23.0

Sendinblue 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 Sendinblue API

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

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

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

The Sendinblue driver covers 84 tables and 965 columns.

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

For technical users there are command-line editions of Invantive Data Hub running on iOS, Android, Windows, MacOS and Linux. Invantive Data Hub is also often used for enterprise server applications such as ETL. High-volume replication of data taken from the Sendinblue API into traditional databases such as SQL Server (on-premises and Azure), MySQL, PostgreSQL and Oracle is possible using Invantive Data Replicator. Invantive Data Replicator automatically creates and maintains Sendinblue datawarehouses, possibly in combination with data from over 75 other (cloud) platforms. Invantive Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an Sendinblue ADO.net provider.

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

Monitor API Calls

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

Specifications

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

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

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

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

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

Alias: `sendinblue`

Recommended alias: `sib`

Driver code for use in settings.xml

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

Updated 10-06-2024 19:02 using Invantive UniversalSQL version 24.1.3-BETA+4689.

2 SQL Driver Attributes for Sendinblue API

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

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

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

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

The set SQL statement can be executed after log on. The syntax is: `set NAME VALUE`, or for a distributed database: `set NAME@ALIAS VALUE`. In some scenarios you may need to

enclose the driver attribute name in square brackets to escape it from parsing, for instance when a reserved SQL keyword is part of the name. The new value takes effect straight after execution of the set-statement. The set-statement can be executed as often as needed during a session.

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

The Sendinblue driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from 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-access-token	Access Token is a security token for multiple OAuth2 Flows. With an Access Token you can access protected resources. An Access Token must be stored securely since once compromised allows access to your protected resources.	OData		✓	✓	✓	✓
api-client-id	The client ID is a unique identifier of your application. It is generated by registering an application.	OData		✓		✓	✓
api-client-secret	The client secret is to be kept confidential. Such as a password for a logon code, the client secret is the confidential part of an app identified by a client ID. It is needed during the OAuth2 Code Grant Flow together with the refresh token to get access.	OData		✓		✓	✓
api-pre-expiry-refresh-sec	The number of seconds before the token expires to acquire a new token.	OData		✓	✓	✓	
api-redirect-url	The redirect URI is the website a browser session is redirected to after the OAuth2 authentication process has been completed.	OData		✓		✓	✓
api-refresh-token	Refresh Token is a security token for the OAuth2 Code Grant Flow. With a Refresh Token and client secret you can retrieve a renewed access token to access protected resources. A Refresh Token and client secret must be stored securely since once compromised allows access to your protected resources.	OData		✓		✓	✓

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

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
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	itgen_pae_download_error_500_internal_server_error_max_tries		5	✓	✓	✓	
download-error-500-internal-server-error-sleep-initial-ms	itgen_pae_download_error_500_internal_server_error_sleep_initial_ms		1000	✓	✓	✓	
download-error-500-internal-server-error-sleep-max-ms	itgen_pae_download_error_500_internal_server_error_sleep_max_ms		5000	✓	✓	✓	
download-error-500-internal-server-error-sleep-multiplicator	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-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports a bad gateway during retrieval of data.		10000	✓	✓	✓	
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	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Con-nec-tion String	Set from Set SQL-State-ment	Set from Driver s File	Set from Log On
dow nload-error-502-server-unavail-able-sleep-multiplic-ator	Multiplication factor for sleep betw een retries HTTP server reports a bad gatew ay during retrieval of data.		2	✓	✓	✓	
dow nload-error-503-server-unavail-able-max-tries	Maximum number of tries w hen HTTP server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
dow nload-error-503-server-unavail-able-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen HTTP server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
dow nload-error-503-server-unavail-able-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen HTTP server reports that the API server is un-available during retrieval of data.		300000	✓	✓	✓	
dow nload-error-503-server-unavail-able-sleep-multiplic-ator	Multiplication factor for sleep betw een retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a gatew ay timeout.		10	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a gatew ay timeout.		10000	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a gatew ay timeout.		300000	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-multi-plicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a gatew ay timeout.		2	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a HTTP status 590.		10	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 590.		10000	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 590.		300000	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a HTTP status 590.		2	✓	✓	✓	

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

Code	Description	Origin	Default Value	Set from Con-nec-tion String	Set from Set SQL-State-ment	Set from Driver s File	Set from Log On
	tion failure occurs during retrieval of data.						
dow nload-error-io-exception-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen a netw ork I/O connection failure occurs during re-trieval of data.		300000	✓	✓	✓	
dow nload-error-io-exception-sleep-mul-tiplicator	Multiplication factor for sleep betw een retries w hen a netw ork I/O connection failure occurs during re-trieval of data.		2	✓	✓	✓	
dow nload-error-json-exception-max-tries	Maximum number of tries w hen an invalid JSON body is returned.		3	✓	✓	✓	
dow nload-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen an invalid JSON body is returned.		1000	✓	✓	✓	
dow nload-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen an invalid JSON body is returned.		10000	✓	✓	✓	
dow nload-error-json-exception-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen an invalid JSON body is returned.		2	✓	✓	✓	
dow nload-error-name-resolution-fail-ure-max-tries	Maximum number of tries w hen the host name could not be resolved during retrieval of data.		5	✓	✓	✓	
dow nload-error-name-resolution-fail-ure-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
dow nload-error-name-resolution-fail-ure-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the host name could not be resolved during retrieval of data.		5000	✓	✓	✓	
dow nload-error-name-resolution-fail-ure-sleep-multiplic-ator	itgen_pae_dow nload_error_in-ame_resolution_failure_sleep_multi-plicator		1	✓	✓	✓	
dow nload-error-other-exception-max-tries	Maximum number of tries w hen an unqualified error occurs during re-trieval of data.		3	✓	✓	✓	
dow nload-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen an unqualified error ocurrs during retrieval of data.		10000	✓	✓	✓	
dow nload-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen an unqualified error ocurrs during retrieval of data.		300000	✓	✓	✓	
dow nload-error-other-exception-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen an unqualified error ocurrs during retrieval of data.		2	✓	✓	✓	

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

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
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	✓	✓	✓	
duplicate-row-mode	Configure behaviour when multiple rows are discovered during a limited search based upon a fingerprint generated from the primary key value, or all column values when no primary key is known.. The following values can be used: "error" to raise error on duplicate fingerprint values in an API-page of rows, "keep" to ignore duplicate fingerprint values silently, "skip" like "keep" but with logging to trace of duplicate values or "null" for default behaviour (typically "error")..			✓	✓	✓	✓
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from: Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.	Shared	5	✓	✓	✓	

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

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	point.						
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-unknown-path-type	Whether to ignore path types not yet supported. An error will be generated when an unsupported type occurs.		True	✓	✓	✓	✓
ignore-values-unknown-path	Whether to ignore values outside of processed paths. An error will be generated when a value occurs outside a path otherwise.		True	✓	✓	✓	✓
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		1	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		1000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
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	itgen_pae_invantive_sql_execution_profile_disk_path	SQL Engine V1	c:\temp\profiles	✓	✓	✓	
invantive-sql-execution-profile-to-disk	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	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
max-odata-filters	Maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
maximum-discovered-column-count	Maximum number of discovered columns. An error will be generated when the column exceeds this value.		250	✓	✓	✓	✓
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
maximum-url-length	Maximum URL Length			✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.	OData	2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	10000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Con-nec-tion String	Set from Set SQL-State-ment	Set from Driver s File	Set from Log On
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds betw een OAuth reauthentication tries w hen the OAuth authentication fails.	OData	1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep betw een OAuth reauthentication tries w hen the OAuth authentication fails.	OData	2	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of row s to ex-change per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data re-quests from individual partitions on the data container.	Shared	32	✓	✓	✓	
simulate-http-400-er-rors-percentage	Percentage of simulated HTTP 400 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-400-er-rors	Simulate HTTP 400 errors w hen ex-changing results w ith the HTTP end-point.		False	✓	✓	✓	
simulate-http-401-er-rors-percentage	Percentage of simulated HTTP 401 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-er-rors	Simulate HTTP 401 errors w hen ex-changing results w ith the HTTP end-point.		False	✓	✓	✓	
simulate-http-403-er-rors-percentage	Percentage of simulated HTTP 403 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-er-rors	Simulate HTTP 403 errors w hen ex-changing results w ith the HTTP end-point.		False	✓	✓	✓	
simulate-http-408-er-rors-percentage	Percentage of simulated HTTP 408 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-er-rors	Simulate HTTP 408 errors w hen ex-changing results w ith the HTTP end-point.		False	✓	✓	✓	
simulate-http-429-er-rors-percentage	Percentage of simulated HTTP 429 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-504-errors-percentage	itgen_pae_simulate_http_504_errors_percentage		0	✓	✓	✓	
simulate-http-504-errors	itgen_pae_simulate_http_504_errors		False	✓	✓	✓	
simulate-http-522-errors-percentage	itgen_pae_simulate_http_522_errors_percentage		0	✓	✓	✓	
simulate-http-522-errors	itgen_pae_simulate_http_522_errors		False	✓	✓	✓	
simulate-http-524-errors-percentage	itgen_pae_simulate_http_524_errors_percentage		0	✓	✓	✓	
simulate-http-524-errors	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	60000	✓		✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
swagger-specification-download-tries	Specification Download Tries		5	✓	✓	✓	
swagger-specification-file	The Swagger file path, such as C:\temp\swagger.json.			✓	✓	✓	✓
swagger-specification-http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of Swagger specification data in the HTTP disk cache.		86400	✓	✓	✓	
swagger-specification-url	The Swagger URL such as https://example.org/rest/swagger.json.			✓	✓	✓	✓
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 Schema: Account

3.1 Tables

3.1.1 SubscriptionAccount: Sendinblue Subscription Account

Get your account information, plan and credits details

Catalog: Sendinblue

Schema: Account

Label: Subscription Account

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

Table Columns

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

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

3.1.2 SubscriptionAccountPlans: Sendinblue Subscription Account Plans

Get your account information, plan and credits details

Catalog: Sendinblue

Schema: Account

Label: Subscription Account Plans

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
address_city	string	Address City	☐	City information
address_country	string	Address Country	☐	Country information
address_street	string	Address Street	☐	Street information

Name	Data Type	Label	Required	Documentation
address_zipCode	string	Address Zip Code	☐	Zip Code information
companyName	string	Company Name	☐	Name of the company
credits	float	Credits	☐	Remaining credits of the user
creditsType	string	Credit Type	☐	This is the type of the credit, "Send Limit" is one of the possible types of credit of a user. "Send Limit" implies the total number of emails you can send to the subscribers in your account.
email	string	Email Address	☐	Login Email
endDate	datetime	End Date	☐	Date of the period from w hich the plan w ill end (only available for "subscription" and "reseller" plan type)
firstName	string	First Name	☐	First Name
lastName	string	Last Name	☐	Last Name
marketingAutomation_enabled	boolean	Marketing Automation Enabled	☐	Status of Marketing Automation Plateform activation for your account (true=enabled, false=disabled)
marketingAutomation_key	string	Marketing Automation Key	☐	Marketing Automation Tracker ID
relay_data_port	int64	Relay Data Port	☐	Port used for SMTP Relay
relay_data_relay	string	Relay Data Relay	☐	URL of the SMTP Relay
relay_data_userName	string	Relay Data Username	☐	Email to use as login on transactional platform
relay_enabled	boolean	Relay Enabled	☐	Status of your transactional email Account (true=Enabled, false=Disabled)
startDate	datetime	Start Date	☐	Date of the period from w hich the plan w ill start (only available for "subscription" and "reseller" plan type)
type	string	Type	☐	Displays the plan type of the user
userLimit	int64	User Limit	☐	Only in case of reseller account. It implies the total number of child accounts you can add to your account.

4 Schema: Contacts

4.1 Tables

4.1.1 Attributes: Sendinblue Attributes

List all attributes

Catalog: Sendinblue

Schema: Contacts

Label: Attributes

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
calculatedValue	string	Calculated Value	☐	Calculated value formula
category	string	Category	☐	Category of the attribute
name	string	Name	☐	Name of the attribute
type	string	Type	☐	Type of the attribute

4.1.2 AttributeValues: Sendinblue Attribute Values

List all attributes

Catalog: Sendinblue

Schema: Contacts

Label: Attribute Values

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
calculatedValue	string	Calculated Value	☐	Calculated value formula
category	string	Category	☐	Category of the attribute
label	string	Label	☐	Label of the "category" type attribute
name	string	Name	☐	Name of the attribute
type	string	Type	☐	Type of the attribute
value	int64	Value	☐	ID of Value of the "category" type attribute

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

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Unsubscriptions by Email (Administrative Mode)

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
eventTime	datetime	Event Time	☐	UTC date-time of the event
ip	string	IP	☐	IP from which the user has been unsubscribed

4.1.4 ContactClickStatisticsByEmail

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event

4.1.5 ContactComplaintStatisticsByEmail: Sendinblue Contact Complaint Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Complaint Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
eventTime	datetime	Event Time	☐	UTC date-time of the event

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

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Unsubscriptions by Email (Administrative Mode)

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

4.1.7 ContactDetailClickStatisticsByEmail

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

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

4.1.8 ContactDetailComplaintStatisticsByEmail: Sendinblue Contact Detail Complaint Statistics by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Complaint Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	☐	Set of attributes of the contact

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

4.1.9 ContactDetailHardBounceStatisticsByEmail: Sendinblue Contact Detail Hard Bounce Statistics by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Hard Bounce Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

4.1.10 ContactDetailLinkClickStatisticsByEmail: Sendinblue Contact Detail Link Click Statistics by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Link Click Statistics by Email

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactDetailLinkClickStatisticsByEmail. A value must be provided at all times for required

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

4.1.11 ContactDetailMessageSentStatisticsByEmail: Sendinblue Contact Detail Message Sent Statistics by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Message Sent Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

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

4.1.12 ContactDetailOpenStatisticsByEmail: Sendinblue Contact Detail Open Statistics by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Open Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	string	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign which generated the event

Name	Data Type	Label	Required	Documentation
count	int64	Count	☐	Number of openings for the campaign
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	string	Email Address	☐	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	datetime	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for which you requested the details
ip	string	IP	☐	IP from which the user has opened the email
listIds	string	List IDs	☐	
listUnsubscribed	string	Unsubscribed List IDs	☐	
modifiedAt	datetime	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	boolean	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)

4.1.13 ContactDetailSoftBounceStatisticsByEmail: Sendinblue Contact Detail Soft Bounce Statistics by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Soft Bounce Statistics by Email

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

Parameters of Table Function

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

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

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

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

4.1.14 ContactDetailTransactionAttributesByEmail: Sendinblue Contact Detail Transaction Attributes by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail Transaction Attributes by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

4.1.15 ContactDetailUserUnsubscriptionsByEmail: Sendinblue Contact Detail User Unsubscriptions by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Label: Contact Detail User Unsubscriptions by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

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

Name	Data Type	Label	Required	Documentation
				false=not blacklisted)

4.1.16 ContactHardBounceStatisticsByEmail: Sendinblue Contact Hard Bounce Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Hard Bounce Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign which generated the event
eventTime	datetime	Event Time	☐	UTC date-time of the event

4.1.17 ContactInfoByEmail: Sendinblue Contact Information by Email

Get a contact's details

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Contact Information by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email (urlencoded) of the contact OR its SMS attribute value

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF_TOT	string		☐	
ABONNEMENTEN_ACTIEF	string		☐	
BLACKLIST	decimal	Blacklist	☐	
BUBS_GBR_ID_MASTER	decimal		☐	

Name	Data Type	Label	Required	Documentation
BUBS_GBR_ID	decimal	Estate Person ID	☐	
CLICKERS	decimal	Clickers	☐	
COMPANY_NAME	string	Company Name	☐	
CONTRACT_CODE	string	Contract Code	☐	
COUNTRY	string	Country	☐	
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	string		☐	
D31BILLINGIDS	string		☐	
D31BUBSGBRIDS	string		☐	
D31DATACONTAINERCNT	decimal		☐	
D31MBIO	decimal		☐	
D31PARTITIONCNT	decimal		☐	
D31PROVIDERCNT	decimal		☐	
D31ROWCNT	decimal		☐	
D31TABLECNT	decimal		☐	
email	string	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
FIRSTNAME	string	First Name	☐	
FORUMS_LOGIN	string	Forums Login	☐	
id	int64	ID	☐	ID of the contact for w hich you requested the details
JOBTITLE	string	Job Title	☐	
LANDLINE_NUMBER	string		☐	
LANGUAGE_CODE	string	Language Code	☐	
LASTNAME	string	Last Name	☐	
listIds	string	List IDs	☐	
listUnsubscribed	string	Unsubscribed List IDs	☐	
MOBILE_NUMBER	string	Mobile Number	☐	
modifiedAt	datetime	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
PHONE_NUMBER	string	Phone Number	☐	
PRIMARY_AUDIENCE	string	Primary Audience	☐	
READERS	decimal	Readers	☐	
SMS	string	SMS	☐	
smsBlacklisted	boolean	SMS is Blacklisted	☐	Blacklist status for SMS cam-paigns (true=blacklisted, false=not blacklisted)
WHATSAPP	string	WhatsApp	☐	

4.1.18 ContactLinkClicksByEmail: Sendinblue Contact Link Clicks by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Link Clicks by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event

Name	Data Type	Label	Required	Documentation
count	int64	Count	☐	Number of clicks on this link for the campaign
eventTime	datetime	Event Time	☐	UTC date-time of the event
ip	string	IP	☐	IP from w hich the user has clicked on the link
url	string	URL	☐	URL of the clicked link

4.1.19 ContactMessageSentStatisticsByEmail: Sendinblue Contact Message Sent Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Message Sent Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign which generated the event
eventTime	datetime	Event Time	☐	UTC date-time of the event

4.1.20 ContactOpenStatisticsByEmail: Sendinblue Contact Open Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Open Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign which generated the event
count	int64	Count	☐	Number of openings of the campaign
eventTime	datetime	Event Time	☐	UTC date-time of the event
ip	string	IP	☐	IP from which the user has opened the campaign

4.1.21 Contacts: Sendinblue Contacts

Get all the contacts

Catalog: Sendinblue

Schema: Contacts

Primary Keys: email

Label: Contacts

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
createdSince	datetime	☐		Filter (urlencoded) the contacts created after a given UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ). Prefer to pass your timezone in date-time format for accurate result.
modifiedSince	datetime	☐		Filter (urlencoded) the contacts modified after a given UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ). Prefer to

Name	Data Type	Required	Default Value	Documentation
				pass your timezone in date-time format for accurate result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF_TOT	string		☐	
ABONNEMENTEN_ACTIEF	string		☐	
BLACKLIST	decimal	Blacklist	☐	
BUBS_GBR_ID_MASTER	decimal		☐	
BUBS_GBR_ID	decimal	Estate Person ID	☐	
CLICKERS	decimal	Clickers	☐	
COMPANY_NAME	string	Company Name	☐	
CONTRACT_CODE	string	Contract Code	☐	
COUNTRY	string	Country	☐	
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	string		☐	
D31BILLINGIDS	string		☐	
D31BUBSGBRIDS	string		☐	
D31DATACONTAINERCNT	decimal		☐	
D31MBIO	decimal		☐	
D31PARTITIONCNT	decimal		☐	
D31PROVIDERCNT	decimal		☐	
D31ROWCNT	decimal		☐	
D31TABLECNT	decimal		☐	
email	string	Email Address	☐	Email address of the contact for which you requested the details
emailBlacklisted	boolean	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
FIRSTNAME	string	First Name	☐	
FORUMS_LOGIN	string	Forums Login	☐	
id	int64	ID	☐	ID of the contact for which you requested the details
JOBTITLE	string	Job Title	☐	
LANDLINE_NUMBER	string		☐	
LANGUAGE_CODE	string	Language Code	☐	
LASTNAME	string	Last Name	☐	

Name	Data Type	Label	Required	Documentation
listIds	string	List IDs	☐	
listUnsubscribed	string	Unsubscribed List IDs	☐	
MOBILE_NUMBER	string	Mobile Number	☐	
modifiedAt	datetime	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
PHONE_NUMBER	string	Phone Number	☐	
PRIMARY_AUDIENCE	string	Primary Audience	☐	
READERS	decimal	Readers	☐	
SMS	string	SMS	☐	
smsBlacklisted	boolean	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	string	WhatsApp	☐	

4.1.22 ContactsByListId: Sendinblue Contacts by List ID

Get contacts in a list

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Contacts by List ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
listId	int64	☑		Id of the list

Name	Data Type	Required	Default Value	Documentation
modifiedSince	datetime	☐		Filter (urlencoded) the contacts modified after a given UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ). Prefer to pass your timezone in date-time format for accurate result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF_TOT	string		☐	
ABONNEMENTEN_ACTIEF	string		☐	
BLACKLIST	decimal	Blacklist	☐	
BUBS_GBR_ID_MASTER	decimal		☐	
BUBS_GBR_ID	decimal	Estate Person ID	☐	
CLICKERS	decimal	Clickers	☐	
COMPANY_NAME	string	Company Name	☐	
CONTRACT_CODE	string	Contract Code	☐	
COUNTRY	string	Country	☐	
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	string		☐	
D31BILLINGIDS	string		☐	
D31BUBSGBRIDS	string		☐	
D31DATACONTAINERCNT	decimal		☐	
D31MBIO	decimal		☐	
D31PARTITIONCNT	decimal		☐	
D31PROVIDERCNT	decimal		☐	
D31ROWCNT	decimal		☐	
D31TABLECNT	decimal		☐	
email	string	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	boolean	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
FIRSTNAME	string	First Name	☐	
FORUMS_LOGIN	string	Forums Login	☐	
id	int64	ID	☐	ID of the contact for w hich you requested the details
JOBTITLE	string	Job Title	☐	
LANDLINE_NUMBER	string		☐	

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

4.1.23 ContactSoftBounceStatisticsByEmail: Sendinblue Contact Soft Bounce Statistics by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Soft Bounce Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign w hich gener-ated the event
eventTime	datetime	Event Time	☐	UTC date-time of the event

4.1.24 ContactTransactionAttributesByEmail: Sendinblue Contact Transaction Attributes by Email

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Transaction Attributes by Email

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

Parameters of Table Function

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

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

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

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
orderDate	datetime	Order Date	☐	Date of the order
orderId	int64	Order ID	☐	ID of the order
orderPrice	float	Order Price	☐	Price of the order

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

Get email campaigns' statistics for a contact

Catalog: Sendinblue

Schema: Contacts

Label: Contact Unsubscriptions by Email (User Mode)

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address (urlencoded) of the contact
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign w hich gener-ated the event
eventTime	datetime	Event Time	☐	UTC date-time of the event
ip	string	IP	☐	IP from w hich the user has un-subscribed

4.1.26 FolderById: Sendinblue Folder by ID

Returns a folder's details

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Folder by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
folderId	int64	☑		id of the folder

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
id	int64	ID	☐	ID of the folder
name	string	Name	☐	Name of the folder
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the folder
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the folder
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the folder

4.1.27 Folders: Sendinblue Folders

Get all folders

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Folders

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
id	int64	ID	☐	ID of the folder
name	string	Name	☐	Name of the folder
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the folder

Name	Data Type	Label	Required	Documentation
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the folder
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the folder

4.1.28 ListById: Sendinblue List by ID

Get a list's details

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: List by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
listId	int64	☑		Id of the list

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	boolean	Dynamic List	☐	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)

Name	Data Type	Label	Required	Documentation
folderId	int64	Folder ID	☐	ID of the folder
id	int64	ID	☐	ID of the list
name	string	Name	☐	Name of the list
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the list

4.1.29 ListCampaignStatisticsByListId: Sendinblue List Campaign Statistics by List ID

Get a list's details

Catalog: Sendinblue

Schema: Contacts

Label: List Campaign Statistics by List ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
listId	int64	☑		Id of the list

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)

Name	Data Type	Label	Required	Documentation
dynamicList	boolean	Dynamic List	☐	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)
folderId	int64	Folder ID	☐	ID of the folder
id	int64	ID	☐	ID of the list
name	string	Name	☐	Name of the list
stats_clickers	int64	#Clickers	☐	Number of total clicks for the campaign
stats_complaints	int64	#Complaints	☐	Number of complaints (Spam reports) for the campaign
stats_deferred	int64	#Deferred	☐	Number of deferred emails for the campaign
stats_delivered	int64	#Delivered	☐	Number of delivered emails for the campaign
stats_hardBounces	int64	#Hard-bounces	☐	Number of harbounce for the campaign
stats_listId	int64	List ID	☐	List Id of email campaign (only in case of get email campaign(s) (not for global stats))
stats_returnBounce	int64	#Return-bounces	☐	Total number of non-delivered campaigns for a particular campaign id.
stats_sent	int64	#Sent	☐	Number of sent emails for the campaign
stats_softBounces	int64	#Soft-bounces	☐	Number of softbounce for the campaign
stats_trackableViews	int64	#Trackable Views	☐	Recipients without any privacy protection option enabled in their email client
stats_uniqueClicks	int64	#Unique Clicks	☐	Number of unique clicks for the campaign
stats_uniqueViews	int64	#Unique Views	☐	Number of unique openings for the campaign
stats_unsubscriptions	int64	#Unsubscriptions	☐	Number of unsubscription for the campaign
stats_viewed	int64	#Viewed	☐	Number of openings for the campaign
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the list

4.1.30 Lists: Sendinblue Lists

Get all the lists

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Lists

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
folderId	int64	Folder ID	☐	ID of the folder
id	int64	ID	☐	ID of the list
name	string	Name	☐	Name of the list
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the list

4.1.31 ListsByFolderId: Sendinblue Lists by Folder ID

Get lists in a folder

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Lists by Folder ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
folderId	int64	☑		Id of the folder

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
id	int64	ID	☐	ID of the list
name	string	Name	☐	Name of the list
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the list

4.1.32 Segments: Sendinblue Segments

Get all the segments

Catalog: Sendinblue

Schema: Contacts

Primary Keys: id

Label: Segments

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
categoryName	string		☐	Name of the Segment Category
id	int64	ID	☐	ID of the list
segmentName	string		☐	Name of the Segment
updatedAt	string		☐	Updation UTC date-time of the segment (YYYY-MM-DDTHH:mm:ss.SSSZ)

5 Schema: EmailCampaigns

5.1 Tables

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

Get an A/B test email campaign resultsObtain winning version of an A/B test email campaign

Catalog: Sendinblue

Schema: EmailCampaigns

Label: A/B Test Campaign Result by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
campaignId	int64	☑		Id of the A/B test campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
clickRate	string	Clickrate	☐	Click rate for current w inning version
openRate	string	Open Rate	☐	Open rate for current w inning version
statistics_clicks_Version_A	string		☐	percentage of an event for version A
statistics_clicks_Version_B	string		☐	percentage of an event for version B
statistics_complaints_Version_A	string		☐	percentage of an event for version A
statistics_complaints_Version_B	string		☐	percentage of an event for version B
statistics_hardBounces_Version_A	string		☐	percentage of an event for version A
statistics_hardBounces_Version_B	string		☐	percentage of an event for version B
statistics_openers_Version_A	string		☐	percentage of an event for version A
statistics_openers_Version_B	string		☐	percentage of an event for version B

Name	Data Type	Label	Required	Documentation
statistics_softBounces_Version_A	string		☐	percentage of an event for version A
statistics_softBounces_Version_B	string		☐	percentage of an event for version B
statistics_unsubscribed_Version_A	string		☐	percentage of an event for version A
statistics_unsubscribed_Version_B	string		☐	percentage of an event for version B
winningCriteria	string	Winning Criteria	☐	Criteria choosen for winning version (Open/Click)
winningSubjectLine	string	Winning Subject Line	☐	Subject Line of current winning version
winningVersion	string	Winning Version	☐	Winning Campaign Info. pending = Campaign has been picked for sending and winning version is yet to be decided, tie = A tie happened between both the versions, notAvailable = Campaign has not yet been picked for sending.
winningVersionRate	string	Winning Version Rate	☐	Open/Click rate for the winner version

5.1.2 EmailCampaignById: Sendinblue Email Campaign by ID

Get an email campaign report

Catalog: Sendinblue

Schema: EmailCampaigns

Primary Keys: id

Label: Email Campaign by ID

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

Parameters of Table Function

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

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

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

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
campaignId	int64	☑		Id of the campaign
statistics	string	☐		Filter on type of the statistics required. Example **globalStats** value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain, statsByDevice, statsByBrowser)

Columns of Table Function

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

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

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

Name	Data Type	Label	Required	Documentation
				able if `abTesting` flag of the campaign is `true`

5.1.3 EmailCampaignRecipientExclusionLists: Sendinblue Email Campaign Recipient Exclusion Lists

Return all your created email campaigns

Catalog: Sendinblue

Schema: EmailCampaigns

Label: Email Campaign Recipient Exclusion Lists

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	datetime	☐		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent email campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
excludeHtmlContent	boolean	☐		Use this flag to exclude htmlContent from the response body. If set to **true**, htmlContent field w ill be returned as empty string in the response body (Waarden: true, false)
startDate	datetime	☐		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the

Name	Data Type	Required	Default Value	Documentation
				sent email campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
statistics	string	☐		Filter on type of the statistics required. Example **globalStats** value will only fetch globalStats info of the campaign in returned response. (Waarden: globalStats, linksStats, statsByDomain)
status	string	☐		Filter on the status of the campaign (Waarden: suspended, archive, sent, queued, draft, inProcess)
type	string	☐		Filter on the type of the campaigns (Waarden: classic, trigger)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	☐	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
footer	string	Footer	☐	Footer of the campaign
header	string	Header	☐	Header of the campaign
htmlContent	string	HTML Content	☐	HTML content of the campaign
id	int64	ID	☐	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	☐	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true means image can be embedded, in the email.
lists	string		☐	
mirrorActive	boolean	Mirror Active	☐	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
modifiedAt	datetime	Modified Date	☐	UTC date-time of last modification of the campaign (YYYY-MM-

Name	Data Type	Label	Required	Documentation
				DDTHH:mm:ss.SSSZ)
name	string	Name	☐	Name of the campaign
recurring	boolean	Recurring	☐	FOR TRIGGER ONLY ! Type of trigger campaign.recurring = false means contact can receive the same Trigger campaign only once, & recurring = true means contact can receive the same Trigger campaign several times
replyTo	string	Reply-to	☐	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	☐	Total number of non-delivered campaigns for a particular campaign id.
scheduledAt	datetime	Schedule Date	☐	UTC date-time on w hich campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	☐	It is true if you have chosen to send your campaign at best time, otherw ise it is false
sender_email	string	Sender Email Address	☐	Sender email of the campaign
sender_id	int64	Sender ID	☐	Sender id of the campaign
sender_name	string	Sender Name	☐	Sender name of the campaign
sentDate	datetime	Sent Date	☐	Sent UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	☐	Link to share the campaign on social medias
splitRule	int64	Split Rule	☐	The size of your ab-test groups. Only available if `abTesting` flag of the campaign is `true`
statistics_globalStats	string	Global Statistics	☐	Overall statistics of the campaign
statistics_linksStats	string	Link Statistics	☐	Statistics about the number of clicks for the links
statistics_mirrorClick	int64	#Mirror Clicks	☐	Number of clicks on mirror link
statistics_remaining	int64	#Remaining	☐	Number of remaning emails to send
statistics_statsByBrow ser	string	Statistics by Brow ser	☐	Statistics about the campaign on the basis of various brow sers
statistics_statsByDomain	string	Statistics by Domain	☐	
status	string	Status	☐	Status of the campaign
subject	string	Subject	☐	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
subjectA	string	Subject A	☐	Subject A of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
subjectB	string	Subject B	☐	Subject B of the ab-test campaign. Only available if `abTest-

Name	Data Type	Label	Required	Documentation
				ing` flag of the campaign is `true`
tag	string	Tag	☐	Tag of the campaign
testSent	boolean	Test Sent	☐	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	string	To Field	☐	Customisation of the "to" field of the campaign
type	string	Type	☐	Type of campaign
w innerCriteria	string	Winner Criteria	☐	Criteria for the w inning version. Only available if `abTesting` flag of the campaign is `true`
w innerDelay	int64	Winner Delay	☐	The duration of the test in hours at the end of w hich the w inning version w ill be sent. Only available if `abTesting` flag of the campaign is `true`

5.1.4 EmailCampaignRecipientLists: Sendinblue Email Campaign Recipient Lists

Return all your created email campaigns

Catalog: Sendinblue

Schema: EmailCampaigns

Label: Email Campaign Recipient Lists

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	datetime	☐		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the

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

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
abTesting	boolean	A/B-Testing	☐	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
exclusionLists	string		☐	
footer	string	Footer	☐	Footer of the campaign
header	string	Header	☐	Header of the campaign
htmlContent	string	HTML Content	☐	HTML content of the campaign

Name	Data Type	Label	Required	Documentation
id	int64	ID	☐	ID of the campaign
inlineImageActivation	boolean	Inline Image Activation	☐	Status of inline image. inlineImageActivation = false means image can't be embedded, & inlineImageActivation = true means image can be embedded, in the email.
mirrorActive	boolean	Mirror Active	☐	Status of mirror links in campaign. mirrorActive = false means mirror links are deactivated, & mirrorActive = true means mirror links are activated, in the campaign
modifiedAt	datetime	Modified Date	☐	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	☐	Name of the campaign
recurring	boolean	Recurring	☐	FOR TRIGGER ONLY ! Type of trigger campaign.recurring = false means contact can receive the same Trigger campaign only once, & recurring = true means contact can receive the same Trigger campaign several times
replyTo	string	Reply-to	☐	Email defined as the "Reply to" of the campaign
returnBounce	int64	Return Bounce	☐	Total number of non-delivered campaigns for a particular campaign id.
scheduledAt	datetime	Schedule Date	☐	UTC date-time on w hich campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	boolean	Send at Best Time	☐	It is true if you have chosen to send your campaign at best time, otherw ise it is false
sender_email	string	Sender Email Address	☐	Sender email of the campaign
sender_id	int64	Sender ID	☐	Sender id of the campaign
sender_name	string	Sender Name	☐	Sender name of the campaign
sentDate	datetime	Sent Date	☐	Sent UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if 'status' of the campaign is 'sent'
shareLink	string	Share Link	☐	Link to share the campaign on social medias
splitRule	int64	Split Rule	☐	The size of your ab-test groups. Only available if `abTesting` flag of the campaign is `true`
statistics_globalStats	string	Global Statistics	☐	Overall statistics of the campaign
statistics_linksStats	string	Link Statistics	☐	Statistics about the number of clicks for the links
statistics_mirrorClick	int64	#Mirror Clicks	☐	Number of clicks on mirror link

Name	Data Type	Label	Required	Documentation
statistics_remaining	int64	#Remaining	☐	Number of remaning emails to send
statistics_statsByBrowser	string	Statistics by Browser	☐	Statistics about the campaign on the basis of various browsers
statistics_statsByDomain	string	Statistics by Domain	☐	
status	string	Status	☐	Status of the campaign
subject	string	Subject	☐	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
subjectA	string	Subject A	☐	Subject A of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
subjectB	string	Subject B	☐	Subject B of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
tag	string	Tag	☐	Tag of the campaign
testSent	boolean	Test Sent	☐	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	string	To Field	☐	Customisation of the "to" field of the campaign
type	string	Type	☐	Type of campaign
winnerCriteria	string	Winner Criteria	☐	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`
winnerDelay	int64	Winner Delay	☐	The duration of the test in hours at the end of which the winning version will be sent. Only available if `abTesting` flag of the campaign is `true`

5.1.5 EmailCampaignRecipientsByCampaignId: Sendinblue Email Campaign Recipients by Campaign ID

Export the recipients of an email campaign

Catalog: Sendinblue

Schema: EmailCampaigns

Label: Email Campaign Recipients by Campaign ID

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

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
campaignId	int64	☑		Id of the campaign
notifyURL	string	☐		Webhook called once the export process is finished. For reference, https://help.sendinblue.com/hc/en-us/articles/360007666479
recipientsType	string	☑		Type of recipients to export for a campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
processId	int64	Process ID	☐	Id of the process created
RESULT	string	Result	☐	Outcome of operation as single plain text column.

5.1.6 EmailCampaigns: Sendinblue Email Campaigns

Return all your created email campaigns

Catalog: Sendinblue

Schema: EmailCampaigns

Primary Keys: id

Label: Email Campaigns

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

Parameters of Table Function

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

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

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

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

Columns of Table Function

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

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

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

5.1.7 EmailCampaignStatistics: Sendinblue Email Campaign Statistics

Return all your created email campaigns

Catalog: Sendinblue

Schema: EmailCampaigns

Label: Email Campaign Statistics

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

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
status	string	☐		Filter on the status of the campaign (Waarden: suspended, archive, sent, queued, draft, inProcess)
type	string	☐		Filter on the type of the campaigns (Waarden: classic, trigger)

Columns of Table Function

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

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

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

Name	Data Type	Label	Required	Documentation
winnerCriteria	string	Winner Criteria	☐	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`
winnerDelay	int64	Winner Delay	☐	The duration of the test in hours at the end of which the winning version will be sent. Only available if `abTesting` flag of the campaign is `true`

5.1.8 SharedTemplateUrlByCampaignId: Sendinblue Shared URL Template by Campaign ID

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

Catalog: Sendinblue

Schema: EmailCampaigns

Label: Shared URL Template by Campaign ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
campaignId	int64	☑		Id of the campaign or template

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
sharedUrl	string	Shared URL	☐	A unique URL for the email campaign or transactional template.

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

6 Schema: Native

6.1 Tables

6.1.1 NATIVEPLATFORMSCALARREQUESTS: Sendinblue Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: Sendinblue

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 Sendinblue 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 Sendinblue 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 Max-imum 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 re-

Name	Data Type	Label	Required	Documentation
				quest triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	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	☐	
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	☐	

7 Schema: Process

7.1 Tables

7.1.1 BackgroundProcessById: Sendinblue Background Process by ID

Return the informations for a process

Catalog: Sendinblue

Schema: Process

Primary Keys: id

Label: Background Process by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
processId	int64	☑		Id of the process

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
export_url	string	Export URL	☐	URL on w hich send export the of contacts once the process is completed
id	int64	ID	☐	Id of the process
name	string	Name	☐	Process name
status	string	Status	☐	Status of the process

7.1.2 BackgroundProcesses: Sendinblue Background Processes

Return all the processes for your account

Catalog: Sendinblue

Schema: Process

Primary Keys: id

Label: Background Processes

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
export_url	string	Export URL	☐	URL on w hich send export the of contacts once the process is completed
id	int64	ID	☐	Id of the process
name	string	Name	☐	Process name
status	string	Status	☐	Status of the process

8 Schema: Reseller

8.1 Tables

8.1.1 ChildAccountCreationStatusById: Sendinblue Child Account Creation Status by ID

Get the status of a reseller's child account creation, whether it is successfully created (exists) or not based on the childIdentifier supplied

Catalog: Sendinblue

Schema: Reseller

Label: Child Account Creation Status by ID

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ChildAccountCreationStatusById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
childAccountCreated	boolean	Child Account Created	☐	Status of child account creation w hether it is successfully cre-ated (exists) or not.

8.1.2 ChildApiKeysV2ByChildId: Sendinblue Child API Keys V2 by Child ID

Get a child account's details

Catalog: Sendinblue

Schema: Reseller

Label: Child API Keys V2 by Child ID

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ChildApiKeysV2ByChildId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

The columns of the table function ChildApiKeysV2ByChildId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
companyName	string	Company Name	☐	Name of the company
credits_emailCredits	int64	Email Credits	☐	Email credits available for your child
credits_smsCredits	int64	SMS Credits	☐	SMS credits available for your child
email	string	Email Address	☐	Login Email

Name	Data Type	Label	Required	Documentation
firstName	string	First Name	☐	First Name
key	string	Key	☐	API Key for version 2
lastName	string	Last Name	☐	Last Name
name	string	Name	☐	Name of the key for version 2
passw ord	string	Passw ord	☐	The encrypted passw ord of child account
statistics_currentMonthTotalSent	int64	#Current Month Sent	☐	Overall emails sent for current month
statistics_previousMonthTotalSent	int64	#Previous Month Sent	☐	Overall emails sent for the previous month
statistics_totalSent	int64	#Sent	☐	Overall emails sent for since the account exists

8.1.3 ChildApiKeysV3ByChildId: Sendinblue Child API Keys V3 by Child ID

Get a child account's details

Catalog: Sendinblue

Schema: Reseller

Label: Child API Keys V3 by Child ID

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ChildApiKeysV3ByChildId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

The columns of the table function ChildApiKeysV3ByChildId are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
companyName	string	Company Name	☐	Name of the company
credits_emailCredits	int64	Email Credits	☐	Email credits available for your child
credits_smsCredits	int64	SMS Credits	☐	SMS credits available for your child
email	string	Email Address	☐	Login Email
firstName	string	First Name	☐	First Name
key	string	Key	☐	API Key for version 3
lastName	string	Last Name	☐	Last Name
name	string	Name	☐	Name of the key for version 3
password	string	Password	☐	The encrypted password of child account
statistics_currentMonthTotalSent	int64	#Current Month Sent	☐	Overall emails sent for current month
statistics_previousMonthTotalSent	int64	#Previous Month Sent	☐	Overall emails sent for the previous month
statistics_totalSent	int64	#Sent	☐	Overall emails sent for since the account exists

8.1.4 ChildById: Sendinblue Child by ID

Get a child account's details

Catalog: Sendinblue

Schema: Reseller

Label: Child by ID

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ChildById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
companyName	string	Company Name	☐	Name of the company
credits_emailCredits	int64	Email Credits	☐	Email credits available for your child
credits_smsCredits	int64	SMS Credits	☐	SMS credits available for your child
email	string	Email Address	☐	Login Email
firstName	string	First Name	☐	First Name
lastName	string	Last Name	☐	Last Name
password	string	Password	☐	The encrypted password of child account
statistics_currentMonthTotalSent	int64	#Current Month Sent	☐	Overall emails sent for current month
statistics_previousMonthTotalSent	int64	#Previous Month Sent	☐	Overall emails sent for the previous month
statistics_totalSent	int64	#Sent	☐	Overall emails sent for since the account exists

8.1.5 ChildDomainsByChildId: Sendinblue Child Domains by Child ID

Get all sender domains for a specific child account

Catalog: Sendinblue

Schema: Reseller

Label: Child Domains by Child ID

Retrieve: true

Insert: true

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ChildDomainsByChildId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

The columns of the table function ChildDomainsByChildId 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
active	boolean	Active	☐	indicates w hether a domain is verified or not
domain	string	Domain	☐	Sender domain

8.1.6 ChildIpAddressesByChildId: Sendinblue Child IP Addresses by Child ID

Get a child account's details

Catalog: Sendinblue

Schema: Reseller

Label: Child IP Addresses by Child ID

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ChildIpAddressesByChildId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
companyName	string	Company Name	☐	Name of the company
credits_emailCredits	int64	Email Credits	☐	Email credits available for your child
credits_smsCredits	int64	SMS Credits	☐	SMS credits available for your child
email	string	Email Address	☐	Login Email
firstName	string	First Name	☐	First Name
lastName	string	Last Name	☐	Last Name
password	string	Password	☐	The encrypted password of child account
statistics_currentMonthTotalSent	int64	#Current Month Sent	☐	Overall emails sent for current month
statistics_previousMonthTotalSent	int64	#Previous Month Sent	☐	Overall emails sent for the previous month
statistics_totalSent	int64	#Sent	☐	Overall emails sent for since the account exists
value	string	Value	☐	IP(s) associated to the user

8.1.7 ResellerChilds: Sendinblue Reseller Childs

Get the list of all children accounts

Catalog: Sendinblue

Schema: Reseller

Primary Keys: id

Label: Reseller Childs

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
companyName	string	Company Name	☐	Name of the company
credits_emailCredits	int64	Email Credits	☐	Email credits available for your child
credits_smsCredits	int64	SMS Credits	☐	SMS credits available for your child
email	string	Email Address	☐	Login Email
firstName	string	First Name	☐	First Name
id	int64	ID	☐	ID of the child
lastName	string	Last Name	☐	Last Name
password	string	Password	☐	The encrypted password of child account
statistics_currentMonthTotalSent	int64	#Current Month Sent	☐	Overall emails sent for current month
statistics_previousMonthTotalSent	int64	#Previous Month Sent	☐	Overall emails sent for the previous month
statistics_totalSent	int64	#Sent	☐	Overall emails sent for since the account exists

8.1.8 SsoTokenByChildId: Sendinblue SSO Token by Child ID

Get session token to access Sendinblue (SSO)It returns a session [token] which will remain valid for a short period of time. A child account will be able to access a white-labeled section by using the following url pattern => https:/email.mydomain.com/login/sso?token=[token]

Catalog: Sendinblue

Schema: Reseller

Label: SSO Token by Child ID

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function SsoTokenByChildId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will 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
childIdentifier	string	☑		Either auth key or id of reseller's child

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
token	string	Token	☐	Session token. It w ill remain valid for a short period of time only.

9 Schema: Senders

9.1 Tables

9.1.1 IpAddresses: Sendinblue IP Addresses

Get all the dedicated IPs for your account

Catalog: Sendinblue

Schema: Senders

Primary Keys: id

Label: IP Addresses

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
active	boolean	Active	☐	Status of the IP (true=active, false=inactive)
domain	string	Domain	☐	Domain associated to the IP
id	int64	ID	☐	ID of the dedicated IP
ip	string	IP	☐	Dedicated IP

9.1.2 IpAddressesBySenderId: Sendinblue IP Addresses by Send ID

Get all the dedicated IPs for a sender

Catalog: Sendinblue

Schema: Senders

Primary Keys: id

Label: IP Addresses by Send ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
senderId	int64	☑		Id of the sender

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
domain	string	Domain	☐	Domain associated to the IP
id	int64	ID	☐	ID of the dedicated IP
ip	string	IP	☐	Dedicated IP
w eight	int64	Weight	☐	Weight of the IP

9.1.3 SenderIpAddresses: Sendinblue Sender IP Addresses

Get the list of all your senders

Catalog: Sendinblue

Schema: Senders

Label: Sender IP Addresses

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
domain	string	☐		Filter your senders for a specific domain
ip	string	☐		Filter your senders for a specific ip (available for dedicated IP usage only)

Columns of Table Function

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

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

9.1.4 Senders: Sendinblue Senders

Get the list of all your senders

Catalog: Sendinblue

Schema: Senders

Primary Keys: id

Label: Senders

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
domain	string	☐		Filter your senders for a specific domain
ip	string	☐		Filter your senders for a specific ip (available for dedicated IP usage only)

Columns of Table Function

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

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

10 Schema: SMSCampaigns

10.1 Tables

10.1.1 SmsCampaignById: Sendinblue SMS Campaign by ID

Get an SMS campaign

Catalog: Sendinblue

Schema: SMSCampaigns

Primary Keys: id

Label: SMS Campaign by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
campaignId	int64	☑		id of the SMS campaign

Columns of Table Function

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

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

10.1.2 SmsCampaignRecipientExclusionLists: Sendinblue SMS Campaign Recipient Exclusion Lists

Returns the information for all your created SMS campaigns

Catalog: Sendinblue

Schema: SMSCampaigns

Label: SMS Campaign Recipient Exclusion Lists

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	datetime	☐		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
startDate	datetime	☐		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
status	string	☐		Status of campaign. (Waarden: suspended, archive, sent, queued, draft, inProcess)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	string	Content	☐	Content of the SMS Campaign
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	☐	ID of the SMS Campaign
lists	string		☐	
modifiedAt	datetime	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	☐	Name of the SMS Campaign
scheduledAt	datetime	Schedule Date	☐	UTC date-time on w hich SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
sender	string	Sender	☐	Sender of the SMS Campaign
statistics_answ ered	int64	#Answ ered	☐	Number of replies to the SMS
statistics_delivered	int64	#Delivered	☐	Number of delivered SMS
statistics_hardBounces	int64	#Hard-bounces	☐	Number of hardbounced SMS
statistics_processing	int64	#Processing	☐	Number of processing SMS
statistics_sent	int64	#Sent	☐	Number of sent SMS
statistics_softBounces	int64	#Soft-bounces	☐	Number of softbounced SMS
statistics_unsubscriptions	int64	#Unsubscriptions	☐	Number of unsubscription SMS
status	string	Status	☐	Status of the SMS Campaign

10.1.3 SmsCampaignRecipientLists: Sendinblue SMS Campaign Recipient Lists

Returns the information for all your created SMS campaigns

Catalog: Sendinblue

Schema: SMSCampaigns

Label: SMS Campaign Recipient Lists

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

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	datetime	☐		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
startDate	datetime	☐		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
status	string	☐		Status of campaign. (Waarden: suspended, archive, sent, queued, draft, inProcess)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	string	Content	☐	Content of the SMS Campaign
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
exclusionLists	string		☐	
id	int64	ID	☐	ID of the SMS Campaign
modifiedAt	datetime	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	☐	Name of the SMS Campaign
scheduledAt	datetime	Schedule Date	☐	UTC date-time on w hich SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
sender	string	Sender	☐	Sender of the SMS Campaign
statistics_answ ered	int64	#Answ ered	☐	Number of replies to the SMS

Name	Data Type	Label	Required	Documentation
statistics_delivered	int64	#Delivered	□	Number of delivered SMS
statistics_hardBounces	int64	#Hard-bounces	□	Number of hardbounced SMS
statistics_processing	int64	#Processing	□	Number of processing SMS
statistics_sent	int64	#Sent	□	Number of sent SMS
statistics_softBounces	int64	#Soft-bounces	□	Number of softbounced SMS
statistics_unsubscriptions	int64	#Unsubscriptions	□	Number of unsubscription SMS
status	string	Status	□	Status of the SMS Campaign

10.1.4 SmsCampaigns: Sendinblue SMS Campaigns

Returns the information for all your created SMS campaigns

Catalog: Sendinblue

Schema: SMSCampaigns

Primary Keys: id

Label: SMS Campaigns

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	datetime	□		Mandatory if startDate is used. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
startDate	datetime	□		Mandatory if endDate is used. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the

Name	Data Type	Required	Default Value	Documentation
				sent sms campaigns. Prefer to pass your timezone in date-time format for accurate result (only available if either 'status' not passed and if passed is set to 'sent')
status	string	☐		Status of campaign. (Waarden: suspended, archive, sent, queued, draft, inProcess)

Columns of Table Function

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

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

11 Schema: SMTP

11.1 Tables

11.1.1 BlockedDomains: Sendinblue Blocked Domains

Get the list of blocked domainsGet the list of blocked domains

Catalog: Sendinblue

Schema: SMTP

Label: Blocked Domains

Retrieve: true

Insert: true

Update: false

Delete: false

Table Columns

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

Name	Data Type	Label	Required	Documentation
TEXT	string	Text	☐	name of blocked domain

11.1.2 EmailEventReport: Sendinblue Email Event Report

Get all your transactional email activity (unaggregated events)

Catalog: Sendinblue

Schema: SMTP

Label: Email Event Report

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

Parameters of Table Function

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

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

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

default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
days	int64	☐		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
email	string	☐		Filter the report for a specific email addresses
endDate	string	☐		Mandatory if startDate is used. Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
event	string	☐		Filter the report for a specific event type (Waarden: bounces, hardBounces, softBounces, delivered, spam, requests, opened, clicks, invalid, deferred, blocked, unsubscribed)
messageId	string	☐		Filter on a specific message id
startDate	string	☐		Mandatory if endDate is used. Starting date of the report (YYYY-MM-DD). Must be lower than equal to endDate
tags	string	☐		Filter the report for tags (serialized and urlencoded array)
templateId	int64	☐		Filter on a specific template id

Columns of Table Function

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

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

Name	Data Type	Label	Required	Documentation
				or softbounce)
subject	string	Subject	□	Subject of the event
tag	string	Tag	□	Tag of the email w hich gener-ated the event

11.1.3 SmtpReport: Sendinblue SMTP Report

Get your transactional email activity aggregated per day

Catalog: Sendinblue

Schema: SMTP

Label: SMTP Report

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
days	int64	□		Number of days in the past in-cluding today (positive integer). Not compatible w ith 'startDate' and 'endDate'
endDate	string	□		Mandatory if startDate is used. Ending date of the report (YYYY-MM-DD)
startDate	string	□		Mandatory if endDate is used. Starting date of the report (YYYY-MM-DD)
tag	string	□		Tag of the emails

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
blocked	int64	Blocked	☐	Number of blocked emails for the date
clicks	int64	Clicks	☐	Number of clicks for the date
date	datetime	Date	☐	Date of the statistics
delivered	int64	Delivered	☐	Number of delivered emails for the date
hardBounces	int64	Hard-bounces	☐	Number of hardbounces for the date
invalid	int64	Invalid	☐	Number of invalid emails for the date
opens	int64	Opens	☐	Number of openings for the date
requests	int64	Requests	☐	Number of requests for the date
softBounces	int64	Soft-bounces	☐	Number of softbounces for the date
spamReports	int64	Spam Reports	☐	Number of complaints (spam reports) for the date
uniqueClicks	int64	#Unique Clicks	☐	Number of unique clicks for the date
uniqueOpens	int64	#Unique Opens	☐	Number of unique openings for the date
unsubscribed	int64	Unsubscribed	☐	Number of unsubscribed emails for the date

11.1.4 SmtpReports: Sendinblue SMTP Reports

Get your transactional email activity aggregated over a period of time

Catalog: Sendinblue

Schema: SMTP

Label: SMTP Reports

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
days	int64	☐		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
endDate	string	☐		Mandatory if startDate is used. Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
startDate	string	☐		Mandatory if endDate is used. Starting date of the report (YYYY-MM-DD). Must be lower than equal to endDate
tag	string	☐		Tag of the emails

Columns of Table Function

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

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

11.1.5 SmtpTemplateById: Sendinblue SMTP Template by ID

Returns the template information

Catalog: Sendinblue

Schema: SMTP

Primary Keys: id

Label: SMTP Template by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
templateId	int64	☑		id of the template

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
doiTemplate	boolean	DOI Template	☐	It is true if template is a valid Double opt-in (DOI) template, otherwise it is false. This field will be available only in case of single template detail call.
htmlContent	string	HTML Content	☐	HTML content of the template
id	int64	ID	☐	ID of the template

Name	Data Type	Label	Required	Documentation
isActive	boolean	Is Active	☐	Status of template (true=active, false=inactive)
modifiedAt	datetime	Modified Date	☐	Last modification UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	☐	Name of the template
replyTo	string	Reply-to	☐	Email defined as the "Reply to" for the template
sender_email	string	Sender Email Address	☐	From email for the template
sender_id	string	Sender ID	☐	Sender id of the template
sender_name	string	Sender Name	☐	From email for the template
subject	string	Subject	☐	Subject of the template
tag	string	Tag	☐	Tag of the template
testSent	boolean	Test Sent	☐	Status of test sending for the template (true=test email has been sent, false=test email has not been sent)
toField	string	To Field	☐	Customisation of the "to" field for the template

11.1.6 SmtpTemplates: Sendinblue SMTP Templates

Get the list of email templates

Catalog: Sendinblue

Schema: SMTP

Primary Keys: id

Label: SMTP Templates

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
templateStatus	boolean	☐		Filter on the status of the template. Active = true, inactive = false

Columns of Table Function

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

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

11.1.7 TransacBlockedContacts: Sendinblue Transaction Blocked Contacts

Get the list of blocked or unsubscribed transactional contacts

Catalog: Sendinblue

Schema: SMTP

Primary Keys: email

Label: Transaction Blocked Contacts

Retrieve: true

Insert: false

Update: false

Delete: true

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	string	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) till which you want to fetch the blocked or unsubscribed contacts
startDate	string	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) from which you want to fetch the blocked or unsubscribed contacts

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
blockedAt	datetime	Blocked at	☐	Date when the contact was blocked or unsubscribed on
email	string	Email Address	☐	Email address of the blocked or unsubscribed contact
reason_code	string	Reason Code	☐	Reason code for blocking / unsubscribing (This code is safe for comparison)
reason_message	string	Reason Message	☐	Reason for blocking / unsubscribing (This string is not safe for comparison)

Name	Data Type	Label	Required	Documentation
senderEmail	string	Sender Email	☐	Sender email address of the blocked or unsubscribed contact

11.1.8 TransacEmailContent_Events

Get the personalized content of a sent transactional email

Catalog: Sendinblue

Schema: SMTP

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
uuid	string	☑		Unique id of the transactional email that has been sent to a particular contact

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attachmentCount	int64	Attachment Count	☐	Count of the attachments that were sent in the email
body	string	Body	☐	Actual content of the transactional email that has been sent
date	datetime	Date	☐	Date on which transactional email was sent
email	string	Email Address	☐	Email address to which transactional email has been sent

Name	Data Type	Label	Required	Documentation
name	string	Name	☐	Name of the event that occurred on the sent email
subject	string	Subject	☐	Subject of the sent email
templateId	int64	Template ID	☐	Id of the template
time	datetime	Time	☐	Time at w hich the event occurred

11.1.9 TransacEmailContent: Sendinblue Transaction Email Contents

Get the personalized content of a sent transactional email

Catalog: Sendinblue

Schema: SMTP

Label: Transaction Email Contents

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
uuid	string	☑		Unique id of the transactional email that has been sent to a particular contact

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attachmentCount	int64	Attachment Count	☐	Count of the attachments that w ere sent in the email

Name	Data Type	Label	Required	Documentation
body	string	Body	☐	Actual content of the transactional email that has been sent
date	datetime	Date	☐	Date on which transactional email was sent
email	string	Email Address	☐	Email address to which transactional email has been sent
subject	string	Subject	☐	Subject of the sent email
templateId	int64	Template ID	☐	Id of the template

11.1.10 TransacEmailsList_TransactionalEmailsTags

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

Catalog: Sendinblue

Schema: SMTP

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☐		Mandatory if templateId and messageId are not passed in query filters. Email address to which transactional email has been sent.
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) till which you want to fetch the list. Maximum time period that can be selected is one month.
messageId	string	☐		Mandatory if templateId and email are not passed in query filters.

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

Columns of Table Function

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

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

11.1.11 TransacEmailsList: Sendinblue Transaction Email Lists

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

Catalog: Sendinblue

Schema: SMTP

Label: Transaction Email Lists

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TransacEmailsList. A value must be provided at all times for required parameters, but op-

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

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

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

Name	Data Type	Required	Default Value	Documentation
email	string	☐		Mandatory if templateId and messageId are not passed in query filters. Email address to which transactional email has been sent.
endDate	datetime	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) till which you want to fetch the list. Maximum time period that can be selected is one month.
messageId	string	☐		Mandatory if templateId and email are not passed in query filters. Message ID of the transactional email sent.
startDate	datetime	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) from which you want to fetch the list. Maximum time period that can be selected is one month.
templateId	int64	☐		Mandatory if email and messageId are not passed in query filters. Id of the template that was used to compose transactional email.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	☐	Date on which transactional email was sent
email	string	Email Address	☐	Email address to which transactional email has been sent
from	string	From	☐	Email address of the sender from which the email was sent
messageId	string	Message ID	☐	Message Id of the sent email
subject	string	Subject	☐	Subject of the sent email

Name	Data Type	Label	Required	Documentation
templateId	int64	Template ID	☐	Id of the template
uuid	string	UUID	☐	Unique id of the email sent to a particular contact

12 Schema: TransactionalSMS

12.1 Tables

12.1.1 SmsEvents: Sendinblue SMS Events

Get all your SMS activity (unaggregated events)

Catalog: Sendinblue

Schema: TransactionalSMS

Label: SMS Events

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
days	int64	☐		Number of days in the past including today (positive integer). Not compatible w ith 'startDate' and 'endDate'
endDate	string	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the report
event	string	☐		Filter the report for specific events (Waarden: bounces, hardBounces, softBounces, delivered, sent, accepted, unsubscription, replies, blocked, rejected)
phoneNumber	string	☐		Filter the report for a specific phone number

Name	Data Type	Required	Default Value	Documentation
startDate	string	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the report
tags	string	☐		Filter the report for specific tags passed as a serialized urlencoded array

Columns of Table Function

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

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

12.1.2 TransacSmsReport: Sendinblue SMS Transactions Report

Get your SMS activity aggregated per day

Catalog: Sendinblue

Schema: TransactionalSMS

Label: SMS Transactions Report

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

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
days	int64	☐		Number of days in the past including today (positive integer). Not compatible with 'startDate' and 'endDate'
endDate	string	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the report
startDate	string	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the report
tag	string	☐		Filter on a tag

Columns of Table Function

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

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

12.1.3 TransactionAggregatedSmsReports: Sendinblue SMS Aggregated Transactions Report

Get your SMS activity aggregated over a period of time

Catalog: Sendinblue

Schema: TransactionalSMS

Label: SMS Aggregated Transactions Report

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
days	int64	☐		Number of days in the past including today (positive integer). Not compatible w ith startDate and endDate
endDate	string	☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD) of the report
startDate	string	☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD) of the report
tag	string	☐		Filter on a tag

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
accepted	int64	Accepted	☐	Number of accepted for the timeframe
blocked	int64	Blocked	☐	Number of blocked contact for the timeframe
delivered	int64	Delivered	☐	Number of delivered SMS for the timeframe
hardBounces	int64	Hard-bounces	☐	Number of hardbounces for the timeframe
range	string	Range	☐	Time frame of the report
rejected	int64	Rejected	☐	Number of rejected for the timeframe

Name	Data Type	Label	Required	Documentation
replied	int64	Replied	☐	Number of answ ered SMS for the timeframe
requests	int64	Requests	☐	Number of requests for the time-frame
softBounces	int64	Soft-bounces	☐	Number of softbounces for the timeframe
unsubscribed	int64	Unsubscribed	☐	Number of unsubscription for the timeframe

13 Schema: Views

13.1 Views

13.1.1 AbTestCampaignResults: Sendinblue A/B Test Campaign Results

Catalog: Sendinblue

Schema: Views

Label: A/B Test Campaign Results

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

View Columns

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

Name	Data Type	Label	Required	Documentation
CampaignId	int64	ID	☐	ID of the campaign
CampaignName	string	Name	☐	Name of the campaign
CampaignSubject	string	Subject	☐	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
clickRate	string	Clickrate	☐	Click rate for current w inning version
openRate	string	Open Rate	☐	Open rate for current w inning version
statistics_clicks_Version_A	string		☐	percentage of an event for version A
statistics_clicks_Version_B	string		☐	percentage of an event for version B
statistics_complaints_Version_A	string		☐	percentage of an event for version A
statistics_complaints_Version_B	string		☐	percentage of an event for version B
statistics_hardBounces_Version_A	string		☐	percentage of an event for version A
statistics_hardBounces_Version_B	string		☐	percentage of an event for version B

Name	Data Type	Label	Required	Documentation
statistics_openers_Version_A	string		☐	percentage of an event for version A
statistics_openers_Version_B	string		☐	percentage of an event for version B
statistics_softBounces_Version_A	string		☐	percentage of an event for version A
statistics_softBounces_Version_B	string		☐	percentage of an event for version B
statistics_unsubscribed_Version_A	string		☐	percentage of an event for version A
statistics_unsubscribed_Version_B	string		☐	percentage of an event for version B
winningCriteria	string	Winning Criteria	☐	Criteria choosen for winning version (Open/Click)
winningSubjectLine	string	Winning Subject Line	☐	Subject Line of current winning version
winningVersion	string	Winning Version	☐	Winning Campaign Info. pending = Campaign has been picked for sending and winning version is yet to be decided, tie = A tie happened between both the versions, notAvailable = Campaign has not yet been picked for sending.
winningVersionRate	string	Winning Version Rate	☐	Open/Click rate for the winner version

13.1.2 FolderLists: Sendinblue Folder Lists

Catalog: Sendinblue

Schema: Views

Label: Folder Lists

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

View Columns

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

Name	Data Type	Label	Required	Documentation
Folderid	int64	ID	☐	ID of the folder
Foldername	string	Name	☐	Name of the folder
FoldertotalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the folder
FoldertotalSubscribers	int64	Total Subscribers	☐	Number of contacts in the folder
FolderuniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the folder
id	int64	ID	☐	ID of the list

Name	Data Type	Label	Required	Documentation
name	string	Name	☐	Name of the list
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the list

13.1.3 ListCampaignStatistics: Sendinblue List Campaign Statistics

Catalog: Sendinblue

Schema: Views

Label: List Campaign Statistics

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

View Columns

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

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign
createdAt	datetime	Creation Date	☐	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	boolean	Dynamic List	☐	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)
folderId	int64	Folder ID	☐	ID of the folder
id	int64	ID	☐	ID of the list
name	string	Name	☐	Name of the list
stats_clickers	int64	#Clickers	☐	Number of total clicks for the campaign
stats_complaints	int64	#Complaints	☐	Number of complaints (Spam reports) for the campaign
stats_deferred	int64	#Deferred	☐	Number of deferred emails for the campaign
stats_delivered	int64	#Delivered	☐	Number of delivered emails for the campaign
stats_hardBounces	int64	#Hard-bounces	☐	Number of harbounce for the campaign
stats_listId	int64	List ID	☐	List Id of email campaign (only in case of get email campaign(s) (not for global stats))
stats_returnBounce	int64	#Return-bounces	☐	Total number of non-delivered campaigns for a particular campaign id.
stats_sent	int64	#Sent	☐	Number of sent emails for the campaign

Name	Data Type	Label	Required	Documentation
stats_softBounces	int64	#Soft-bounces	☐	Number of softbounce for the campaign
stats_trackableViews	int64	#Trackable Views	☐	Recipients without any privacy protection option enabled in their email client
stats_uniqueClicks	int64	#Unique Clicks	☐	Number of unique clicks for the campaign
stats_uniqueViews	int64	#Unique Views	☐	Number of unique openings for the campaign
stats_unsubscriptions	int64	#Unsubscriptions	☐	Number of unsubscription for the campaign
stats_viewed	int64	#Viewed	☐	Number of openings for the campaign
totalBlacklisted	int64	Total Blacklisted	☐	Number of blacklisted contacts in the list
totalSubscribers	int64	Total Subscribers	☐	Number of contacts in the list

14 Schema: Webhooks

14.1 Tables

14.1.1 WebhookById: Sendinblue Webhook by ID

Get a webhook details

Catalog: Sendinblue

Schema: Webhooks

Primary Keys: id

Label: Webhook by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webhookId	int64	☑		Id of the webhook

Columns of Table Function

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

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

14.1.2 WebhookEventsByWebhookId

Get a webhook details

Catalog: Sendinblue

Schema: Webhooks

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webhookId	int64	☑		Id of the webhook

Columns of Table Function

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

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

14.1.3 Webhooks: Sendinblue Webhooks

Get all webhooks

Catalog: Sendinblue

Schema: Webhooks

Primary Keys: id

Label: Webhooks

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
type	string	☐	transactional	Filter on webhook type (Waarden: marketing, transactional)

Columns of Table Function

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

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

15 Schema: WhatsAppCampaigns

15.1 Tables

15.1.1 WhatsAppCampaignById: Sendinblue WhatsApp Campaign by ID

Get a WhatsApp campaign

Catalog: Sendinblue

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp Campaign by ID

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

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
campaignId	int64	☑		Id of the campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignName	string		☐	Name of the WhatsApp Campaign
campaignStatus	string		☐	Status of the WhatsApp Campaign
createdAt	string	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	☐	ID of the WhatsApp Campaign
modifiedAt	string	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
scheduledAt	string	Schedule Date	☐	UTC date-time on which WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
senderNumber	string		☐	Sender of the WhatsApp Campaign
stats_delivered	int64	#Delivered	☐	
stats_notSent	int64		☐	
stats_read	int64		☐	
stats_sent	int64	#Sent	☐	
stats_unsubscribe	int64		☐	
template_button_type	string		☐	
template_category	string		☐	description of the template
template_contains_button	boolean		☐	
template_display_header	boolean		☐	
template_header_type	string		☐	type of header
template_hide_footer	boolean		☐	
template_language	string		☐	language of the template
template_name	string		☐	name of the template

15.1.2 WhatsAppCampaigns: Sendinblue WhatsApp Campaigns

Return all your created WhatsApp campaigns

Catalog: Sendinblue

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp Campaigns

Retrieve: true

Insert: false

Update: false

Delete: true

Parameters of Table Function

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

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

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

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

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignName	string		☐	Name of the WhatsApp Campaign
campaignStatus	string		☐	Status of the WhatsApp Campaign
createdAt	string	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
errorReason	string		☐	Error reason in the campaign creation
id	int64	ID	☐	ID of the WhatsApp Campaign
invalidatedContacts	int64		☐	Count of invalidated contacts
modifiedAt	string	Modified Date	☐	UTC date-time of last modification of the w hatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
readPercentage	float		☐	Read percentage of the the WhatsApp campaign created
scheduledAt	string	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
stats_delivered	int64	#Delivered	☐	
stats_notSent	int64		☐	
stats_read	int64		☐	
stats_sent	int64	#Sent	☐	
stats_unsubscribe	int64		☐	
templateId	string	Template ID	☐	Id of the WhatsApp template

15.1.3 WhatsAppTemplates: Sendinblue WhatsApp Templates

Return all your created WhatsApp templates

Catalog: Sendinblue

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp Templates

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate	string	☐		**Mandatory if startDate is used**. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the templates created.**Prefer to pass your timezone in date-time format for accurate result**
startDate	string	☐		**Mandatory if endDate is used**. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the templates created.**Prefer to pass your timezone in date-time format for accurate result**

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
category	string	Category	☐	category of the template
createdAt	string	Creation Date	☐	Creation UTC date-time of the w hatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
errorReason	string		☐	Error reason in the template creation
id	string	ID	☐	id of the template
language	string		☐	Language in w hich template exists
modifiedAt	string	Modified Date	☐	UTC date-time of last modification of the w hatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	string	Name	☐	Name of the WhatsApp template
status	string	Status	☐	Status of the WhatsApp template

Index

- # -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- 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
24/08/2024, 20:35
ISBN 978-94-6302-000-8: