23.0

Knack API Data Model

for use with Invantive SQL

invantive
the SQL company

Copyright

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 an 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

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 paradigma). 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.

AuthenticationAuthentication 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 LimitsInvantive 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 Drivers 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 Con-nec-tion String	Set from Set SQL-State-ment	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 re-ports unprocessable entity during retrieval of data.		2	✓	✓	✓	
dow nload-error-429-too-many-re-quests-max-tries	Maximum number of tries w hen the w ebsite reports that too many re-quests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
dow nload-error-429-too-many-re-quests-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-re-quests-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-re-quests-sleep-multi-plicator	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-unavail-able-max-tries	Maximum number of tries w hen OData server reports a bad gate-w ay 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 Con-nec-tion String	Set from Set SQL-State-ment	Set from Driver s File	Set from Log On
able-sleep-initial-ms	bad gatew ay during retrieval of data.						
dow nload-error-502-server-unavail-able-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen OData server reports that a bad gatew ay during retrieval of data.		300000	✓	✓	✓	
dow nload-error-502-server-unavail-able-sleep-multiplic-ator	Multiplication factor for sleep betw een retries OData server re-ports a bad gatew ay during retrieval of data.		2	✓	✓	✓	
dow nload-error-503-server-unavail-able-max-tries	Maximum number of tries w hen OData server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
dow nload-error-503-server-unavail-able-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen OData server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
dow nload-error-503-server-unavail-able-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen OData server reports that the API server is un-available during retrieval of data.		300000	✓	✓	✓	
dow nload-error-503-server-unavail-able-sleep-multiplic-ator	Multiplication factor for sleep betw een retries OData server re-ports that the API server is unavail-able during retrieval of data.		2	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a gatew ay timeout.		10	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a gatew ay timeout.		10000	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a gatew ay timeout.		300000	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-multi-plicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a gatew ay timeout.		2	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a HTTP status 590.		10	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 590.		10000	✓	✓	✓	
dow nload-error-590-netw ork-con-	Maximum sleep in milliseconds betw een retries w hen the w ebsite		300000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Con-nec-tion String	Set from Set SQL-State-ment	Set from Driver s File	Set from Log On
nect-timeout-sleep-max-ms	reports a HTTP status 590.						
dow nload-error-590-netw ork-con-nect-timeout-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a HTTP status 590.		2	✓	✓	✓	
dow nload-error-599-netw ork-con-nect-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a HTTP status 599.		10	✓	✓	✓	
dow nload-error-599-netw ork-con-nect-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 599.		10000	✓	✓	✓	
dow nload-error-599-netw ork-con-nect-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 599.		300000	✓	✓	✓	
dow nload-error-599-netw ork-con-nect-timeout-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a HTTP status 599.		2	✓	✓	✓	
dow nload-error-ar-gument-exception-max-tries	Maximum number of tries w hen an argument exception is returned w hen dow nloading a blob.		10	✓	✓	✓	
dow nload-error-ar-gument-exception-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen an argument exception is returned w hen dow nloading a blob.		10000	✓	✓	✓	
dow nload-error-ar-gument-exception-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen an argument exception is returned w hen dow n-loading a blob.		300000	✓	✓	✓	
dow nload-error-ar-gument-exception-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen an argument exception is returned w hen dow n-loading a blob.		2	✓	✓	✓	
dow nload-error-in-ternet-dow n-max-tries	Maximum number of tries w hen the Internet connection seems dow n during retrieval of data.		10	✓	✓	✓	
dow nload-error-in-ternet-dow n-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the Internet connection seems dow n during retrieval of data.		10000	✓	✓	✓	
dow nload-error-in-ternet-dow n-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the Internet connection seems dow n during re-trieval of data.		300000	✓	✓	✓	
dow nload-error-in-ternet-dow n-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the Internet connection seems dow n during re-trieval 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
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 ocurrs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		300000	✓	✓	✓	
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error ocurrs during retrieval of data.		2	✓	✓	✓	
download-error-socket-exception-max-tries	Maximum number of tries when the network connection is forcible dropped during retrieval of data.		10	✓	✓	✓	
download-error-socket-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		10000	✓	✓	✓	
download-error-socket-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		300000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcible dropped during retrieval of data.		2	✓	✓	✓	
download-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-implemented-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports not implemented.		10000	✓	✓	✓	
download-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	
download-error-web-not-implemented-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports not implemented.		2	✓	✓	✓	
download-error-web-timeout-max-tries	Maximum number of tries when the connection reports a timeout.		10	✓	✓	✓	
download-error-web-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports a timeout.		1000	✓	✓	✓	
download-error-web-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports a timeout.		30000	✓	✓	✓	
download-error-web-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports a timeout.		2	✓	✓	✓	
download-error-web-unauthorized-max-tries	Maximum number of tries when the connection reports an unauthorized error.		1	✓	✓	✓	
download-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 Drivers 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\Invantive\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 en-		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 Drivers 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 Drivers 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 Con-nec-tion String	Set from Set SQL-State-ment	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 answ er the current query.	OData	True	✓	✓	✓	
use-http-memory-cache-w rite	Whether to memorize HTTP re-sponses 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 Invantive SQL driver for Knack does not cover it. In the latter case, please use the table `Nat-ivePlatformScalarRequests` 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
api_limit_extra_quantity	int64		☐	
application_id	string	Application ID	☑	
application_name	string		☑	
beta_deadline	datetime		☐	
billing_custom_info	string		☐	
billing_customer_id	string		☐	
billing_email	string		☐	
billing_has_paid	boolean		☐	

Name	Data Type	Label	Required	Documentation
billing_ignore_capacity_until	datetime		☐	
billing_ignore_delinquency_until	datetime		☐	
billing_last_freeze_date	datetime		☐	
billing_notice_count	int32		☐	
billing_notice_date	datetime		☐	
billing_status	string		☐	
counts_builders	int64		☐	
id	string		☐	
name	string		☐	
plan_limits_api_limit	int64		☐	
plan_limits_applications	int64		☐	
plan_limits_records	int64		☐	
plan_limits_storage	int64		☐	
product_plan_api_limit	int64		☐	
product_plan_apps	int64		☐	
product_plan_babu_unit	string		☐	
product_plan_babu_value	int64		☐	
product_plan_builders	int64		☐	
product_plan_id	string		☐	
product_plan_level	int32		☐	
product_plan_name	string		☐	
product_plan_price	decimal		☐	
product_plan_records	int64		☐	
product_plan_storage	string		☐	
product_plan_subscription_id	string		☐	
product_plan_type	string		☐	
referral	string		☐	
settings_agreements	string		☐	
settings_force_region_-for_new_apps_enabled	boolean		☐	
settings_force_region_-for_new_apps_region	string		☐	
settings_has_open_beta_access	boolean		☐	
settings_has_opted_in-to_open_beta	boolean		☐	
settings_hipaa_enabled	boolean		☐	
settings_hipaa_region	string		☐	
settings_mongo	string		☐	
settings_new_apps_are_sql	boolean		☐	
settings_rate_limit_login_requests	string		☐	
settings_security_builder_inactiv-ity_message	string		☐	

Name	Data Type	Label	Required	Documentation
settings_security_builder_inactivity_timeout_enabled	boolean		☐	
settings_security_builder_inactivity_timeout	int64		☐	
slug	string		☐	
sso	boolean		☐	
status	string		☐	
user_id	string		☐	

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 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`

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	☑		

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		☐	
application_id	string	Application ID	☑	
application_name	string		☑	
beta_deadline	datetime		☐	
billing_custom_info	string		☐	
billing_customer_id	string		☐	
billing_email	string		☐	
billing_has_paid	boolean		☐	
billing_ignore_capacity_until	datetime		☐	
billing_ignore_delinquency_until	datetime		☐	
billing_last_freeze_date	datetime		☐	
billing_notice_count	int32		☐	
billing_notice_date	datetime		☐	
billing_status	string		☐	
counts_builders	int64		☐	
id	string		☐	
name	string		☐	
plan_limits_api_limit	int64		☐	
plan_limits_applications	int64		☐	
plan_limits_records	int64		☐	
plan_limits_storage	int64		☐	
product_plan_api_limit	int64		☐	
product_plan_apps	int64		☐	
product_plan_babu_unit	string		☐	
product_plan_babu_value	int64		☐	
product_plan_builders	int64		☐	
product_plan_id	string		☐	
product_plan_level	int32		☐	
product_plan_name	string		☐	
product_plan_price	decimal		☐	
product_plan_records	int64		☐	
product_plan_storage	string		☐	
product_plan_subscription_id	string		☐	
product_plan_type	string		☐	
referral	string		☐	
settings_agreements	string		☐	

Name	Data Type	Label	Required	Documentation
settings_force_region_-for_new _apps_enabled	boolean		☐	
settings_force_region_-for_new _apps_region	string		☐	
settings_has_open_beta_access	boolean		☐	
settings_has_opted_in-to_open_beta	boolean		☐	
settings_hipaa_enabled	boolean		☐	
settings_hipaa_region	string		☐	
settings_mongo	string		☐	
settings_new _apps_are_sql	boolean		☐	
settings_rate_limit_login_requests	string		☐	
settings_security_builder_inactiv-ity_message	string		☐	
settings_security_builder_inactiv-ity_timeout_enabled	boolean		☐	
settings_security_builder_inactiv-ity_timeout	int64		☐	
slug	string		☐	
sso	boolean		☐	
status	string		☐	
user_id	string		☐	

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 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`

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	☑		

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		☐	
HERE_appCode	string		☐	
HERE_appId	string		☐	
home_scene_key	string		☐	
home_scene_slug	string		☐	
id	string		☑	Unique ID.
layout_app_fullw idth	boolean		☐	
layout_app_menu_auth	boolean		☐	
layout_entry_scene_menu	boolean		☐	
layout_page_menu_style	string		☐	
layout_theme	string		☐	
logo_url	string		☐	
name	string		☑	
s3_bucket	string		☐	
s3_domain	string		☐	
s3_secure_bucket	string		☐	
s3_secure_domain	string		☐	
slug	string		☐	
status	string		☐	
subdomain	string		☐	
users_enabled	boolean		☐	
users_registration	string		☐	

Name	Data Type	Label	Required	Documentation
users_scope	string		☐	

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 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/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	☑	
application_name	string		☑	
button_bg_color	string		☐	
button_border_color	string		☐	
button_hover_color	string		☐	
button_text_color	string		☐	
button_textshadow _style	string		☐	
header_bg_color	string		☐	
header_text_color	string		☐	
header_textshadow _style	string		☐	
link_text_color	string		☐	
notification_bg_color	string		☐	
notification_text_color	string		☐	

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 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`

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	☑		

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	☑	
application_name	string		☑	
button_bg_color	string		☐	
button_border_color	string		☐	
button_hover_color	string		☐	
button_text_color	string		☐	
button_textshadow _style	string		☐	
header_bg_color	string		☐	

Name	Data Type	Label	Required	Documentation
header_text_color	string		☐	
header_textshadow _style	string		☐	
link_text_color	string		☐	
notification_bg_color	string		☐	
notification_text_color	string		☐	

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 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/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		☐	
HERE_appCode	string		☐	
HERE_appId	string		☐	
home_scene_key	string		☐	
home_scene_slug	string		☐	
id	string		☑	Unique ID.
layout_app_fullw idth	boolean		☐	
layout_app_menu_auth	boolean		☐	
layout_entry_scene_menu	boolean		☐	
layout_page_menu_style	string		☐	
layout_theme	string		☐	
logo_url	string		☐	
name	string		☑	

Name	Data Type	Label	Required	Documentation
s3_bucket	string		☐	
s3_domain	string		☐	
s3_secure_bucket	string		☐	
s3_secure_domain	string		☐	
slug	string		☐	
status	string		☐	
subdomain	string		☐	
users_enabled	boolean		☐	
users_registration	string		☐	
users_scope	string		☐	

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 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/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	☑	
application_name	string		☑	
babu_preseed_status	string		☐	
cluster	string		☐	
css	string		☐	
from_email	string		☐	
geo	boolean		☐	
has_new_object_types	boolean		☐	

Name	Data Type	Label	Required	Documentation
https_redirect	boolean		☐	
icons	boolean		☐	
inactivity_message	string		☐	
inactivity_timeout_enabled	boolean		☐	
inactivity_timeout	int64		☐	
ip_whitelist	string		☐	
javascript	string		☐	
language	string		☐	
last_update	string		☐	
legacy_entry_history_nuked	string		☐	
logo	string		☐	
mongo	string		☐	
new_count	boolean		☐	
powered_by_link	boolean		☐	
registration_migration_status	string		☐	
runner	string		☐	
script_protection_enabled	boolean		☐	
should_purge_record_history	boolean		☐	
solr	string		☐	
support_access	boolean		☐	
technical_contact	string		☐	
timezone_dst	string		☐	
timezone_offset	string		☐	
timezone	string		☐	
use_allowed_html_whitelist	boolean		☐	
use_multiple_api_subdomains	boolean		☐	
v3_beta	boolean		☐	
v3_open_beta	boolean		☐	

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 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`

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	☑		

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	☑	
application_name	string		☑	
babu_preseed_status	string		☐	
cluster	string		☐	
css	string		☐	
from_email	string		☐	
geo	boolean		☐	
has_new_object_types	boolean		☐	
https_redirect	boolean		☐	
icons	boolean		☐	
inactivity_message	string		☐	
inactivity_timeout_enabled	boolean		☐	
inactivity_timeout	int64		☐	
ip_whitelist	string		☐	
javascript	string		☐	
language	string		☐	

Name	Data Type	Label	Required	Documentation
last_update	string		☐	
legacy_entry_history_nuked	string		☐	
logo	string		☐	
mongo	string		☐	
new_count	boolean		☐	
powered_by_link	boolean		☐	
registration_migration_status	string		☐	
runner	string		☐	
script_protection_enabled	boolean		☐	
should_purge_record_history	boolean		☐	
solr	string		☐	
support_access	boolean		☐	
technical_contact	string		☐	
timezone_dst	string		☐	
timezone_offset	string		☐	
timezone	string		☐	
use_allowed_html_whitelist	boolean		☐	
use_multiple_api_subdomains	boolean		☐	
v3_beta	boolean		☐	
v3_open_beta	boolean		☐	

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 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/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	☑	
application_name	string		☑	
conditional	boolean		☐	
default	string		☐	
format_format	string		☐	
format_label	string		☐	
id	string		☐	Unique ID.
immutable	boolean		☐	
object_id	string		☑	
object_key	string		☑	
object_name	string		☑	
required	boolean		☐	
type	string		☐	
unique	boolean		☐	
user	boolean		☐	

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 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[*].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	☑		

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	☑	
application_name	string		☑	
conditional	boolean		☐	
default	string		☐	
format_format	string		☐	
format_label	string		☐	
id	string		☐	Unique ID.
immutable	boolean		☐	
object_id	string		☑	
object_key	string		☑	
object_name	string		☑	
required	boolean		☐	
type	string		☐	
unique	boolean		☐	
user	boolean		☐	

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 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/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	☑	
application_name	string		☑	
id	string		☑	Unique ID.
identifier	string		☐	
inflections_plural	string		☐	
inflections_singular	string		☐	
key	string		☑	
name	string		☑	
sort_field	string		☐	
sort_order	string		☐	
status	string		☐	
type	string		☐	
user	string		☐	

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	☑		

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	☑	
application_name	string		☑	
id	string		☑	Unique ID.
identifier	string		☐	
inflections_plural	string		☐	
inflections_singular	string		☐	
key	string		☑	
name	string		☑	
sort_field	string		☐	
sort_order	string		☐	
status	string		☐	
type	string		☐	
user	string		☐	

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 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/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	☑	
application_name	string		☑	
authenticated	boolean		☐	
id	string		☑	Unique ID.
key	string		☑	
limit_profile_access	string		☐	
menu	string		☐	
modal	boolean		☐	
object_key	string		☐	
parent	string		☐	
print	boolean		☐	
slug	string		☐	
type	string		☐	

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 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.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	☑		

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	☑	
application_name	string		☑	
authenticated	boolean		☐	
id	string		☑	Unique ID.
key	string		☑	
limit_profile_access	string		☐	
menu	string		☐	
modal	boolean		☐	
object_key	string		☐	
parent	string		☐	
print	boolean		☐	
slug	string		☐	
type	string		☐	

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 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/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		☐	
alert	string		☐	
allow_exporting	boolean		☐	
allow_preset_filters	boolean		☐	
application_id	string	Application ID	☑	
application_name	string		☑	
description	string		☐	
display_pagination_below	boolean		☐	
filter_type	string		☐	
id	string		☑	Unique ID.
ignore_reload	boolean		☐	
key	string		☑	
keyword_search	boolean		☐	
label	string		☐	
layout	string		☐	
limit_profile_access	boolean		☐	
parent	string		☐	
rows_per_page	int32		☐	
scene_id	string		☑	
scene_key	string		☑	
scene_name	string		☑	
title	string		☐	
type	string		☐	

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 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.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	☑		

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		☐	
alert	string		☐	
allow_exporting	boolean		☐	
allow_preset_filters	boolean		☐	
application_id	string	Application ID	☑	
application_name	string		☑	
description	string		☐	
display_pagination_below	boolean		☐	
filter_type	string		☐	
id	string		☑	Unique ID.

Name	Data Type	Label	Required	Documentation
ignore_reload	boolean		☐	
key	string		☑	
keyword_search	boolean		☐	
label	string		☐	
layout	string		☐	
limit_profile_access	boolean		☐	
parent	string		☐	
rows_per_page	int32		☐	
scene_id	string		☑	
scene_key	string		☑	
scene_name	string		☑	
title	string		☐	
type	string		☐	

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 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: `/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	☑		

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		☑	Default column added since the specification specifies that no data is returned.
record_id	string		☑	

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 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: `/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		☐	
Description	string		☐	
record_id	string		☑	

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	☑	Indicator whether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☑	
DATE_STARTED	datetime	Start Date	☑	
DRY_RUN	boolean	Run without Actions	☑	
DURATION_MS	int32	Duration (ms)	☑	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	☐	
FAIL_ON_ERROR	boolean	Fail on Error	☑	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)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.

Name	Data Type	Label	Required	Documentation
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	☐	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	string(30)	HTTP Method	☐	
HTTP_STATUS_CODE	int16	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	☐	
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	☐	
PAYLOAD_TEXT	string	Payload	☐	
RESULT_BLOB	byte[]	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime		☐	
RESULT_NUMBER	decimal		☐	
RESULT_TEXT	string	Result Text	☐	
SUCCESSFUL	boolean	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int32	Transaction ID	☑	Incrementing ID of the transaction.
URL	string(4000)	URL	☐	

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		☐	
record_id	string		☑	

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		☐	
Email_label	string		☐	
Name_firstname	string		☐	
Name_lastname	string		☐	
Name_middlename	string		☐	
Name_title	string		☐	
Password	string		☐	
record_id	string		☑	
User_Roles	string		☐	
User_Status	string		☐	

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		☐	
Address_state	string		☐	
Address_street	string		☐	
Address_street2	string		☐	
Address_zip	string		☐	
Companies_Name	string		☐	
Description	string		☐	
Industry	string		☐	
Logo_filename	string		☐	
Logo_s3	boolean		☐	
Logo_size	int32		☐	
Logo_thumb_url	string		☐	
Logo_type	string		☐	
Logo_url	string		☐	
Notes	string		☐	
Phone	string		☐	
record_id	string		☑	
Website_label	string		☐	
Website_url	string		☐	

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		☐	
Address_state	string		☐	
Address_street	string		☐	
Address_street2	string		☐	
Address_zip	string		☐	
Companies_Name	string		☐	
Description	string		☐	
Industry	string		☐	
Logo_data	byte[]		☐	
Logo_filename	string		☐	
Logo_s3	boolean		☐	
Logo_size	int32		☐	
Logo_thumb_url	string		☐	
Logo_type	string		☐	
Logo_url	string		☐	
Notes	string		☐	
Phone	string		☐	
record_id	string		☑	
Website_label	string		☐	
Website_url	string		☐	

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		☐	
Address_state	string		☐	
Address_street	string		☐	
Address_street2	string		☐	
Address_zip	string		☐	
Birthday	datetime		☐	
Contacts_Name	string		☐	
Email_email	string		☐	
Email_label	string		☐	
Gender	string		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
Notes	string		☐	
Occupation	string		☐	
Phone	string		☐	
record_id	string		☑	

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		☐	
Address_state	string		☐	
Address_street	string		☐	
Address_street2	string		☐	
Address_zip	string		☐	
Birthday	datetime		☐	
Contacts_Name	string		☐	
Email_email	string		☐	
Email_label	string		☐	
Gender	string		☐	
Image_data	byte[]		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
Notes	string		☐	
Occupation	string		☐	
Phone	string		☐	
record_id	string		☑	

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		☐	
n	decimal		☐	
Name_firstname	string		☐	
Name_lastname	string		☐	
Name_middlename	string		☐	
Name_title	string		☐	
record_id	string		☑	
t	string		☐	

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		☐	
Company_id	string		☐	
Company_identifier	string		☐	
Description	string		☐	
Multiple_Choice	string		☐	
record_id	string		☑	
Text_Formula	string		☐	

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		☑	
value	string		☑	

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		☐	
Description	string		☐	
Order_Date	datetime		☐	
Order_ID	decimal		☐	
Orders_Name	string		☐	
Product_Name	string		☐	
record_id	string		☑	
Ship_Date	datetime		☐	
Ship_Type	string		☐	
Shipping_Address_city	string		☐	
Shipping_Address_state	string		☐	
Shipping_Address_street	string		☐	
Shipping_Address_street2	string		☐	
Shipping_Address_zip	string		☐	
Status	string		☐	
Total	decimal		☐	

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		☐	
Account_identifier	string		☐	
Amount	decimal		☐	
Charge	string		☐	
Date	datetime		☐	
Description	string		☐	
ID	string		☐	
IP_Address	string		☐	
Notes	string		☐	
Processor	string		☐	
record_id	string		☑	
Source	string		☐	
Status	string		☐	

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		☐	
Description	string		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
Price	decimal		☐	
Product_ID	decimal		☐	
Products_Name_id	string		☐	
Products_Name_identifier	string		☐	
record_id	string		☑	

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		☐	
Description	string		☐	
Image_data	byte[]		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
Price	decimal		☐	
Product_ID	decimal		☐	
Products_Name_id	string		☐	
Products_Name_identifier	string		☐	
record_id	string		☑	

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		☐	
AddressWithGeocoding_latitude	decimal		☐	
AddressWithGeocoding_longitude	decimal		☐	
AddressWithGeocoding_state	string		☐	
AddressWithGeocoding_street	string		☐	
AddressWithGeocoding_street2	string		☐	
AddressWithGeocoding_zip	string		☐	
Auto_Increment	decimal		☐	
Currency	decimal		☐	
Date_Time	datetime		☐	
Email_email	string		☐	
Email_label	string		☐	
Equation	decimal		☐	
File_filename	string		☐	
File_s3	boolean		☐	
File_size	int32		☐	
File_thumb_url	string		☐	
File_type	string		☐	
File_url	string		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
IntlAddress_city	string		☐	
IntlAddress_state	string		☐	
IntlAddress_street	string		☐	
IntlAddress_street2	string		☐	
IntlAddress_zip	string		☐	
IntlAddressLatLng_latitude	decimal		☐	
IntlAddressLatLng_longitude	decimal		☐	
IntlAddressWithCountry_city	string		☐	
IntlAddressWithCountry_country	string		☐	
IntlAddressWithCountry_state	string		☐	
IntlAddressWithCountry_street	string		☐	
IntlAddressWithCountry_street2	string		☐	
IntlAddressWithCountry_zip	string		☐	
Link_label	string		☐	
Link_url	string		☐	
Multiple_Choice	string		☐	

Name	Data Type	Label	Required	Documentation
Name_firstname	string		☐	
Name_lastname	string		☐	
Name_middlename	string		☐	
Name_title	string		☐	
Number	decimal		☐	
Paragraph_Text	string		☐	
Phone	string		☐	
Rating	decimal		☐	
record_id	string		☑	
Rich_Text	string		☐	
Signature_base30	string		☐	
Signature_svg	string		☐	
Test_Name	string		☐	
Text_Formula	string		☐	
Timer_from_min	datetime		☐	
Timer_stop_count	int32		☐	
Timer_to_max	datetime		☐	
Timer_total_time	int64		☐	
USAddress_city	string		☐	
USAddress_state	string		☐	
USAddress_street	string		☐	
USAddress_street2	string		☐	
USAddress_zip	string		☐	
Yes_No	boolean		☐	

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		☐	
File_s3	boolean		☐	
File_size	int32		☐	
File_thumb_url	string		☐	
File_type	string		☐	
File_url	string		☐	
ID	decimal		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
Name	string		☐	
record_id	string		☑	

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[]		☐	
File_filename	string		☐	
File_s3	boolean		☐	
File_size	int32		☐	
File_thumb_url	string		☐	
File_type	string		☐	
File_url	string		☐	
ID	decimal		☐	
Image_data	byte[]		☐	
Image_filename	string		☐	
Image_s3	boolean		☐	
Image_size	int32		☐	
Image_thumb_url	string		☐	
Image_type	string		☐	
Image_url	string		☐	
Name	string		☐	
record_id	string		☑	

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		☐	

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

Name	Data Type	Label	Required	Documentation
Multiple_Choice	string		☐	
Name_firstname	string		☐	
Name_lastname	string		☐	
Name_middlename	string		☐	
Name_title	string		☐	
Number	decimal		☐	
Paragraph_Text	string		☐	
Phone	string		☐	
Rating	decimal		☐	
record_id	string		☑	
Rich_Text	string		☐	
Signature_base30	string		☐	
Signature_svg	string		☐	
Test_Name	string		☐	
Text_Formula	string		☐	
Timer_from_min	datetime		☐	
Timer_stop_count	int32		☐	
Timer_to_max	datetime		☐	
Timer_total_time	int64		☐	
USAddress_city	string		☐	
USAddress_state	string		☐	
USAddress_street	string		☐	
USAddress_street2	string		☐	
USAddress_zip	string		☐	
Yes_No	boolean		☐	

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		☐	
record_id	string		☑	
text	string		☐	

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		

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		☑	Default column added since the specification specifies that no data is returned.
record_id	string		☑	

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 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: `/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		☐	
Description	string		☐	
Image	string		☐	
Price	string		☐	
Product_ID	string		☐	
Products_Name	string		☐	
record_id	string		☑	

Index

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -

- V -

- W -

- Y -

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

23.0.92
24/08/2024, 20:26
ISBN 978-94-6302-000-8: