Knack testphos	55
8	Schema: Product_Details	56
8.1	Tables	56
8.1.1	Product_Details	56
9	Schema: ProductsMenu	57
9.1	Tables	57
9.1.1	Products: Knack Table	57
	Index	58

1 SQL Driver for Knack API

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

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

Knack is a low-code platform to build online databases.

The Knack driver covers 39 tables and 751 columns.

Knack API Clients

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

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

The [Knack Power BI connector](#) is based on the Invantive SQL driver for Knack, 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 Knack API into traditional databases such as SQL Server (on-premise and Azure), MySQL, PostgreSQL and Oracle is possible using [Invantive Data Replicator](#). Invantive Data Replicator automatically creates and maintains Knack datawarehouses, possibly in combination with data from over 70 other (cloud) platforms. Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Knack ADO.net provider.

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

Monitor API Calls

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

Specifications

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

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

The configuration can be changed using various attributes during log on and use. A full list of configuration options is listed in the [driver attributes](#) ².

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 Knack the comparison of two texts is case sensitive by default.

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

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

Alias: `knack`

Recommended alias: `knk`

More technical documentation as provided by the supplier of the Knack API on the native API-connection used can be found at <https://www.knack.com/developer-documentation>.

Authentication can be done using one of the following two alternatives: 1. Using the user log on code and password to access Knack views. 2. Using an API key to access Knack objects. The API key can be found on the 'API & Code' page in Knack available through the settings gear on the top-left corner next to the word 'Integration'. The URL for the API key is `https://builder.knack.com/SAMPLE/integration#settings/api`, with 'SAMPLE' replaced by your slug. The slug is the word before 'knack.com' when running your app.

Usage Limits Invantive SQL executes API calls to retrieve and upload data. The number of API calls allowed per hour depends on your Knack plan. The default usage limits vary depending on plan between 1.000 and 25.000 calls per day, with a limit of 10 calls per second. Invantive SQL ensures that within your session the number of calls allowed per hour is not exceeded. To get an impression of how Invantive SQL translates into API calls, please query the data dictionary view 'sessionios', such as with 'select * from sessionios@datadictionary'.

Updated: 16-06-2022 09:35 using Invantive SQL version 22.1.46-BETA+3385.

2 SQL Driver Attributes for Knack API

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

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

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

The connection string for Knack can be found in the settings*.xml file used for the database. Settings*.xml files are typically located in the `%USERPROFILE%\invantive` folder in most deployment scenarios. The reference manuals contain instructions how to relocate the settings*.xml files. Each data container of a database in the connection string can have a `con-`

nectionString 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 Knack 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	Use for analysis only! Enforce rows to be unique.	Shared	False	✓	✓	✓	
api-key	{res:itgen_knack_api_key_description}	Knack		✓		✓	✓
api-url	URL to access the API.	OData		✓		✓	
application-id	{res:itgen_knack_application_id_description}	Knack		✓		✓	✓
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 OData server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when OData 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 OData 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 OData server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	

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-408-request-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a HTTP status 408.		10	✓	✓	✓	
dow nload-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 408.		10000	✓	✓	✓	
dow nload-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 408.		300000	✓	✓	✓	
dow nload-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a HTTP status 408.		2	✓	✓	✓	
dow nload-error-422-bad-request-max-tries	Maximum number of tries w hen OData server reports unprocessable entity during retrieval of data.		30	✓	✓	✓	
dow nload-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen OData server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen OData server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep betw een retries OData server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
dow nload-error-429-too-many-requests-max-tries	Maximum number of tries w hen the w ebsite 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 betw een retries w hen the w ebsite 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 betw een retries w hen the w ebsite 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 betw een retries w hen the w ebsite reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
dow nload-error-502-server-unavailable-max-tries	Maximum number of tries w hen OData server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
dow nload-error-502-server-unavail-	Initial sleep in milliseconds betw een retries w hen OData server reports a		10000	✓	✓	✓	

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

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
nect-timeout-sleep-max-ms	reports a HTTP status 590.						
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	✓	✓	✓	
dow nload-error-599-netw ork-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 599.		10	✓	✓	✓	
dow nload-error-599-netw ork-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 599.		10000	✓	✓	✓	
dow nload-error-599-netw ork-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 599.		300000	✓	✓	✓	
dow nload-error-599-netw ork-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 599.		2	✓	✓	✓	
dow nload-error-argument-exception-max-tries	Maximum number of tries when an argument exception is returned when downloading a blob.		10	✓	✓	✓	
dow nload-error-argument-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		10000	✓	✓	✓	
dow nload-error-argument-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		300000	✓	✓	✓	
dow nload-error-argument-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.		2	✓	✓	✓	
dow nload-error-internet-dow n-max-tries	Maximum number of tries when the Internet connection seems down during retrieval of data.		10	✓	✓	✓	
dow nload-error-internet-dow n-sleep-initial-ms	Initial sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		10000	✓	✓	✓	
dow nload-error-internet-dow n-sleep-max-ms	Maximum sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		300000	✓	✓	✓	
dow nload-error-internet-dow n-sleep-multiplicator	Multiplication factor for sleep between retries when the Internet connection seems down 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-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	✓	✓	✓	
download-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	✓	✓	✓	
download-error-io-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a network I/O connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-json-exception-max-tries	Maximum number of tries when an invalid JSON body is returned.		3	✓	✓	✓	
download-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an invalid JSON body is returned.		1000	✓	✓	✓	
download-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an invalid JSON body is returned.		10000	✓	✓	✓	
download-error-json-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an invalid JSON body is returned.		2	✓	✓	✓	
download-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
download-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		300000	✓	✓	✓	
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error occurs during retrieval of data.		2	✓	✓	✓	
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	✓	✓	✓	

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-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcibly dropped during retrieval of data.		2	✓	✓	✓	
dow nload-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
dow nload-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
dow nload-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
dow nload-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
dow nload-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
dow nload-error-web-not-implemented-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports not implemented.		10000	✓	✓	✓	
dow nload-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	
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	✓	✓	✓	

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-web-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports an unauthorized error.		300000	✓	✓	✓	
download-error-web-unauthorized-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports an unauthorized error.		2	✓	✓	✓	
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	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.	Shared	C:\Users\gle3.WS212\In-vantive\Cache\http\gle3\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).		300000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		60000	✓	✓	✓	
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).		300000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		300000	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	dpoint.						
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		10	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		300000	✓	✓	✓	
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.		10000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		300000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
invantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-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	✓	✓	✓	

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-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
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 events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when an error occurred.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when successful.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
max-odata-filters	The maximum number of OData filter elements.	OData	100	✓	✓	✓	
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	✓	✓	✓	
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	✓	✓	✓	
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	OData	1000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	tries when the OAuth authentication fails.						
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 ms 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	Simulate HTTP 400 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	

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-500-errors	Simulate HTTP 500 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in ms across all slots of a slot-based rate limit.	Shared	1000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit	Shared	10	✓		✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
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	✓	✓	✓	
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	True	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.	Shared	True	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
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 Schema: DataDictionary

3.1 Tables

3.1.1 ApplicationAccounts

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: `/applications/5f68406d6618ae0015531148`

Delete Knack API URL: `/applications/5f68406d6618ae0015531148`

Field Selection Method: NotRequired

Base Path: `application`

Table Columns

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

Name	Data Type	Label	Required	Documentation
<code>api_limit_extra_quantity</code>	int64		<input type="checkbox"/>	
<code>application_id</code>	string	Application ID	<input checked="" type="checkbox"/>	
<code>application_name</code>	string		<input checked="" type="checkbox"/>	
<code>beta_deadline</code>	datetime		<input type="checkbox"/>	
<code>billing_custom_info</code>	string		<input type="checkbox"/>	
<code>billing_customer_id</code>	string		<input type="checkbox"/>	
<code>billing_email</code>	string		<input type="checkbox"/>	
<code>billing_has_paid</code>	boolean		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
billing_ignore_capacity_until	datetime		<input type="checkbox"/>	
billing_ignore_delinquency_until	datetime		<input type="checkbox"/>	
billing_last_freeze_date	datetime		<input type="checkbox"/>	
billing_notice_count	int32		<input type="checkbox"/>	
billing_notice_date	datetime		<input type="checkbox"/>	
billing_status	string		<input type="checkbox"/>	
counts_builders	int64		<input type="checkbox"/>	
id	string		<input type="checkbox"/>	
name	string		<input type="checkbox"/>	
plan_limits_api_limit	int64		<input type="checkbox"/>	
plan_limits_applications	int64		<input type="checkbox"/>	
plan_limits_records	int64		<input type="checkbox"/>	
plan_limits_storage	int64		<input type="checkbox"/>	
product_plan_api_limit	int64		<input type="checkbox"/>	
product_plan_apps	int64		<input type="checkbox"/>	
product_plan_babu_unit	string		<input type="checkbox"/>	
product_plan_babu_value	int64		<input type="checkbox"/>	
product_plan_builders	int64		<input type="checkbox"/>	
product_plan_id	string		<input type="checkbox"/>	
product_plan_level	int32		<input type="checkbox"/>	
product_plan_name	string		<input type="checkbox"/>	
product_plan_price	decimal		<input type="checkbox"/>	
product_plan_records	int64		<input type="checkbox"/>	
product_plan_storage	string		<input type="checkbox"/>	
product_plan_subscription_id	string		<input type="checkbox"/>	
product_plan_type	string		<input type="checkbox"/>	
referral	string		<input type="checkbox"/>	
settings_agreements	string		<input type="checkbox"/>	
settings_force_region_for_new_apps_enabled	boolean		<input type="checkbox"/>	
settings_force_region_for_new_apps_region	string		<input type="checkbox"/>	
settings_has_open_beta_access	boolean		<input type="checkbox"/>	
settings_has_opted_into_open_beta	boolean		<input type="checkbox"/>	
settings_hipaa_enabled	boolean		<input type="checkbox"/>	
settings_hipaa_region	string		<input type="checkbox"/>	
settings_mongo	string		<input type="checkbox"/>	
settings_new_apps_are_sql	boolean		<input type="checkbox"/>	
settings_rate_limit_login_requests	string		<input type="checkbox"/>	
settings_security_builder_inactivity_message	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
settings_security_builder_inactivity_timeout_enabled	boolean		<input type="checkbox"/>	
settings_security_builder_inactivity_timeout	int64		<input type="checkbox"/>	
slug	string		<input type="checkbox"/>	
sso	boolean		<input type="checkbox"/>	
status	string		<input type="checkbox"/>	
user_id	string		<input type="checkbox"/>	

3.1.2 ApplicationAccountsByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/{id}`

Insert Knack API URL: `/applications/{id}`

Update Knack API URL: `/applications/{id}`

Delete Knack API URL: `/applications/{id}`

Field Selection Method: NotRequired

Base Path: `application`

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
api_limit_extra_quantity	int64		<input type="checkbox"/>	
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
beta_deadline	datetime		<input type="checkbox"/>	
billing_custom_info	string		<input type="checkbox"/>	
billing_customer_id	string		<input type="checkbox"/>	
billing_email	string		<input type="checkbox"/>	
billing_has_paid	boolean		<input type="checkbox"/>	
billing_ignore_capacity_until	datetime		<input type="checkbox"/>	
billing_ignore_delinquency_until	datetime		<input type="checkbox"/>	
billing_last_freeze_date	datetime		<input type="checkbox"/>	
billing_notice_count	int32		<input type="checkbox"/>	
billing_notice_date	datetime		<input type="checkbox"/>	
billing_status	string		<input type="checkbox"/>	
counts_builders	int64		<input type="checkbox"/>	
id	string		<input type="checkbox"/>	
name	string		<input type="checkbox"/>	
plan_limits_api_limit	int64		<input type="checkbox"/>	
plan_limits_applications	int64		<input type="checkbox"/>	
plan_limits_records	int64		<input type="checkbox"/>	
plan_limits_storage	int64		<input type="checkbox"/>	
product_plan_api_limit	int64		<input type="checkbox"/>	
product_plan_apps	int64		<input type="checkbox"/>	
product_plan_babu_unit	string		<input type="checkbox"/>	
product_plan_babu_value	int64		<input type="checkbox"/>	
product_plan_builders	int64		<input type="checkbox"/>	
product_plan_id	string		<input type="checkbox"/>	
product_plan_level	int32		<input type="checkbox"/>	
product_plan_name	string		<input type="checkbox"/>	
product_plan_price	decimal		<input type="checkbox"/>	
product_plan_records	int64		<input type="checkbox"/>	
product_plan_storage	string		<input type="checkbox"/>	
product_plan_subscription_id	string		<input type="checkbox"/>	
product_plan_type	string		<input type="checkbox"/>	
referral	string		<input type="checkbox"/>	
settings_agreements	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
settings_force_region_for_new_apps_enabled	boolean		<input type="checkbox"/>	
settings_force_region_for_new_apps_region	string		<input type="checkbox"/>	
settings_has_open_beta_access	boolean		<input type="checkbox"/>	
settings_has_opted_into_open_beta	boolean		<input type="checkbox"/>	
settings_hipaa_enabled	boolean		<input type="checkbox"/>	
settings_hipaa_region	string		<input type="checkbox"/>	
settings_mongo	string		<input type="checkbox"/>	
settings_new_apps_are_sql	boolean		<input type="checkbox"/>	
settings_rate_limit_login_requests	string		<input type="checkbox"/>	
settings_security_builder_inactivity_message	string		<input type="checkbox"/>	
settings_security_builder_inactivity_timeout_enabled	boolean		<input type="checkbox"/>	
settings_security_builder_inactivity_timeout	int64		<input type="checkbox"/>	
slug	string		<input type="checkbox"/>	
sso	boolean		<input type="checkbox"/>	
status	string		<input type="checkbox"/>	
user_id	string		<input type="checkbox"/>	

3.1.3 ApplicationById: Knack Application by ID

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

Label: Application by ID

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

Select Knack API URL: `/applications/{id}`

Insert Knack API URL: `/applications/{id}`

Update Knack API URL: `/applications/{id}`

Delete Knack API URL: `/applications/{id}`

Field Selection Method: NotRequired

Base Path: `application`

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
ecommerce_enabled	boolean		<input type="checkbox"/>	
HERE_appCode	string		<input type="checkbox"/>	
HERE_appId	string		<input type="checkbox"/>	
home_scene_key	string		<input type="checkbox"/>	
home_scene_slug	string		<input type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
layout_app_fullwidth	boolean		<input type="checkbox"/>	
layout_app_menu_auth	boolean		<input type="checkbox"/>	
layout_entry_scene_menu	boolean		<input type="checkbox"/>	
layout_page_menu_style	string		<input type="checkbox"/>	
layout_theme	string		<input type="checkbox"/>	
logo_url	string		<input type="checkbox"/>	
name	string		<input checked="" type="checkbox"/>	
s3_bucket	string		<input type="checkbox"/>	
s3_domain	string		<input type="checkbox"/>	
s3_secure_bucket	string		<input type="checkbox"/>	
s3_secure_domain	string		<input type="checkbox"/>	
slug	string		<input type="checkbox"/>	
status	string		<input type="checkbox"/>	
subdomain	string		<input type="checkbox"/>	
users_enabled	boolean		<input type="checkbox"/>	
users_registration	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
users_scope	string		<input type="checkbox"/>	

3.1.4 ApplicationDesigns

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: `/applications/5f68406d6618ae0015531148`

Delete Knack API URL: `/applications/5f68406d6618ae0015531148`

Field Selection Method: NotRequired

Base Path: `application`

Table Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
button_bg_color	string		<input type="checkbox"/>	
button_border_color	string		<input type="checkbox"/>	
button_hover_color	string		<input type="checkbox"/>	
button_text_color	string		<input type="checkbox"/>	
button_textshadow_style	string		<input type="checkbox"/>	
header_bg_color	string		<input type="checkbox"/>	
header_text_color	string		<input type="checkbox"/>	
header_textshadow_style	string		<input type="checkbox"/>	
link_text_color	string		<input type="checkbox"/>	
notification_bg_color	string		<input type="checkbox"/>	
notification_text_color	string		<input type="checkbox"/>	

3.1.5 ApplicationDesignsByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/{id}`

Insert Knack API URL: `/applications/{id}`

Update Knack API URL: `/applications/{id}`

Delete Knack API URL: `/applications/{id}`

Field Selection Method: NotRequired

Base Path: `application`

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
button_bg_color	string		<input type="checkbox"/>	
button_border_color	string		<input type="checkbox"/>	
button_hover_color	string		<input type="checkbox"/>	
button_text_color	string		<input type="checkbox"/>	
button_textshadow_style	string		<input type="checkbox"/>	
header_bg_color	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
header_text_color	string		<input type="checkbox"/>	
header_textshadow_style	string		<input type="checkbox"/>	
link_text_color	string		<input type="checkbox"/>	
notification_bg_color	string		<input type="checkbox"/>	
notification_text_color	string		<input type="checkbox"/>	

3.1.6 Applications: Knack Applications

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

Label: Applications

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: `/applications/5f68406d6618ae0015531148`

Delete Knack API URL: `/applications/5f68406d6618ae0015531148`

Field Selection Method: NotRequired

Base Path: `application`

Table Columns

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

Name	Data Type	Label	Required	Documentation
ecommerce_enabled	boolean		<input type="checkbox"/>	
HERE_appCode	string		<input type="checkbox"/>	
HERE_appld	string		<input type="checkbox"/>	
home_scene_key	string		<input type="checkbox"/>	
home_scene_slug	string		<input type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
layout_app_fullwidth	boolean		<input type="checkbox"/>	
layout_app_menu_auth	boolean		<input type="checkbox"/>	
layout_entry_scene_menu	boolean		<input type="checkbox"/>	
layout_page_menu_style	string		<input type="checkbox"/>	
layout_theme	string		<input type="checkbox"/>	
logo_url	string		<input type="checkbox"/>	
name	string		<input checked="" type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
s3_bucket	string		<input type="checkbox"/>	
s3_domain	string		<input type="checkbox"/>	
s3_secure_bucket	string		<input type="checkbox"/>	
s3_secure_domain	string		<input type="checkbox"/>	
slug	string		<input type="checkbox"/>	
status	string		<input type="checkbox"/>	
subdomain	string		<input type="checkbox"/>	
users_enabled	boolean		<input type="checkbox"/>	
users_registration	string		<input type="checkbox"/>	
users_scope	string		<input type="checkbox"/>	

3.1.7 ApplicationSettings: Knack Application Settings

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

Label: Application Settings

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: `/applications/5f68406d6618ae0015531148`

Delete Knack API URL: `/applications/5f68406d6618ae0015531148`

Field Selection Method: NotRequired

Base Path: `application`

Table Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
babu_preseed_status	string		<input type="checkbox"/>	
cluster	string		<input type="checkbox"/>	
css	string		<input type="checkbox"/>	
from_email	string		<input type="checkbox"/>	
geo	boolean		<input type="checkbox"/>	
has_new_object_types	boolean		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
https_redirect	boolean		<input type="checkbox"/>	
icons	boolean		<input type="checkbox"/>	
inactivity_message	string		<input type="checkbox"/>	
inactivity_timeout_enabled	boolean		<input type="checkbox"/>	
inactivity_timeout	int64		<input type="checkbox"/>	
ip_w hitelist	string		<input type="checkbox"/>	
javascript	string		<input type="checkbox"/>	
language	string		<input type="checkbox"/>	
last_update	string		<input type="checkbox"/>	
legacy_entry_history_nuked	string		<input type="checkbox"/>	
logo	string		<input type="checkbox"/>	
mongo	string		<input type="checkbox"/>	
new_count	boolean		<input type="checkbox"/>	
powered_by_link	boolean		<input type="checkbox"/>	
registration_migration_status	string		<input type="checkbox"/>	
runner	string		<input type="checkbox"/>	
script_protection_enabled	boolean		<input type="checkbox"/>	
should_purge_record_history	boolean		<input type="checkbox"/>	
solr	string		<input type="checkbox"/>	
support_access	boolean		<input type="checkbox"/>	
technical_contact	string		<input type="checkbox"/>	
timezone_dst	string		<input type="checkbox"/>	
timezone_offset	string		<input type="checkbox"/>	
timezone	string		<input type="checkbox"/>	
use_allowed_html_w hitelist	boolean		<input type="checkbox"/>	
use_multiple_api_subdomains	boolean		<input type="checkbox"/>	
v3_beta	boolean		<input type="checkbox"/>	
v3_open_beta	boolean		<input type="checkbox"/>	

3.1.8 ApplicationSettingsByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: /applications/{id}

Insert Knack API URL: /applications/{id}

Update Knack API URL: /applications/{id}

Delete Knack API URL: `/applications/{id}`

Field Selection Method: NotRequired

Base Path: `application`

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
babu_preseed_status	string		<input type="checkbox"/>	
cluster	string		<input type="checkbox"/>	
css	string		<input type="checkbox"/>	
from_email	string		<input type="checkbox"/>	
geo	boolean		<input type="checkbox"/>	
has_new_object_types	boolean		<input type="checkbox"/>	
https_redirect	boolean		<input type="checkbox"/>	
icons	boolean		<input type="checkbox"/>	
inactivity_message	string		<input type="checkbox"/>	
inactivity_timeout_enabled	boolean		<input type="checkbox"/>	
inactivity_timeout	int64		<input type="checkbox"/>	
ip_w hitelist	string		<input type="checkbox"/>	
javascript	string		<input type="checkbox"/>	
language	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
last_update	string		<input type="checkbox"/>	
legacy_entry_history_nuked	string		<input type="checkbox"/>	
logo	string		<input type="checkbox"/>	
mongo	string		<input type="checkbox"/>	
new_count	boolean		<input type="checkbox"/>	
powered_by_link	boolean		<input type="checkbox"/>	
registration_migration_status	string		<input type="checkbox"/>	
runner	string		<input type="checkbox"/>	
script_protection_enabled	boolean		<input type="checkbox"/>	
should_purge_record_history	boolean		<input type="checkbox"/>	
solr	string		<input type="checkbox"/>	
support_access	boolean		<input type="checkbox"/>	
technical_contact	string		<input type="checkbox"/>	
timezone_dst	string		<input type="checkbox"/>	
timezone_offset	string		<input type="checkbox"/>	
timezone	string		<input type="checkbox"/>	
use_allowed_html_whitelist	boolean		<input type="checkbox"/>	
use_multiple_api_subdomains	boolean		<input type="checkbox"/>	
v3_beta	boolean		<input type="checkbox"/>	
v3_open_beta	boolean		<input type="checkbox"/>	

3.1.9 ObjectFields

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: `/applications/5f68406d6618ae0015531148`

Delete Knack API URL: `/applications/5f68406d6618ae0015531148`

Field Selection Method: NotRequired

Base Path: `application.objects[*].fields[*]`

Table Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
conditional	boolean		<input type="checkbox"/>	
default	string		<input type="checkbox"/>	
format_format	string		<input type="checkbox"/>	
format_label	string		<input type="checkbox"/>	
id	string		<input type="checkbox"/>	Unique ID.
immutable	boolean		<input type="checkbox"/>	
object_id	string		<input checked="" type="checkbox"/>	
object_key	string		<input checked="" type="checkbox"/>	
object_name	string		<input checked="" type="checkbox"/>	
required	boolean		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	
unique	boolean		<input type="checkbox"/>	
user	boolean		<input type="checkbox"/>	

3.1.10 ObjectFieldsByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/{id}`

Insert Knack API URL: `/applications/{id}`

Update Knack API URL: `/applications/{id}`

Delete Knack API URL: `/applications/{id}`

Field Selection Method: NotRequired

Base Path: `application.objects[*].fields[*]`

Parameters of Table Function

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

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
conditional	boolean		<input type="checkbox"/>	
default	string		<input type="checkbox"/>	
format_format	string		<input type="checkbox"/>	
format_label	string		<input type="checkbox"/>	
id	string		<input type="checkbox"/>	Unique ID.
immutable	boolean		<input type="checkbox"/>	
object_id	string		<input checked="" type="checkbox"/>	
object_key	string		<input checked="" type="checkbox"/>	
object_name	string		<input checked="" type="checkbox"/>	
required	boolean		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	
unique	boolean		<input type="checkbox"/>	
user	boolean		<input type="checkbox"/>	

3.1.11 Objects

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: `/applications/5f68406d6618ae0015531148`

Delete Knack API URL: /applications/5f68406d6618ae0015531148

Field Selection Method: NotRequired

Base Path: application.objects[*]

Table Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
identifier	string		<input type="checkbox"/>	
inflections_plural	string		<input type="checkbox"/>	
inflections_singular	string		<input type="checkbox"/>	
key	string		<input checked="" type="checkbox"/>	
name	string		<input checked="" type="checkbox"/>	
sort_field	string		<input type="checkbox"/>	
sort_order	string		<input type="checkbox"/>	
status	string		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	
user	string		<input type="checkbox"/>	

3.1.12 ObjectsByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: /applications/{id}

Insert Knack API URL: /applications/{id}

Update Knack API URL: /applications/{id}

Delete Knack API URL: /applications/{id}

Field Selection Method: NotRequired

Base Path: application.objects[*]

Parameters of Table Function

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

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
identifier	string		<input type="checkbox"/>	
inflections_plural	string		<input type="checkbox"/>	
inflections_singular	string		<input type="checkbox"/>	
key	string		<input checked="" type="checkbox"/>	
name	string		<input checked="" type="checkbox"/>	
sort_field	string		<input type="checkbox"/>	
sort_order	string		<input type="checkbox"/>	
status	string		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	
user	string		<input type="checkbox"/>	

3.1.13 Scenes

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/5f68406d6618ae0015531148`

Insert Knack API URL: `/applications/5f68406d6618ae0015531148`

Update Knack API URL: /applications/5f68406d6618ae0015531148

Delete Knack API URL: /applications/5f68406d6618ae0015531148

Field Selection Method: NotRequired

Base Path: application.scenes[*]

Table Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
authenticated	boolean		<input type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
key	string		<input checked="" type="checkbox"/>	
limit_profile_access	string		<input type="checkbox"/>	
menu	string		<input type="checkbox"/>	
modal	boolean		<input type="checkbox"/>	
object_key	string		<input type="checkbox"/>	
parent	string		<input type="checkbox"/>	
print	boolean		<input type="checkbox"/>	
slug	string		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	

3.1.14 ScenesByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: /applications/{id}

Insert Knack API URL: /applications/{id}

Update Knack API URL: /applications/{id}

Delete Knack API URL: /applications/{id}

Field Selection Method: NotRequired

Base Path: application.scenes[*]

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
authenticated	boolean		<input type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
key	string		<input checked="" type="checkbox"/>	
limit_profile_access	string		<input type="checkbox"/>	
menu	string		<input type="checkbox"/>	
modal	boolean		<input type="checkbox"/>	
object_key	string		<input type="checkbox"/>	
parent	string		<input type="checkbox"/>	
print	boolean		<input type="checkbox"/>	
slug	string		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	

3.1.15 SceneViews

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: /applications/5f68406d6618ae0015531148

Insert Knack API URL: /applications/5f68406d6618ae0015531148

Update Knack API URL: /applications/5f68406d6618ae0015531148

Delete Knack API URL: /applications/5f68406d6618ae0015531148

Field Selection Method: NotRequired

Base Path: application.scenes[*].views[*]

Table Columns

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

Name	Data Type	Label	Required	Documentation
action	string		<input type="checkbox"/>	
alert	string		<input type="checkbox"/>	
allow_exporting	boolean		<input type="checkbox"/>	
allow_preset_filters	boolean		<input type="checkbox"/>	
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
description	string		<input type="checkbox"/>	
display_pagination_below	boolean		<input type="checkbox"/>	
filter_type	string		<input type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.
ignore_reload	boolean		<input type="checkbox"/>	
key	string		<input checked="" type="checkbox"/>	
keyword_search	boolean		<input type="checkbox"/>	
label	string		<input type="checkbox"/>	
layout	string		<input type="checkbox"/>	
limit_profile_access	boolean		<input type="checkbox"/>	
parent	string		<input type="checkbox"/>	
rows_per_page	int32		<input type="checkbox"/>	
scene_id	string		<input checked="" type="checkbox"/>	
scene_key	string		<input checked="" type="checkbox"/>	
scene_name	string		<input checked="" type="checkbox"/>	
title	string		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	

3.1.16 SceneViewsByApplicationId

Catalog: Knack

Schema: DataDictionary

Primary Keys: id

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

Select Knack API URL: `/applications/{id}`

Insert Knack API URL: `/applications/{id}`

Update Knack API URL: `/applications/{id}`

Delete Knack API URL: `/applications/{id}`

Field Selection Method: `NotRequired`

Base Path: `application.scenes[*].views[*]`

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
action	string		<input type="checkbox"/>	
alert	string		<input type="checkbox"/>	
allow_exporting	boolean		<input type="checkbox"/>	
allow_preset_filters	boolean		<input type="checkbox"/>	
application_id	string	Application ID	<input checked="" type="checkbox"/>	
application_name	string		<input checked="" type="checkbox"/>	
description	string		<input type="checkbox"/>	
display_pagination_below	boolean		<input type="checkbox"/>	
filter_type	string		<input type="checkbox"/>	
id	string		<input checked="" type="checkbox"/>	Unique ID.

Name	Data Type	Label	Required	Documentation
ignore_reload	boolean		<input type="checkbox"/>	
key	string		<input checked="" type="checkbox"/>	
keyword_search	boolean		<input type="checkbox"/>	
label	string		<input type="checkbox"/>	
layout	string		<input type="checkbox"/>	
limit_profile_access	boolean		<input type="checkbox"/>	
parent	string		<input type="checkbox"/>	
rows_per_page	int32		<input type="checkbox"/>	
scene_id	string		<input checked="" type="checkbox"/>	
scene_key	string		<input checked="" type="checkbox"/>	
scene_name	string		<input checked="" type="checkbox"/>	
title	string		<input type="checkbox"/>	
type	string		<input type="checkbox"/>	

4 Schema: Journal_Details

4.1 Tables

4.1.1 Journal_Details

Catalog: Knack

Schema: Journal_Details

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

Select Knack API URL: `/pages/scene_3/views/view_4/records/{id}`

Insert Knack API URL: `/pages/scene_3/views/view_4/records/{id}`

Update Knack API URL: `/pages/scene_3/views/view_4/records/{id}`

Delete Knack API URL: `/pages/scene_3/views/view_4/records/{id}`

Field Selection Method: NotRequired

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
DUMMY	string		<input checked="" type="checkbox"/>	Default column added since the specification specifies that no data is returned.
record_id	string		<input checked="" type="checkbox"/>	

5 Schema: Journals

5.1 Tables

5.1.1 Journals: Knack Table

Catalog: Knack

Schema: Journals

Label: Table

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

Select Knack API URL: `/pages/scene_1/views/view_1/records`

Insert Knack API URL: `/pages/scene_1/views/view_1/records`

Update Knack API URL: `/pages/scene_1/views/view_1/records`

Delete Knack API URL: `/pages/scene_1/views/view_1/records`

Field Selection Method: NotRequired

Base Path: `records[*]`

View Columns

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

Name	Data Type	Label	Required	Documentation
Code	string		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

6 Schema: Native

6.1 Tables

6.1.1 NATIVEPLATFORMSCALARREQUESTS: Knack Native Platform Scalar Requests

Direct access to native API.

Catalog: Knack

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 Knack 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 Knack 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	int32	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 request 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.

Name	Data Type	Label	Required	Documentation
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	int16	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		<input type="checkbox"/>	
RESULT_NUMBER	decimal		<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	int32	Transaction ID	<input checked="" type="checkbox"/>	Incrementing ID of the transaction.
URL	string(4000)	URL	<input type="checkbox"/>	

7 Schema: Objects

7.1 Tables

7.1.1 A_B_C: Knack A B Cs

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: A B Cs

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

Select Knack API URL: /objects/object_5/records

Insert Knack API URL: /objects/object_5/records

Update Knack API URL: /objects/object_5/records

Delete Knack API URL: /objects/object_5/records

Field Selection Method: NotRequired

Base Path: records [*]

Table Columns

The columns of the table `A_B_C` 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 or update.

Name	Data Type	Label	Required	Documentation
A_B_C_Name	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.2 Accounts: Knack Accounts

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Accounts

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

Select Knack API URL: `/objects/object_2/records`

Insert Knack API URL: `/objects/object_2/records`

Update Knack API URL: `/objects/object_2/records`

Delete Knack API URL: `/objects/object_2/records`

Field Selection Method: NotRequired

Base Path: `records[*]`

Table Columns

The columns of the table `Accounts` 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 or update.

Name	Data Type	Label	Required	Documentation
Email_email	string		<input type="checkbox"/>	
Email_label	string		<input type="checkbox"/>	
Name_firstname	string		<input type="checkbox"/>	
Name_lastname	string		<input type="checkbox"/>	
Name_middlename	string		<input type="checkbox"/>	
Name_title	string		<input type="checkbox"/>	
Password	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
User_Roles	string		<input type="checkbox"/>	
User_Status	string		<input type="checkbox"/>	

7.1.3 Companies: Knack Companies

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Companies

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

Select Knack API URL: /objects/object_6/records

Insert Knack API URL: /objects/object_6/records

Update Knack API URL: /objects/object_6/records

Delete Knack API URL: /objects/object_6/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `Companies` 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 or update.

Name	Data Type	Label	Required	Documentation
Address_city	string		<input type="checkbox"/>	
Address_state	string		<input type="checkbox"/>	
Address_street	string		<input type="checkbox"/>	
Address_street2	string		<input type="checkbox"/>	
Address_zip	string		<input type="checkbox"/>	
Companies_Name	string		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Industry	string		<input type="checkbox"/>	
Logo_filename	string		<input type="checkbox"/>	
Logo_s3	boolean		<input type="checkbox"/>	
Logo_size	int32		<input type="checkbox"/>	
Logo_thumb_url	string		<input type="checkbox"/>	
Logo_type	string		<input type="checkbox"/>	
Logo_url	string		<input type="checkbox"/>	
Notes	string		<input type="checkbox"/>	
Phone	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Website_label	string		<input type="checkbox"/>	
Website_url	string		<input type="checkbox"/>	

7.1.4 CompaniesExpanded: Knack Companies

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Companies

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

Select Knack API URL: /objects/object_6/records

Insert Knack API URL: /objects/object_6/records

Update Knack API URL: /objects/object_6/records

Delete Knack API URL: /objects/object_6/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `CompaniesExpanded` 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 or update.

Name	Data Type	Label	Required	Documentation
Address_city	string		<input type="checkbox"/>	
Address_state	string		<input type="checkbox"/>	
Address_street	string		<input type="checkbox"/>	
Address_street2	string		<input type="checkbox"/>	
Address_zip	string		<input type="checkbox"/>	
Companies_Name	string		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Industry	string		<input type="checkbox"/>	
Logo_data	byte[]		<input type="checkbox"/>	
Logo_filename	string		<input type="checkbox"/>	
Logo_s3	boolean		<input type="checkbox"/>	
Logo_size	int32		<input type="checkbox"/>	
Logo_thumb_url	string		<input type="checkbox"/>	
Logo_type	string		<input type="checkbox"/>	
Logo_url	string		<input type="checkbox"/>	
Notes	string		<input type="checkbox"/>	
Phone	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Website_label	string		<input type="checkbox"/>	
Website_url	string		<input type="checkbox"/>	

7.1.5 Contacts: Knack Contacts

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Contacts

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

Select Knack API URL: /objects/object_7/records

Insert Knack API URL: /objects/object_7/records

Update Knack API URL: /objects/object_7/records

Delete Knack API URL: /objects/object_7/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `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 or update.

Name	Data Type	Label	Required	Documentation
Address_city	string		<input type="checkbox"/>	
Address_state	string		<input type="checkbox"/>	
Address_street	string		<input type="checkbox"/>	
Address_street2	string		<input type="checkbox"/>	
Address_zip	string		<input type="checkbox"/>	
Birthday	datetime		<input type="checkbox"/>	
Contacts_Name	string		<input type="checkbox"/>	
Email_email	string		<input type="checkbox"/>	
Email_label	string		<input type="checkbox"/>	
Gender	string		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
Notes	string		<input type="checkbox"/>	
Occupation	string		<input type="checkbox"/>	
Phone	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.6 ContactsExpanded: Knack Contacts

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Contacts

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

Select Knack API URL: /objects/object_7/records

Insert Knack API URL: /objects/object_7/records

Update Knack API URL: /objects/object_7/records

Delete Knack API URL: /objects/object_7/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `ContactsExpanded` 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 or update.

Name	Data Type	Label	Required	Documentation
Address_city	string		<input type="checkbox"/>	
Address_state	string		<input type="checkbox"/>	
Address_street	string		<input type="checkbox"/>	
Address_street2	string		<input type="checkbox"/>	
Address_zip	string		<input type="checkbox"/>	
Birthday	datetime		<input type="checkbox"/>	
Contacts_Name	string		<input type="checkbox"/>	
Email_email	string		<input type="checkbox"/>	
Email_label	string		<input type="checkbox"/>	
Gender	string		<input type="checkbox"/>	
Image_data	byte[]		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
Notes	string		<input type="checkbox"/>	
Occupation	string		<input type="checkbox"/>	
Phone	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.7 DateTest: Knack DateTests

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: DateTests

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

Select Knack API URL: /objects/object_10/records

Insert Knack API URL: /objects/object_10/records

Update Knack API URL: /objects/object_10/records

Delete Knack API URL: /objects/object_10/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `DateTest` 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 or update.

Name	Data Type	Label	Required	Documentation
a	datetime		<input type="checkbox"/>	
n	decimal		<input type="checkbox"/>	
Name_firstname	string		<input type="checkbox"/>	
Name_lastname	string		<input type="checkbox"/>	
Name_middlename	string		<input type="checkbox"/>	
Name_title	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
t	string		<input type="checkbox"/>	

7.1.8 Journals: Knack Journals

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Journals

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

Select Knack API URL: /objects/object_1/records

Insert Knack API URL: /objects/object_1/records

Update Knack API URL: /objects/object_1/records

Delete Knack API URL: /objects/object_1/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `Journals` 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 or update.

Name	Data Type	Label	Required	Documentation
Code	string		<input type="checkbox"/>	
Company_id	string		<input type="checkbox"/>	
Company_identifier	string		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Multiple_Choice	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Text_Formula	string		<input type="checkbox"/>	

7.1.9 JournalsMultiple_Choice: Knack Multiple choices for Journals

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Multiple choices for Journals

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

Select Knack API URL: /objects/object_1/records

Insert Knack API URL: /objects/object_1/records

Update Knack API URL: /objects/object_1/records

Delete Knack API URL: /objects/object_1/records

Field Selection Method: NotRequired

Base Path: records[*].field_82_raw[*]

Table Columns

The columns of the table `JournalsMultiple_Choice` 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 or update.

Name	Data Type	Label	Required	Documentation
record_id	string		<input checked="" type="checkbox"/>	
value	string		<input checked="" type="checkbox"/>	

7.1.10 OrdersObject: Knack OrdersObjects

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: OrdersObjects

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

Select Knack API URL: /objects/object_8/records

Insert Knack API URL: /objects/object_8/records

Update Knack API URL: /objects/object_8/records

Delete Knack API URL: /objects/object_8/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `OrdersObject` 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 or update.

Name	Data Type	Label	Required	Documentation
Customer	string		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Order_Date	datetime		<input type="checkbox"/>	
Order_ID	decimal		<input type="checkbox"/>	
Orders_Name	string		<input type="checkbox"/>	
Product_Name	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Ship_Date	datetime		<input type="checkbox"/>	
Ship_Type	string		<input type="checkbox"/>	
Shipping_Address_city	string		<input type="checkbox"/>	
Shipping_Address_state	string		<input type="checkbox"/>	
Shipping_Address_street	string		<input type="checkbox"/>	
Shipping_Address_street2	string		<input type="checkbox"/>	
Shipping_Address_zip	string		<input type="checkbox"/>	
Status	string		<input type="checkbox"/>	
Total	decimal		<input type="checkbox"/>	

7.1.11 Payments: Knack Payments

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Payments

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

Select Knack API URL: /objects/object_9/records

Insert Knack API URL: /objects/object_9/records

Update Knack API URL: /objects/object_9/records

Delete Knack API URL: /objects/object_9/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `Payments` 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 or update.

Name	Data Type	Label	Required	Documentation
Account_id	string		<input type="checkbox"/>	
Account_identifier	string		<input type="checkbox"/>	
Amount	decimal		<input type="checkbox"/>	
Charge	string		<input type="checkbox"/>	
Date	datetime		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
ID	string		<input type="checkbox"/>	
IP_Address	string		<input type="checkbox"/>	
Notes	string		<input type="checkbox"/>	
Processor	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Source	string		<input type="checkbox"/>	
Status	string		<input type="checkbox"/>	

7.1.12 Products: Knack Products

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Products

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

Select Knack API URL: /objects/object_3/records

Insert Knack API URL: /objects/object_3/records

Update Knack API URL: /objects/object_3/records

Delete Knack API URL: /objects/object_3/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table `Products` 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 or update.

Name	Data Type	Label	Required	Documentation
Available	boolean		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
Price	decimal		<input type="checkbox"/>	
Product_ID	decimal		<input type="checkbox"/>	
Products_Name_id	string		<input type="checkbox"/>	
Products_Name_identifier	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.13 ProductsExpanded: Knack Products

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Products

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

Select Knack API URL: /objects/object_3/records

Insert Knack API URL: /objects/object_3/records

Update Knack API URL: /objects/object_3/records

Delete Knack API URL: /objects/object_3/records

Field Selection Method: NotRequired

Base Path: records [*]

Table Columns

The columns of the table `ProductsExpanded` 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 or update.

Name	Data Type	Label	Required	Documentation
Available	boolean		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Image_data	byte[]		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
Price	decimal		<input type="checkbox"/>	
Product_ID	decimal		<input type="checkbox"/>	
Products_Name_id	string		<input type="checkbox"/>	
Products_Name_identifier	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.14 Test: Knack Tests

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Tests

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

Select Knack API URL: /objects/object_4/records

Insert Knack API URL: /objects/object_4/records

Update Knack API URL: /objects/object_4/records

Delete Knack API URL: /objects/object_4/records

Field Selection Method: NotRequired

Base Path: records [*]

Table Columns

The columns of the table `Test` 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 or update.

Name	Data Type	Label	Required	Documentation
AddressWithGeocoding_city	string		<input type="checkbox"/>	
AddressWithGeocoding_latitude	decimal		<input type="checkbox"/>	
AddressWithGeocoding_longitude	decimal		<input type="checkbox"/>	
AddressWithGeocoding_state	string		<input type="checkbox"/>	
AddressWithGeocoding_street	string		<input type="checkbox"/>	
AddressWithGeocoding_street2	string		<input type="checkbox"/>	
AddressWithGeocoding_zip	string		<input type="checkbox"/>	
Auto_Increment	decimal		<input type="checkbox"/>	
Currency	decimal		<input type="checkbox"/>	
Date_Time	datetime		<input type="checkbox"/>	
Email_email	string		<input type="checkbox"/>	
Email_label	string		<input type="checkbox"/>	
Equation	decimal		<input type="checkbox"/>	
File_filename	string		<input type="checkbox"/>	
File_s3	boolean		<input type="checkbox"/>	
File_size	int32		<input type="checkbox"/>	
File_thumb_url	string		<input type="checkbox"/>	
File_type	string		<input type="checkbox"/>	
File_url	string		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
IntlAddress_city	string		<input type="checkbox"/>	
IntlAddress_state	string		<input type="checkbox"/>	
IntlAddress_street	string		<input type="checkbox"/>	
IntlAddress_street2	string		<input type="checkbox"/>	
IntlAddress_zip	string		<input type="checkbox"/>	
IntlAddressLatLng_latitude	decimal		<input type="checkbox"/>	
IntlAddressLatLng_longitude	decimal		<input type="checkbox"/>	
IntlAddressWithCountry_city	string		<input type="checkbox"/>	
IntlAddressWithCountry_country	string		<input type="checkbox"/>	
IntlAddressWithCountry_state	string		<input type="checkbox"/>	
IntlAddressWithCountry_street	string		<input type="checkbox"/>	
IntlAddressWithCountry_street2	string		<input type="checkbox"/>	
IntlAddressWithCountry_zip	string		<input type="checkbox"/>	
Link_label	string		<input type="checkbox"/>	
Link_url	string		<input type="checkbox"/>	
Multiple_Choice	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
Name_firstname	string		<input type="checkbox"/>	
Name_lastname	string		<input type="checkbox"/>	
Name_middlename	string		<input type="checkbox"/>	
Name_title	string		<input type="checkbox"/>	
Number	decimal		<input type="checkbox"/>	
Paragraph_Text	string		<input type="checkbox"/>	
Phone	string		<input type="checkbox"/>	
Rating	decimal		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Rich_Text	string		<input type="checkbox"/>	
Signature_base30	string		<input type="checkbox"/>	
Signature_svg	string		<input type="checkbox"/>	
Test_Name	string		<input type="checkbox"/>	
Text_Formula	string		<input type="checkbox"/>	
Timer_from_min	datetime		<input type="checkbox"/>	
Timer_stop_count	int32		<input type="checkbox"/>	
Timer_to_max	datetime		<input type="checkbox"/>	
Timer_total_time	int64		<input type="checkbox"/>	
USAddress_city	string		<input type="checkbox"/>	
USAddress_state	string		<input type="checkbox"/>	
USAddress_street	string		<input type="checkbox"/>	
USAddress_street2	string		<input type="checkbox"/>	
USAddress_zip	string		<input type="checkbox"/>	
Yes_No	boolean		<input type="checkbox"/>	

7.1.15 TestcaseIUD: Knack TestcaseIUDs

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: TestcaseIUDs

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

Select Knack API URL: /objects/object_11/records

Insert Knack API URL: /objects/object_11/records

Update Knack API URL: /objects/object_11/records

Delete Knack API URL: /objects/object_11/records

Field Selection Method: NotRequired

Base Path: records [*]

Table Columns

The columns of the table `TestcaseIUD` 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 or update.

Name	Data Type	Label	Required	Documentation
File_filename	string		<input type="checkbox"/>	
File_s3	boolean		<input type="checkbox"/>	
File_size	int32		<input type="checkbox"/>	
File_thumb_url	string		<input type="checkbox"/>	
File_type	string		<input type="checkbox"/>	
File_url	string		<input type="checkbox"/>	
ID	decimal		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
Name	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.16 TestcaseIUDExpanded: Knack TestcaseIUDs

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: TestcaseIUDs

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

Select Knack API URL: `/objects/object_11/records`

Insert Knack API URL: `/objects/object_11/records`

Update Knack API URL: `/objects/object_11/records`

Delete Knack API URL: `/objects/object_11/records`

Field Selection Method: NotRequired

Base Path: `records[*]`

Table Columns

The columns of the table `TestcaseIUDExpanded` 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 or update.

Name	Data Type	Label	Required	Documentation
File_data	byte[]		<input type="checkbox"/>	
File_filename	string		<input type="checkbox"/>	
File_s3	boolean		<input type="checkbox"/>	
File_size	int32		<input type="checkbox"/>	
File_thumb_url	string		<input type="checkbox"/>	
File_type	string		<input type="checkbox"/>	
File_url	string		<input type="checkbox"/>	
ID	decimal		<input type="checkbox"/>	
Image_data	byte[]		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
Name	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

7.1.17 TestExpanded: Knack Tests

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: Tests

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

Select Knack API URL: /objects/object_4/records

Insert Knack API URL: /objects/object_4/records

Update Knack API URL: /objects/object_4/records

Delete Knack API URL: /objects/object_4/records

Field Selection Method: NotRequired

Base Path: records[*]

Table Columns

The columns of the table TestExpanded 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 or update.

Name	Data Type	Label	Required	Documentation
AddressWithGeocoding_city	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
AddressWithGeocoding_latitude	decimal		<input type="checkbox"/>	
AddressWithGeocoding_longitude	decimal		<input type="checkbox"/>	
AddressWithGeocoding_state	string		<input type="checkbox"/>	
AddressWithGeocoding_street	string		<input type="checkbox"/>	
AddressWithGeocoding_street2	string		<input type="checkbox"/>	
AddressWithGeocoding_zip	string		<input type="checkbox"/>	
Auto_Increment	decimal		<input type="checkbox"/>	
Currency	decimal		<input type="checkbox"/>	
Date_Time	datetime		<input type="checkbox"/>	
Email_email	string		<input type="checkbox"/>	
Email_label	string		<input type="checkbox"/>	
Equation	decimal		<input type="checkbox"/>	
File_data	byte[]		<input type="checkbox"/>	
File_filename	string		<input type="checkbox"/>	
File_s3	boolean		<input type="checkbox"/>	
File_size	int32		<input type="checkbox"/>	
File_thumb_url	string		<input type="checkbox"/>	
File_type	string		<input type="checkbox"/>	
File_url	string		<input type="checkbox"/>	
Image_data	byte[]		<input type="checkbox"/>	
Image_filename	string		<input type="checkbox"/>	
Image_s3	boolean		<input type="checkbox"/>	
Image_size	int32		<input type="checkbox"/>	
Image_thumb_url	string		<input type="checkbox"/>	
Image_type	string		<input type="checkbox"/>	
Image_url	string		<input type="checkbox"/>	
IntlAddress_city	string		<input type="checkbox"/>	
IntlAddress_state	string		<input type="checkbox"/>	
IntlAddress_street	string		<input type="checkbox"/>	
IntlAddress_street2	string		<input type="checkbox"/>	
IntlAddress_zip	string		<input type="checkbox"/>	
IntlAddressLatLng_latitude	decimal		<input type="checkbox"/>	
IntlAddressLatLng_longitude	decimal		<input type="checkbox"/>	
IntlAddressWithCountry_city	string		<input type="checkbox"/>	
IntlAddressWithCountry_country	string		<input type="checkbox"/>	
IntlAddressWithCountry_state	string		<input type="checkbox"/>	
IntlAddressWithCountry_street	string		<input type="checkbox"/>	
IntlAddressWithCountry_street2	string		<input type="checkbox"/>	
IntlAddressWithCountry_zip	string		<input type="checkbox"/>	
Link_label	string		<input type="checkbox"/>	
Link_url	string		<input type="checkbox"/>	

Name	Data Type	Label	Required	Documentation
Multiple_Choice	string		<input type="checkbox"/>	
Name_firstname	string		<input type="checkbox"/>	
Name_lastname	string		<input type="checkbox"/>	
Name_middlename	string		<input type="checkbox"/>	
Name_title	string		<input type="checkbox"/>	
Number	decimal		<input type="checkbox"/>	
Paragraph_Text	string		<input type="checkbox"/>	
Phone	string		<input type="checkbox"/>	
Rating	decimal		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
Rich_Text	string		<input type="checkbox"/>	
Signature_base30	string		<input type="checkbox"/>	
Signature_svg	string		<input type="checkbox"/>	
Test_Name	string		<input type="checkbox"/>	
Text_Formula	string		<input type="checkbox"/>	
Timer_from_min	datetime		<input type="checkbox"/>	
Timer_stop_count	int32		<input type="checkbox"/>	
Timer_to_max	datetime		<input type="checkbox"/>	
Timer_total_time	int64		<input type="checkbox"/>	
USAddress_city	string		<input type="checkbox"/>	
USAddress_state	string		<input type="checkbox"/>	
USAddress_street	string		<input type="checkbox"/>	
USAddress_street2	string		<input type="checkbox"/>	
USAddress_zip	string		<input type="checkbox"/>	
Yes_No	boolean		<input type="checkbox"/>	

7.1.18 testpho: Knack testphos

Catalog: Knack

Schema: Objects

Primary Keys: record_id

Label: testphos

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

Select Knack API URL: /objects/object_12/records

Insert Knack API URL: /objects/object_12/records

Update Knack API URL: /objects/object_12/records

Delete Knack API URL: /objects/object_12/records

Field Selection Method: NotRequired

Base Path: `records[*]`

Table Columns

The columns of the table `testpho` 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 or update.

Name	Data Type	Label	Required	Documentation
datefield	datetime		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	
text	string		<input type="checkbox"/>	

8 Schema: Product_Details

8.1 Tables

8.1.1 Product_Details

Catalog: Knack

Schema: Product_Details

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

Select Knack API URL: `/pages/scene_6/views/view_9/records/{id}`

Insert Knack API URL: `/pages/scene_6/views/view_9/records/{id}`

Update Knack API URL: `/pages/scene_6/views/view_9/records/{id}`

Delete Knack API URL: `/pages/scene_6/views/view_9/records/{id}`

Field Selection Method: NotRequired

Parameters of Table Function

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

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

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
DUMMY	string		<input checked="" type="checkbox"/>	Default column added since the specification specifies that no data is returned.
record_id	string		<input checked="" type="checkbox"/>	

9 Schema: ProductsMenu

9.1 Tables

9.1.1 Products: Knack Table

Catalog: Knack

Schema: ProductsMenu

Label: Table

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

Select Knack API URL: `/pages/scene_5/views/view_7/records`

Insert Knack API URL: `/pages/scene_5/views/view_7/records`

Update Knack API URL: `/pages/scene_5/views/view_7/records`

Delete Knack API URL: `/pages/scene_5/views/view_7/records`

Field Selection Method: NotRequired

Base Path: `records[*]`

View Columns

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

Name	Data Type	Label	Required	Documentation
Available	string		<input type="checkbox"/>	
Description	string		<input type="checkbox"/>	
Image	string		<input type="checkbox"/>	
Price	string		<input type="checkbox"/>	
Product_ID	string		<input type="checkbox"/>	
Products_Name	string		<input type="checkbox"/>	
record_id	string		<input checked="" type="checkbox"/>	

Index

- A -

a 44
 A B Cs 38
 A_B_C 38
 A_B_C_Name 38
 Account_id 47
 Account_identifier 47
 Accounts 39
 action 32, 33
 add-odata-mandatory-filters 2
 Address_city 40, 41, 42, 43
 Address_state 40, 41, 42, 43
 Address_street 40, 41, 42, 43
 Address_street2 40, 41, 42, 43
 Address_zip 40, 41, 42, 43
 AddressWithGeocoding_city 49, 53
 AddressWithGeocoding_latitude 49, 53
 AddressWithGeocoding_longitude 49, 53
 AddressWithGeocoding_state 49, 53
 AddressWithGeocoding_street 49, 53
 AddressWithGeocoding_street2 49, 53
 AddressWithGeocoding_zip 49, 53
 alert 32, 33
 allow_exporting 32, 33
 allow_preset_filters 32, 33
 Amount 47
 analysis-enforce-row-uniqueness 2
 api_limit_extra_quantity 14, 16
 api-key 2
 api-url 2
 Application by ID 18
 Application ID 14, 16, 20, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33
 Application Settings 23
 application_id 14, 16, 20, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33
 application_name 14, 16, 20, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33
 ApplicationAccounts 14
 ApplicationAccountsByApplicationId 16
 ApplicationById 18
 ApplicationDesigns 20
 ApplicationDesignsByApplicationId 20
 application-id 2
 Applications 22
 ApplicationSettings 23

ApplicationSettingsByApplicationId 24
 authenticated 30, 31
 Auto_Increment 49, 53
 Available 47, 48, 57

- B -

babu_preseed_status 23, 24
 beta_deadline 14, 16
 billing_custom_info 14, 16
 billing_customer_id 14, 16
 billing_email 14, 16
 billing_has_paid 14, 16
 billing_ignore_capacity_until 14, 16
 billing_ignore_delinquency_until 14, 16
 billing_last_freeze_date 14, 16
 billing_notice_count 14, 16
 billing_notice_date 14, 16
 billing_status 14, 16
 Birthday 42, 43
 BLOB Preferred 37
 BLOB_PREFERRED 37
 BOL_RESPONSE_CACHE_MAX_AGE_SEC 37
 bulk-delete-page-size-rows 2
 bulk-insert-page-size-bytes 2
 bulk-insert-page-size-rows 2
 button_bg_color 20
 button_border_color 20
 button_hover_color 20
 button_text_color 20
 button_textshadow_style 20

- C -

Charge 47
 cluster 23, 24
 Companies 40, 41
 Companies_Name 40, 41
 CompaniesExpanded 41
 Company_id 44
 Company_identifier 44
 conditional 26, 27
 Contacts 42, 43
 Contacts_Name 42, 43
 ContactsExpanded 43
 Content Type 37
 CONTENT_TYPE 37
 counts_builders 14, 16
 css 23, 24
 Currency 49, 53

Customer 46

- D -

Date 47

DATE_ENDED 37

DATE_STARTED 37

Date_Time 49, 53

datefield 55

DateTest 44

DateTests 44

default 26, 27

display_pagination_below 32, 33

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-multiplicator 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-multiplicator 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-429-too-many-requests-max-tries 2

download-error-429-too-many-requests-sleep-initial-ms 2

download-error-429-too-many-requests-sleep-max-ms 2

download-error-429-too-many-requests-sleep-multiplicator 2

download-error-502-server-unavailable-max-tries 2

download-error-502-server-unavailable-sleep-initial-ms 2

download-error-502-server-unavailable-sleep-max-ms 2

download-error-502-server-unavailable-sleep-multiplicator 2

download-error-503-server-unavailable-max-tries 2

download-error-503-server-unavailable-sleep-initial-ms 2

download-error-503-server-unavailable-sleep-max-ms 2

download-error-503-server-unavailable-sleep-multiplicator 2

download-error-504-gateway-timeout-max-tries 2

download-error-504-gateway-timeout-sleep-initial-ms 2

download-error-504-gateway-timeout-sleep-max-ms 2

download-error-504-gateway-timeout-sleep-multiplicator 2

download-error-590-network-connect-timeout-max-tries 2

download-error-590-network-connect-timeout-sleep-initial-ms 2

download-error-590-network-connect-timeout-sleep-max-ms 2

download-error-590-network-connect-timeout-sleep-multiplicator 2

download-error-599-network-connect-timeout-max-tries 2

download-error-599-network-connect-timeout-sleep-initial-ms 2

download-error-599-network-connect-timeout-sleep-max-ms 2

download-error-599-network-connect-timeout-sleep-multiplicator 2

download-error-argument-exception-max-tries 2

download-error-argument-exception-sleep-initial-ms 2

download-error-argument-exception-sleep-max-ms 2

download-error-argument-exception-sleep-multiplicator 2

download-error-internet-down-max-tries 2

download-error-internet-down-sleep-initial-ms 2

download-error-internet-down-sleep-max-ms 2

download-error-internet-down-sleep-multiplicator 2

download-error-io-exception-max-tries 2

download-error-io-exception-sleep-initial-ms 2

download-error-io-exception-sleep-max-ms 2

download-error-io-exception-sleep-multiplicator 2

download-error-json-exception-max-tries 2

download-error-json-exception-sleep-initial-ms 2

download-error-json-exception-sleep-max-ms 2

download-error-json-exception-sleep-multiplicator 2

download-error-other-exception-max-tries 2

download-error-other-exception-sleep-initial-ms 2

download-error-other-exception-sleep-max-ms 2

download-error-other-exception-sleep-multiplicator 2

download-error-socket-exception-max-tries 2

download-error-socket-exception-sleep-initial-ms 2

download-error-socket-exception-sleep-max-ms 2

download-error-socket-exception-sleep-multiplicator 2

download-error-web-exception-max-tries 2

download-error-web-exception-sleep-initial-ms 2

download-error-web-exception-sleep-max-ms 2

download-error-web-exception-sleep-multiplicator 2

download-error-web-not-implemented-max-tries 2

download-error-web-not-implemented-sleep-initial-ms 2

download-error-web-not-implemented-sleep-max-ms 2

download-error-web-not-implemented-sleep-multiplicat
or 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

Driver 1

DRY_RUN 37

DUMMY 35, 56

Duration (ms) 37

DURATION_MS 37

- E -

ecommerce_enabled 18, 22

Email_email 39, 42, 43, 49, 53

Email_label 39, 42, 43, 49, 53

End Date 37

Equation 49, 53

Error Message Code 37

Error Message Text 37

ERROR_MESSAGE_CODE 37

ERROR_MESSAGE_TEXT 37

- F -

Fail on Error 37

FAIL_ON_ERROR 37

File_data 52, 53

File_filename 49, 51, 52, 53

File_s3 49, 51, 52, 53

File_size 49, 51, 52, 53

File_thumb_url 49, 51, 52, 53

File_type 49, 51, 52, 53

File_url 49, 51, 52, 53

filter_type 32, 33

force-case-sensitive-identifiers 2

forced-casing-identifiers 2

format_format 26, 27

format_label 26, 27

from_email 23, 24

- G -

Gender 42, 43

geo 23, 24

- H -

has_new_object_types 23, 24

header_bg_color 20

header_text_color 20

header_textshadow_style 20

HERE_appCode 18, 22

HERE_appld 18, 22

home_scene_key 18, 22

home_scene_slug 18, 22

HTTP Disk Cache Maximum Age (sec) 37

HTTP Memory Cache Maximum Age (sec) 37

HTTP Method 37

HTTP Status Code 37

HTTP_DISK_CACHE_MAX_AGE_SEC 37

HTTP_DISK_CACHE_SAVE 37

HTTP_DISK_CACHE_USE 37

HTTP_MEMORY_CACHE_MAX_AGE_SEC 37

HTTP_MEMORY_CACHE_SAVE 37

HTTP_MEMORY_CACHE_USE 37

HTTP_METHOD 37

HTTP_STATUS_CODE 37

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

https_redirect 23, 24

- I -

icons 23, 24

id 16, 18, 20, 24, 27, 29, 31, 33, 35, 56

identifier 28, 29

ignore_reload 32, 33

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
 Image 57
 Image_data 43, 48, 52, 53
 Image_filename 42, 43, 47, 48, 49, 51, 52, 53
 Image_s3 42, 43, 47, 48, 49, 51, 52, 53
 Image_size 42, 43, 47, 48, 49, 51, 52, 53
 Image_thumb_url 42, 43, 47, 48, 49, 51, 52, 53
 Image_type 42, 43, 47, 48, 49, 51, 52, 53
 Image_url 42, 43, 47, 48, 49, 51, 52, 53
 immutable 26, 27
 inactivity_message 23, 24
 inactivity_timeout 23, 24
 inactivity_timeout_enabled 23, 24
 Industry 40, 41
 inflections_plural 28, 29
 inflections_singular 28, 29
 IntlAddress_city 49, 53
 IntlAddress_state 49, 53
 IntlAddress_street 49, 53
 IntlAddress_street2 49, 53
 IntlAddress_zip 49, 53
 IntlAddressLatLng_latitude 49, 53
 IntlAddressLatLng_longitude 49, 53
 IntlAddressWithCountry_city 49, 53
 IntlAddressWithCountry_country 49, 53
 IntlAddressWithCountry_state 49, 53
 IntlAddressWithCountry_street 49, 53
 IntlAddressWithCountry_street2 49, 53
 IntlAddressWithCountry_zip 49, 53
 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-multiplier 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-multiplier 2
 invative-sql-compress-sparse-arrays 2
 invative-sql-correct-invalid-date 2
 invative-sql-forward-filters-to-data-containers 2
 invative-sql-share-byte-arrays 2
 invative-sql-share-strings 2
 invative-sql-shuffle-fetch-results-data-containers
 invative-use-cache 2
 IP_Address 47
 ip_whitelist 23, 24

- J -

javascript 23, 24
 join-set-points-per-request 2
 Journal_Details 35
 Journals 36, 44
 JournalsMultiple_Choice 45

- K -

key 28, 29, 30, 31, 32, 33
 keyword_search 32, 33
 Knack 1, 14, 16, 18, 20, 22, 23, 24, 26, 27, 28, 29,
 30, 31, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
 45, 46, 47, 48, 49, 51, 52, 53, 55, 56, 57

- L -

label 32, 33
 language 23, 24
 last_update 23, 24
 layout 32, 33
 layout_app_fullwidth 18, 22
 layout_app_menu_auth 18, 22
 layout_entry_scene_menu 18, 22
 layout_page_menu_style 18, 22
 layout_theme 18, 22
 legacy_entry_history_nuked 23, 24
 limit_profile_access 30, 31, 32, 33
 limit-partition-calls-left 2
 Link_label 49, 53
 link_text_color 20
 Link_url 49, 53
 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
 logo 23, 24
 Logo_data 41
 Logo_filename 40, 41
 Logo_s3 40, 41
 Logo_size 40, 41
 Logo_thumb_url 40, 41
 Logo_type 40, 41
 Logo_url 18, 22, 40, 41

- M -

maximum-length-identifiers 2
 max-odata-filters 2
 max-url-length-accepted 2
 max-url-length-desired 2
 menu 30, 31
 metadata-cache-max-age-sec 2
 modal 30, 31
 mongo 23, 24
 Multiple choices for Journals 45
 Multiple_Choice 44, 49, 53

- N -

n 44
 Name 14, 16, 18, 22, 28, 29, 51, 52
 Name_firstname 39, 44, 49, 53
 Name_lastname 39, 44, 49, 53
 Name_middlename 39, 44, 49, 53
 Name_title 39, 44, 49, 53
 Native Platform Scalar Requests 37
 NATIVEPLATFORMSCALARREQUESTS 37
 new_count 23, 24
 notification_bg_color 20
 notification_text_color 20
 npt 37
 Number 49, 53

- O -

oauth-unauthorized-max-tries 2
 oauth-unauthorized-sleep-initial-ms 2
 oauth-unauthorized-sleep-max-ms 2
 oauth-unauthorized-sleep-multiplicator 2
 object_id 26, 27
 object_key 26, 27, 30, 31
 object_name 26, 27
 ObjectFields 26
 ObjectFieldsByApplicationId 27
 Objects 28
 ObjectsByApplicationId 29
 Occupation 42, 43
 Order_Date 46
 Order_ID 46
 Orders_Name 46
 OrdersObject 46
 OrdersObjects 46
 ORIG_SYSTEM_GROUP 37

ORIG_SYSTEM_REFERENCE 37
 Original System Group 37
 Original System Reference 37

- P -

Paragraph_Text 49, 53
 parent 30, 31, 32, 33
 partition-slot-based-rate-limit-length-ms 2
 partition-slot-based-rate-limit-slots 2
 Password 39
 Payload 37
 PAYLOAD_TEXT 37
 Payments 47
 Phone 40, 41, 42, 43, 49, 53
 plan_limits_api_limit 14, 16
 plan_limits_applications 14, 16
 plan_limits_records 14, 16
 plan_limits_storage 14, 16
 powered_by_link 23, 24
 pre-request-delay-ms 2
 Price 47, 48, 57
 print 30, 31
 Processor 47
 Product_Details 56
 Product_ID 47, 48, 57
 Product_Name 46
 product_plan_api_limit 14, 16
 product_plan_apps 14, 16
 product_plan_babu_unit 14, 16
 product_plan_babu_value 14, 16
 product_plan_builders 14, 16
 product_plan_id 14, 16
 product_plan_level 14, 16
 product_plan_name 14, 16
 product_plan_price 14, 16
 product_plan_records 14, 16
 product_plan_storage 14, 16
 product_plan_subscription_id 14, 16
 product_plan_type 14, 16
 Products 47, 48, 57
 Products_Name 57
 Products_Name_id 47, 48
 Products_Name_identifier 47, 48
 ProductsExpanded 48

- R -

Rating 49, 53
 record_id 35, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46,
 47, 48, 49, 51, 52, 53, 55, 56, 57

referral 14, 16
 registration_migration_status 23, 24
 requested-page-size 2
 requests-parallel-max 2
 required 26, 27
 Response Cache Maximum Age (sec) 37
 Result BLOB 37
 Result Text 37
 RESULT_BLOB 37
 RESULT_DATE_TIME_UTC 37
 RESULT_NUMBER 37
 RESULT_TEXT 37
 Rich_Text 49, 53
 rows_per_page 32, 33
 Run without Actions 37
 runner 23, 24

- S -

s3_bucket 18, 22
 s3_domain 18, 22
 s3_secure_bucket 18, 22
 s3_secure_domain 18, 22
 Save HTTP Disk Cache 37
 Save HTTP Memory Cache 37
 scene_id 32, 33
 scene_key 32, 33
 scene_name 32, 33
 Scenes 30
 ScenesByApplicationId 31
 SceneViews 32
 SceneViewsByApplicationId 33
 script_protection_enabled 23, 24
 settings_agreements 14, 16
 settings_force_region_for_new_apps_enabled 14, 16
 settings_force_region_for_new_apps_region 14, 16
 settings_has_open_beta_access 14, 16
 settings_has_opted_into_open_beta 14, 16
 settings_hipaa_enabled 14, 16
 settings_hipaa_region 14, 16
 settings_mongo 14, 16
 settings_new_apps_are_sql 14, 16
 settings_rate_limit_login_requests 14, 16
 settings_security_builder_inactivity_message 14, 16
 settings_security_builder_inactivity_timeout 14, 16
 settings_security_builder_inactivity_timeout_enabled 14, 16
 Ship_Date 46
 Ship_Type 46

Shipping_Address_city 46
 Shipping_Address_state 46
 Shipping_Address_street 46
 Shipping_Address_street2 46
 Shipping_Address_zip 46
 should_purge_record_history 23, 24
 Signature_base30 49, 53
 Signature_svg 49, 53
 simulate-http-400-errors 2
 simulate-http-400-errors-percentage 2
 simulate-http-401-errors 2
 simulate-http-401-errors-percentage 2
 simulate-http-403-errors 2
 simulate-http-403-errors-percentage 2
 simulate-http-408-errors 2
 simulate-http-408-errors-percentage 2
 simulate-http-429-errors 2
 simulate-http-429-errors-percentage 2
 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-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
 slug 14, 16, 18, 22, 30, 31
 solr 23, 24
 sort_field 28, 29
 sort_order 28, 29
 Source 47
 sso 14, 16
 standardize-identifiers 2
 standardize-identifiers-casing 2
 Start Date 37
 Status 14, 16, 18, 22, 28, 29, 46, 47
 subdomain 18, 22
 Successful 37
 SUCCESSFUL 37
 support_access 23, 24

- T -

Table 36, 57
 technical_contact 23, 24
 Test 49

Test_Name 49, 53
 TestcaseUD 51
 TestcaseUDEExpanded 52
 TestcaseUDs 51, 52
 TestExpanded 53
 testpho 55
 testphos 55
 Tests 49, 53
 text 55
 Text_Formula 44, 49, 53
 Timeout (sec) 37
 TIMEOUT_SEC 37
 Timer_from_min 49, 53
 Timer_stop_count 49, 53
 Timer_to_max 49, 53
 Timer_total_time 49, 53
 timezone 23, 24
 timezone_dst 23, 24
 timezone_offset 23, 24
 title 32, 33
 Total 46
 Transaction ID 37
 TRANSACTION_ID 37
 type 26, 27, 28, 29, 30, 31, 32, 33

- U -

unique 26, 27
 URL 37
 USAddress_city 49, 53
 USAddress_state 49, 53
 USAddress_street 49, 53
 USAddress_street2 49, 53
 USAddress_zip 49, 53
 Use HTTP Disk Cache 37
 Use HTTP Memory Cache 37
 use_allowed_html_whitelist 23, 24
 use_multiple_api_subdomains 23, 24
 use-batch-insert 2
 use-http-disk-cache-read 2
 use-http-disk-cache-write 2
 use-http-memory-cache-read 2
 use-http-memory-cache-write 2
 user 26, 27, 28, 29
 user_id 14, 16
 User_Roles 39
 User_Status 39
 users_enabled 18, 22
 users_registration 18, 22
 users_scope 18, 22

- V -

v3_beta 23, 24
 v3_open_beta 23, 24
 value 45

- W -

Website_label 40, 41
 Website_url 40, 41

- Y -

Yes_No 49, 53



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