23.0

Localazy 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 Localazy API

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

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

Localazy is online software for computer assisted translation of software applications.

The Localazy driver covers 43 tables and 248 columns.

Localazy API Clients

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

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

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

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

Monitor API Calls

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

Specifications

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

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

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

Changes and bug fixes on the Localazy SQL driver can be found in the release notes. Get access to the community through the Localazy section of the Invantive forums.

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

Alias: `lcy`

Recommended alias: `lcy`

Driver code for use in settings.xml

https://localazy.com/docs

Updated 20-09-2024 18:35 using Invantive UniversalSQL version 24.1.8-BETA+4877.

2 SQL Driver Attributes for Localazy API

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

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

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

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

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

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

The Localazy 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	Enforce rows to be unique for software analysis. A fingerprint is calculated from the whole row of data when the primary key column is unknown.	Shared	False	✓	✓	✓	
api-url	URL of web service.	OData		✓		✓	
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	250	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when HTTP server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-408-request-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 408.		10	✓	✓	✓	
download-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 408.		10000	✓	✓	✓	
download-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 408.		300000	✓	✓	✓	
download-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 408.		2	✓	✓	✓	
download-error-422-bad-request-	Maximum number of tries when HTTP server reports unprocessable		30	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
max-tries	entity during retrieval of data.						
download-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
download-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
download-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
download-error-429-too-many-requests-max-tries	Maximum number of tries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
download-error-429-too-many-requests-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10000	✓	✓	✓	
download-error-429-too-many-requests-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		300000	✓	✓	✓	
download-error-429-too-many-requests-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
download-error-500-internal-server-error-max-tries	{res:itgen_pae_download_error_500_internal_server_error_max_tries}		10	✓	✓	✓	
download-error-500-internal-server-error-sleep-initial-ms	{res:itgen_pae_download_error_500_internal_server_error_sleep_initial_ms}		10000	✓	✓	✓	
download-error-500-internal-server-error-sleep-max-ms	{res:itgen_pae_download_error_500_internal_server_error_sleep_max_ms}		300000	✓	✓	✓	
download-error-500-internal-server-error-sleep-multiplicator	{res:itgen_pae_download_error_500_internal_server_error_sleep_multiplicator}		2	✓	✓	✓	
download-error-502-server-unavailable-max-tries	Maximum number of tries when HTTP server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
download-error-502-server-	Initial sleep in milliseconds between retries when HTTP server reports a		10000	✓	✓	✓	

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

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
connect-timeout-sleep-max-ms	reports a HTTP status 590.						
download-error-590-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 590.		2	✓	✓	✓	
download-error-599-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 599.		10	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 599.		10000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 599.		300000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 599.		2	✓	✓	✓	
download-error-argument-exception-max-tries	Maximum number of tries when an argument exception is returned when downloading a blob.		10	✓	✓	✓	
download-error-argument-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		10000	✓	✓	✓	
download-error-argument-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		300000	✓	✓	✓	
download-error-argument-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.		2	✓	✓	✓	
download-error-internet-down-max-tries	Maximum number of tries when the Internet connection seems down during retrieval of data.		10	✓	✓	✓	
download-error-internet-down-sleep-initial-ms	Initial sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		10000	✓	✓	✓	
download-error-internet-down-sleep-max-ms	Maximum sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		300000	✓	✓	✓	
download-error-internet-down-sleep-multiplicator	Multiplication factor for sleep between retries when the Internet connection seems down during retrieval of data.		2	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from 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-name-resolution-failure-max-tries	Maximum number of tries when the host name could not be resolved during retrieval of data.		5	✓	✓	✓	
download-error-name-resolution-failure-sleep-initial-ms	Initial sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
download-error-name-resolution-failure-sleep-max-ms	Maximum sleep in milliseconds between retries when the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
download-error-name-resolution-failure-sleep-multiplicator	{res:itgen_pae_download_error_iname_resolution_failure_sleep_multiplicator}		1	✓	✓	✓	
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	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
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	✓	✓	✓	
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-found-max-tries	{res:itgen_pae_download_error_web_not_found_max_tries}		1	✓	✓	✓	
download-error-web-not-found-sleep-initial-ms	{res:itgen_pae_download_error_web_not_found_sleep_initial_ms}		10000	✓	✓	✓	
download-error-web-not-found-sleep-max-ms	{res:itgen_pae_download_error_web_not_found_sleep_max_ms}		300000	✓	✓	✓	
download-error-web-not-found-sleep-multiplicator	{res:itgen_pae_download_error_web_not_found_sleep_multiplicator}		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-	Initial sleep in milliseconds between retries when the connection reports		10000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
implemented-sleep-initial-ms	not implemented.						
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	✓	✓	✓	
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\guido\Invantive\Cache\http\guido\shared	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.	Shared	False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.	Shared	2592000	✓	✓	✓	
http-get-timeout-max-ms	HTTP GET maximum timeout on retry (ms).		24000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		56000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.	OData	5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.	OData	14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		57000	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		1	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		1000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
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	✓	✓	✓	
invantive-sql-execution-profile-disk-path	{res:itgen_pae_invantive_sql_execution_profile_disk_path}	SQL Engine V1	c:\temp\profiles	✓	✓	✓	
invantive-sql-execution-profile-to-disk	{res:itgen_pae_invantive_sql_execution_profile_to_disk}	SQL Engine V1	True	✓	✓	✓	
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	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
max-odata-filters	Maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-odata-rewrite-in-count	{res:itgen_pae_max_odata_rewrite_in_count}	OData	500	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
maximum-url-length	Maximum URL Length			✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.	OData	2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	10000	✓	✓	✓	
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.	OData	2	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	1100	✓		✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared	10	✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results		0	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	with the HTTP endpoint.						
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-504-errors-percentage	{res:itgen_pae_simulate_http_504_errors_percentage}		0	✓	✓	✓	
simulate-http-504-errors	{res:itgen_pae_simulate_http_504_errors}		False	✓	✓	✓	
simulate-http-522-errors-percentage	{res:itgen_pae_simulate_http_522_errors_percentage}		0	✓	✓	✓	
simulate-http-522-errors	{res:itgen_pae_simulate_http_522_errors}		False	✓	✓	✓	
simulate-http-524-errors-percentage	{res:itgen_pae_simulate_http_524_errors_percentage}		0	✓	✓	✓	
simulate-http-524-errors	{res:itgen_pae_simulate_http_524_errors}		False	✓	✓	✓	
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors when exchanging results with the HTTP endpoint..		False	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	61000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared	100	✓		✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
use-batch-insert	Whether to use batch insert.	OData	True	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.	Shared	False	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.	Shared	False	✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.	OData	True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.	OData	True	✓	✓	✓	

3 Catalog: Localazy

3.1 Schemas

3.1.1 Schema: CodeValues

3.1.1.1 Tables

ProjectRoles: Localazy Domain ProjectRoles

Lookup-table that translates code values of the domain 'ProjectRoles' into descriptions.

Catalog: Localazy

Schema: CodeValues

Label: Domain ProjectRoles

Example: The code value 'none' means 'None'.

Retrieve: true

View Columns

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

Name	Data Type	Label	Required	Documentation
code	string(240)	Code	☑	Unique code within the domain.
description	string(240)	Description	☐	Meaning of the unique code within the domain.
domain	string(240)	Domain	☑	Each domain defines a map of codes to values.
resource_code	string(240)	Resource Code	☐	Resource code for internationalization.

Values

Code	Description
developer	{res:itgen_lcy_projectroles_developer}
manager	{res:itgen_lcy_projectroles_manager}
none	{res:itgen_lcy_projectroles_none}
ow ner	{res:itgen_lcy_projectroles_ow ner}
review er	{res:itgen_lcy_projectroles_review er}
translator	{res:itgen_lcy_projectroles_translator}
trusted_translator	{res:itgen_lcy_projectroles_trusted_translator}

ProjectTones: Localazy Domain ProjectTones

Lookup-table that translates code values of the domain 'ProjectTones' into descriptions.

Catalog: Localazy

Schema: CodeValues

Label: Domain ProjectTones

Example: The code value 'not_specified' means 'Unspecified'.

Retrieve: true

View Columns

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

Name	Data Type	Label	Required	Documentation
code	string(240)	Code	☑	Unique code w ithin the domain.
description	string(240)	Description	☐	Meaning of the unique code w ithin the domain.
domain	string(240)	Domain	☑	Each domain defines a map of codes to values.
resource_code	string(240)	Resource Code	☐	Resource code for internationalization.

Values

Code	Description
formal	{res:itgen_lcy_projecttones_formal}
informal	{res:itgen_lcy_projecttones_informal}
not_specified	{res:itgen_lcy_projecttones_not_specified}

ProjectTypes: Localazy Domain ProjectTypes

Lookup-table that translates code values of the domain 'ProjectTypes' into descriptions.

Catalog: Localazy

Schema: CodeValues

Label: Domain ProjectTypes

Example: The code value 'public' means 'Public'.

Retrieve: true

View Columns

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

Name	Data Type	Label	Required	Documentation
code	string(240)	Code	☑	Unique code within the domain.
description	string(240)	Description	☐	Meaning of the unique code within the domain.
domain	string(240)	Domain	☑	Each domain defines a map of codes to values.
resource_code	string(240)	Resource Code	☐	Resource code for internationalization.

Values

Code	Description
private	{res:itgen_lcy_projecttypes_private}
public	{res:itgen_lcy_projecttypes_public}
restricted	{res:itgen_lcy_projecttypes_restricted}

WebhookEventTypes: Localazy Domain WebhookEventTypes

Lookup-table that translates code values of the domain 'WebhookEventTypes' into descriptions.

Catalog: Localazy

Schema: CodeValues

Label: Domain WebhookEventTypes

Example: The code value 'comment_added' means 'Comment added'.

Retrieve: true

View Columns

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

Name	Data Type	Label	Required	Documentation
code	string(240)	Code	☑	Unique code within the domain.
description	string(240)	Description	☐	Meaning of the unique code within the domain.

Name	Data Type	Label	Required	Documentation
domain	string(240)	Domain	☑	Each domain defines a map of codes to values.
resource_code	string(240)	Resource Code	☐	Resource code for internationalization.

Values

Code	Description
comment_added	{res:itgen_lcy_webhookeventtypes_comment_added}
import_finished_empty	{res:itgen_lcy_webhookeventtypes_import_finished_empty}
import_finished	{res:itgen_lcy_webhookeventtypes_import_finished}
project_published	{res:itgen_lcy_webhookeventtypes_project_published}
tag_promoted	{res:itgen_lcy_webhookeventtypes_tag_promoted}

3.1.2 Schema: Localazy

3.1.2.1 Tables

ImportFormatArrays

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_import_format_arrays}

Documentation:

Import format arrays.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
isDefault	boolean	{res:itgen_lcy_isdefault}	☑	
name	string	{res:itgen_lcy_name}	☑	
parent_type	string	{res:itgen_lcy_parent_type}	☑	Type of the file that can be used in content.type.
type	string	{res:itgen_lcy_type}	☑	

ImportFormatPluralRequiredParameters

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_import_format_plural_required_parameters}

Documentation:

Import format plural required parameters.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
description	string	{res:itgen_lcy_description}	☑	
parent_parent_type	string	{res:itgen_lcy_parent_parent_type}	☑	Type of the file that can be used in content.type.
parent_type	string	{res:itgen_lcy_parent_type}	☑	
type	string	{res:itgen_lcy_type}	☑	

ImportFormatPlurals

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_import_format_plurals}

Documentation:

Import format plurals.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
isDefault	boolean	{res:itgen_lcy_isdefault}	☑	
name	string	{res:itgen_lcy_name}	☑	
parent_type	string	{res:itgen_lcy_parent_type}	☑	Type of the file that can be used in content.type.

Name	Data Type	Label	Required	Documentation
type	string	{res:itgen_lcy_type}	☑	

ImportFormats

Catalog: Localazy

Schema: Localazy

Primary Keys: type

Label: {res:itgen_localazy_import_formats}

Documentation:

Import formats.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
name	string	{res:itgen_lcy_name}	☑	Name of the type.
supportArrays	boolean	{res:itgen_lcy_supportarrays}	☑	Indicates w hether the type supports string arrays.
supportPlurals	boolean	{res:itgen_lcy_supportplurals}	☑	Indicates w hether the type supports plurals.
supportStrings	boolean	{res:itgen_lcy_supportstrings}	☑	Indicates w hether the type supports plain strings.
supportStructuredKeys	boolean	{res:itgen_lcy_supportstructuredkeys}	☑	Indicates w hether the type supports structured/nested keys.
type	string	{res:itgen_lcy_type}	☑	Type of the file that can be used in content.type.

ImportFormatTransformers

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_import_format_transformers}

Documentation:

Import format key transformers.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
isDefault	boolean	{res:itgen_lcy_isdefault}	☑	
name	string	{res:itgen_lcy_name}	☑	
parent_type	string	{res:itgen_lcy_parent_type}	☑	Type of the file that can be used in content.type.
type	string	{res:itgen_lcy_type}	☑	

ProjectCdnByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_cdn_by_project_id}

Documentation:

Project CDN files.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
enabled	boolean	{res:itgen_lcy_enabled}	☑	

ProjectCdnFilesByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_cdn_files_by_project_id}

Documentation:

Project CDN files.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
metadataUrl	string	{res:itgen_lcy_metadataurl}	☐	
tagId	string	{res:itgen_lcy_tagid}	☑	
tagName	string	{res:itgen_lcy_tagname}	☑	

ProjectFileDownloadsByProjectFileAndLanguage

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_file_downloads_by_project_file_and_language}

Documentation:

Project file downloads.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
fileId	string	☑		File ID.
id	string	☑		Project ID.
lang	string	☑		Language.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
contents	string	{res:itgen_lcy_contents}	☐	Raw file contents.

ProjectFileFlavoursByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_file_flavours_by_project_id}

Documentation:

Project file flavours.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
file_id	string	{res:itgen_lcy_file_id}	☑	Unique identifier of the file.
product_flavor	string	{res:itgen_lcy_product_flavor}	☑	Product flavor.

ProjectFileKeysByProjectFileAndLanguage

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_file_keys_by_project_file_and_language}

Documentation:

Project file keys.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
fileId	string	☑		File ID.
id	string	☑		Project ID.
lang	string	☑		Locale code from ProjectLanguages.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
comment	string	{res:itgen_lcy_comment}	☐	Translation note for context.
deprecated	int32	{res:itgen_lcy_deprecated}	☑	Whether the string is deprecated.
hidden	boolean	{res:itgen_lcy_hidden}	☑	Whether the string is hidden from translation interface.
id	string	{res:itgen_lcy_id}	☑	Unique Id of the key in Localazy.
keys	string	{res:itgen_lcy_keys}	☑	Aray of key components. For nested keys it contains the separate levels. For simple string keys it contains just one item.
limit	int32	{res:itgen_lcy_limit}	☑	Translation length limit for this key.
value	string	{res:itgen_lcy_value}	☐	Value represents the translation. It can be either string, array or object for plurals.
vid	int64	{res:itgen_lcy_vid}	☑	Unique identifier of the current version of the translation. It can be used to determine whether the translation has changed from the last time. Useful for two-way synchronization.

ProjectFilesByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_files_by_project_id}

Documentation:

Project files.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
buildType	string	{res:itgen_lcy_buildtype}	☐	A build type the file is associated with. Optional and only available if provided.
id	string	{res:itgen_lcy_id}	☑	Unique identifier of the file.
module	string	{res:itgen_lcy_module}	☐	The module the file belongs to. Optional and only available if provided.
name	string	{res:itgen_lcy_name}	☑	Name of the file.
path	string	{res:itgen_lcy_path}	☑	Stored path to the file. Optional and only available if provided.

Name	Data Type	Label	Required	Documentation
type	string	{res:itgen_lcy_type}	☑	Type of the file; please refer to file formats. Value complex is used for complex files described above.

ProjectGlossaryTermsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_glossary_terms_by_project_id}

Documentation:

Project glossary terms.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
caseSensitive	boolean	{res:itgen_lcy_casesensitive}	☑	Wheter the term is case sensitive or not.
description	string	{res:itgen_lcy_description}	☐	Description of the glossary term.

Name	Data Type	Label	Required	Documentation
exactMatch	boolean	{res:itgen_lcy_exactmatch}	☑	
id	string	{res:itgen_lcy_id}	☑	ID of the glossary term.
translateTerm	boolean	{res:itgen_lcy_translateterm}	☑	Whether the term should be translated or left as is.

ProjectGlossaryTermTranslationsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_glossary_term_translations_by_project_id}

Documentation:

Project glossary term translations.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
lang	string	{res:itgen_lcy_lang}	☑	Language code in which the term is used. Use source language if not translatable.

Name	Data Type	Label	Required	Documentation
project_id	string	{res:itgen_lcy_project_id}	☑	ID of the glossary term.
term	string	{res:itgen_lcy_term}	☑	The value of the glossary term.

ProjectLanguages

Catalog: Localazy

Schema: Localazy

Primary Keys: id

Label: {res:itgen_localazy_project_languages}

Documentation:

Project languages.

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
active	int32	{res:itgen_lcy_active}	☑	Number of active keys.
code	string	{res:itgen_lcy_code}	☑	Locale code.
current	int32	{res:itgen_lcy_current}	☑	Number of keys with approved version/translation.
id	string	{res:itgen_lcy_id}	☑	ID.
name	string	{res:itgen_lcy_name}	☑	English name of the language / locale.
needImprovement	int32	{res:itgen_lcy_needimprovement}	☑	Number of keys in the 'need review' state.
project_id	string	{res:itgen_lcy_project_id}	☑	Project ID.
project_name	string	{res:itgen_lcy_project_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_project_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
review	int32	{res:itgen_lcy_review}	☑	Number of keys waiting for review.
sourceChanged	int32	{res:itgen_lcy_sourcechanged}	☑	Number of keys in the 'source changed' state.
translated	int32	{res:itgen_lcy_translated}	☑	Number of keys that are already translated (but may not be approved yet).

ProjectLinksByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_links_by_project_id}

Documentation:

Project links.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
keyId	string	{res:itgen_lcy_keyid}	☑	ID of the key in Localazy.
linkedKeyId	string	{res:itgen_lcy_linkedkeyid}	☑	ID of the target key it is linked to.
linkedProjectId	string	{res:itgen_lcy_linkedprojectid}	☑	ID of the project in Localazy the target key comes from. The cross-project linking is not available yet.

Projects

Catalog: Localazy

Schema: Localazy

Primary Keys: id

Label: {res:itgen_localazy_projects}

Documentation:

Projects.

Retrieve: true

Insert: true

Update: false

Delete: false

Table Columns

The columns of the table Projects 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
description	string	{res:itgen_lcy_description}	☑	Project description.
id	string	{res:itgen_lcy_id}	☑	ID.
image	string	{res:itgen_lcy_image}	☑	Full URL to the project image or empty string if there is no image available.
name	string	{res:itgen_lcy_name}	☑	Project name.
organization_additionalMt	boolean	{res:itgen_lcy_organization_additionalmt}	☑	
organization_availableKeys	int32	{res:itgen_lcy_organization_availablekeys}	☑	
organization_connectedApps	boolean	{res:itgen_lcy_organization_connectedapps}	☑	
organization_figma	boolean	{res:itgen_lcy_organization_figma}	☑	
organization_formatConversions	boolean	{res:itgen_lcy_organization_formatconversions}	☑	
organization_mtPretranslate	boolean	{res:itgen_lcy_organization_mtpretranslate}	☑	
organization_releaseTags	boolean	{res:itgen_lcy_organization_releasetags}	☑	
organization_screenshots	boolean	{res:itgen_lcy_organization_screenshots}	☑	
organization_screenshotsForFigma	boolean	{res:itgen_lcy_organization_screenshotsforfigma}	☑	
organization_usedKeys	int32	{res:itgen_lcy_organization_usedkeys}	☑	
organization_webhooks	boolean	{res:itgen_lcy_organization_webhooks}	☑	
orgId	string	{res:itgen_lcy_orgid}	☑	dentifier of the organization the project belongs to.

Name	Data Type	Label	Required	Documentation
role	string	{res:itgen_lcy_role}	☑	Role of the current user accessing API (based on the token); one of none, translator, trusted_translator, reviewer, manager, owner and developer.
slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
sourceLanguage	int32	{res:itgen_lcy_sourcelanguage}	☐	he identifier of the source language of the project.
tone	string	{res:itgen_lcy_tone}	☑	Project tone; one of not_specified, formal and informal.
type	string	{res:itgen_lcy_type}	☑	Project type; one of public, private and restricted.
url	string	{res:itgen_lcy_url}	☑	Full URL to the project on Localazy.

ProjectScreenshotAvailableTagsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_screenshot_available_tags_by_project_id}

Documentation:

Project screenshot available tags.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
tag	string	{res:itgen_lcy_tag}	☑	Tag.

ProjectScreenshotPhrasesByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_screenshot_phrases_by_project_id}

Documentation:

Project screenshot phrases.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
phase	string	{res:itgen_lcy_phase}	☑	Phrase.

Name	Data Type	Label	Required	Documentation
screenshot_id	string	{res:itgen_lcy_screenshot_id}	☑	Localazy identifier of a screenshot.

ProjectScreenshotsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_screenshots_by_project_id}

Documentation:

Project screenshots.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
comment	string	{res:itgen_lcy_comment}	☐	
id	string	{res:itgen_lcy_id}	☑	
metadata_name	string	{res:itgen_lcy_metadata_name}	☐	
ocrData	string	{res:itgen_lcy_ocrdata}	☐	

Name	Data Type	Label	Required	Documentation
url	string	{res:itgen_lcy_url}	☑	

ProjectScreenshotTagsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_screenshot_tags_by_project_id}

Documentation:

Project screenshot tags.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
screenshot_id	string	{res:itgen_lcy_screenshot_id}	☑	Localazy identifier of a screenshot.
tag	string	{res:itgen_lcy_tag}	☑	Tag.

ProjectWebhookEventsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_webhook_events_by_project_id}

Documentation:

Project web hook events.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
event	string	{res:itgen_lcy_event}	☑	Event.
parent_customId	string	{res:itgen_lcy_parent_customid}	☑	Custom ID that is passed w hen the w ebhook is invoked. Empty by default.

ProjectWebhooksByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_webhooks_by_project_id}

Documentation:

Project web hooks.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
customId	string	{res:itgen_lcy_customid}	☑	Custom ID that is passed when the webhook is invoked. Empty by default.
description	string	{res:itgen_lcy_description}	☑	Description of the webhook. Empty by default.
enabled	boolean	{res:itgen_lcy_enabled}	☑	Wheter the webhook is enabled or disabled.
events	string	{res:itgen_lcy_events}	☑	The list of event types for which this webhook is invoked.
url	string	{res:itgen_lcy_url}	☑	URL which is invoked on the webhook event.

ProjectWebhookSecretsByProjectId

Catalog: Localazy

Schema: Localazy

Label: {res:itgen_localazy_project_webhook_secrets_by_project_id}

Documentation:

Project web hook secret.

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id	string	☑		Project ID.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
secret	string	{res:itgen_lcy_secret}	☑	Webhook secret.

3.1.3 Schema: Native

3.1.3.1 Tables

NATIVEPLATFORMSCALARREQUESTS: Localazy Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: Localazy

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 Localazy 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 Localazy 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 w hether 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 w ithout Actions	☑	
DURATION_MS	int64	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 w hen 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.
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	int32	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	☐	

Name	Data Type	Label	Required	Documentation
RESULT_DATE_TIME_UTC	datetime	Result Date Time	☐	
RESULT_NUMBER	decimal	Result Number	☐	
RESULT_TEXT	string	Result Text	☐	
SUCCESSFUL	boolean	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int64	Transaction ID	☑	Incrementing ID of the transaction.
URL	string(4000)	URL	☐	

3.1.4 Schema: Views

3.1.4.1 Views

ProjectCdnFiles

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
metadataUrl	string	{res:itgen_lcy_metadataurl}	☐	
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests w here projectId is required.
tagId	string	{res:itgen_lcy_tagid}	☑	
tagName	string	{res:itgen_lcy_tagname}	☑	

ProjectCdns

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
enabled	boolean	{res:itgen_lcy_enabled}	☑	
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.

ProjectFileFlavours

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
file_id	string	{res:itgen_lcy_file_id}	☑	Unique identifier of the file.
product_flavor	string	{res:itgen_lcy_product_flavor}	☑	Product flavor.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.

ProjectFileKeys

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
comment	string	{res:itgen_lcy_comment}	☐	Translation note for context.
deprecated	int32	{res:itgen_lcy_deprecated}	☑	Whether the string is deprecated.
hidden	boolean	{res:itgen_lcy_hidden}	☑	Whether the string is hidden from translation interface.
id	string	{res:itgen_lcy_id}	☑	Unique Id of the key in Localazy.
keys	string	{res:itgen_lcy_keys}	☑	Aray of key components. For nested keys it contains the separate levels. For simple string keys it contains just one item.
language_code	string	{res:itgen_lcy_code}	☑	Locale code.
language_id	string	{res:itgen_lcy_id}	☑	ID.
language_name	string	{res:itgen_lcy_name}	☑	English name of the language / locale.
limit	int32	{res:itgen_lcy_limit}	☑	Translation length limit for this key.
project_id	string	{res:itgen_lcy_project_id}	☑	Project ID.
project_name	string	{res:itgen_lcy_project_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_project_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
value	string	{res:itgen_lcy_value}	☐	Value represents the translation. It can be either string, array or object for plurals.
vid	int64	{res:itgen_lcy_vid}	☑	Unique identifier of the current version of the translation. It can be used to determine whether the translation has changed from the last time. Useful for two-way synchronization.

ProjectFiles

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
buildType	string	{res:itgen_lcy_buildtype}	☐	A build type the file is associated with. Optional and only available if provided.

Name	Data Type	Label	Required	Documentation
id	string	{res:itgen_lcy_id}	☑	Unique identifier of the file.
module	string	{res:itgen_lcy_module}	☐	The module the file belongs to. Optional and only available if provided.
name	string	{res:itgen_lcy_name}	☑	Name of the file.
path	string	{res:itgen_lcy_path}	☑	Stored path to the file. Optional and only available if provided.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
type	string	{res:itgen_lcy_type}	☑	Type of the file; please refer to file formats. Value complex is used for complex files described above.

ProjectGlossaryTerms

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
caseSensitive	boolean	{res:itgen_lcy_casesensitive}	☑	Wheter the term is case sensitive or not.
description	string	{res:itgen_lcy_description}	☐	Description of the glossary term.
exactMatch	boolean	{res:itgen_lcy_exactmatch}	☑	
id	string	{res:itgen_lcy_id}	☑	ID of the glossary term.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
translateTerm	boolean	{res:itgen_lcy_translateterm}	☑	Whether the term should be translated or left as is.

ProjectGlossaryTermTranslations

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
lang	string	{res:itgen_lcy_lang}	☑	Language code in which the term is used. Use source language if not translatable.
project_id	string	{res:itgen_lcy_project_id}	☑	ID of the glossary term.
project_id1	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
term	string	{res:itgen_lcy_term}	☑	The value of the glossary term.

ProjectLinks

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
keyId	string	{res:itgen_lcy_keyid}	☑	ID of the key in Localazy.
linkedKeyId	string	{res:itgen_lcy_linkedkeyid}	☑	ID of the target key it is linked to.
linkedProjectId	string	{res:itgen_lcy_linkedprojectid}	☑	ID of the project in Localazy the target key comes from. The cross-project linking is not available yet.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.

ProjectScreenshotAvailableTags

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests w here projectId is required.
tag	string	{res:itgen_lcy_tag}	☑	Tag.

ProjectScreenshotPhrases

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
phase	string	{res:itgen_lcy_phase}	☑	Phrase.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests w here projectId is required.
screenshot_id	string	{res:itgen_lcy_screenshot_id}	☑	Localazy identifier of a screenshot.

ProjectScreenshots

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
comment	string	{res:itgen_lcy_comment}	☐	
id	string	{res:itgen_lcy_id}	☑	
metadata_name	string	{res:itgen_lcy_metadata_name}	☐	
ocrData	string	{res:itgen_lcy_ocrdata}	☐	
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
url	string	{res:itgen_lcy_url}	☑	

ProjectScreenshotTags

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
screenshot_id	string	{res:itgen_lcy_screenshot_id}	☑	Localazy identifier of a screenshot.
tag	string	{res:itgen_lcy_tag}	☑	Tag.

ProjectWebhookEvents

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
event	string	{res:itgen_lcy_event}	☑	Event.
parent_customId	string	{res:itgen_lcy_parent_customid}	☑	Custom ID that is passed when the webhook is invoked. Empty by default.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.

ProjectWebhooks

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
customId	string	{res:itgen_lcy_customid}	☑	Custom ID that is passed when the webhook is invoked. Empty by default.
description	string	{res:itgen_lcy_description}	☑	Description of the webhook. Empty by default.
enabled	boolean	{res:itgen_lcy_enabled}	☑	Wheter the webhook is enabled or disabled.
events	string	{res:itgen_lcy_events}	☑	The list of event types for which this webhook is invoked.
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.

Name	Data Type	Label	Required	Documentation
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
url	string	{res:itgen_lcy_url}	☑	URL which is invoked on the webhook event.

ProjectWebhookSecrets

Catalog: Localazy

Schema: Views

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

View Columns

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

Name	Data Type	Label	Required	Documentation
project_id	string	{res:itgen_lcy_id}	☑	ID.
project_name	string	{res:itgen_lcy_name}	☑	Project name.
project_slug	string	{res:itgen_lcy_slug}	☑	Project slug. Can be used instead of id in requests where projectId is required.
secret	string	{res:itgen_lcy_secret}	☑	Webhook secret.

4 Package: dcr_metadata

4.1 Procedures

4.1.1 dcr_metadata.get_partitions: Localazy Data container metadata package

Get all partitions.

Documentation:

List all partitions.

Index

- A -

- B -

- C -

- D -

- E -

- F -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -

- V -

- W -

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
17/10/2024, 08:38
ISBN 978-94-6302-000-8: