Sendinblue API Data Model

for use with Invantive SQL

24.0

invantive
the SQL company

Copyright

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Important Safety and Usage Information

Intended Use and Limitations: This software, developed by Invantive, is designed to support a variety of business and information technology data processing functions, such as accounting, financial reporting 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 1

2 {res:itgen_doc_sql_attr_api} 2

3 Catalog: Brevo 17

3.1 Schemas ... 17
3.1.1 Schema: Account ... 17
3.1.2 Schema: Balance ... 20
3.1.3 Schema: Companies ... 33
3.1.4 Schema: Contacts ... 41
3.1.5 Schema: Conversations ... 93
3.1.6 Schema: Coupons ... 117
3.1.7 Schema: CustomObjects ... 120
3.1.8 Schema: Deals ... 123
3.1.9 Schema: Domains ... 134
3.1.10 Schema: Ecommerce ... 136
3.1.11 Schema: EmailCampaigns ... 151
3.1.12 Schema: Event ... 161
3.1.13 Schema: ExternalFeeds ... 162
3.1.14 Schema: Files ... 168
3.1.15 Schema: InboundParsing ... 171
3.1.16 Schema: MasterAccount ... 176
3.1.17 Schema: Native ... 202
3.1.18 Schema: Notes ... 204
3.1.19 Schema: Payments ... 213
3.1.20 Schema: Process ... 214
3.1.21 Schema: Program ... 217
3.1.22 Schema: Reward ... 224
3.1.23 Schema: Senders ... 241
3.1.24 Schema: SMSCampaigns ... 245
3.1.25 Schema: SMSTemplates ... 252
3.1.26 Schema: Tasks ... 253
3.1.27 Schema: Tier ... 264
3.1.28 Schema: TransactionalEmails ... 273
3.1.29 Schema: TransactionalSMS ... 289
3.1.30 Schema: TransactionalWhatsApp ... 294
3.1.31 Schema: User ... 295
3.1.32 Schema: Views ... 299
3.1.33 Schema: Webhooks ... 302
3.1.34 Schema: WhatsAppCampaigns ... 307

4 Package: dcr_metadata 323

4.1 Procedures ... 323
4.1.1 dcr_metadata.get_partitions: Sendinblue Data container metadata package ... 323

5 Package: sib_balance 323

5.1 Procedures ... 323
5.1.1 sib_balance.begintransaction: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.2 sib_balance.canceltransaction: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.3 sib_balance.completetransaction: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.4 sib_balance.createbalancelimit: Sendinblue Package for Balance operations that could not be mapped to a table.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

5.1.5 sib_balance.createbalanceorder: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.6 sib_balance.deletebalancedefinition: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.7 sib_balance.deletebalancelimit: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.8 sib_balance.updatebalancedefinition: Sendinblue Package for Balance operations that could not be mapped to a table.
5.1.9 sib_balance.updatebalancelimit: Sendinblue Package for Balance operations that could not be mapped to a table.

6 Package: sib_contacts 330

6.1 Procedures 330
6.1.1 sib_contacts.addcontacttolist: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.2 sib_contacts.createattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.3 sib_contacts.createdoicontact: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.4 sib_contacts.deleteattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.5 sib_contacts.deletemultiattributeoptions: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.6 sib_contacts.importcontacts: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.7 sib_contacts.removecontactfromlist: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.8 sib_contacts.requestcontactexport: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.9 sib_contacts.updateattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.
6.1.10 sib_contacts.updatebatchcontacts: Sendinblue Package for Contacts operations that could not be mapped to a table.

7 Package: sib_coupons 338

7.1 Procedures 338
7.1.1 sib_coupons.createcoupons: Sendinblue Package for Coupons operations that could not be mapped to a table.
7.1.2 sib_coupons.updatecouponcollection: Sendinblue Package for Coupons operations that could not be mapped to a table.

8 Package: sib_custom_objects 339

8.1 Procedures 339
8.1.1 sib_custom_objects.batchdeleteobjectrecords: Sendinblue Package for Custom Objects operations that could not be mapped to a tabl
8.1.2 sib_custom_objects.upsertrecords: Sendinblue Package for Custom Objects operations that could not be mapped to a table.

9 Package: sib_domains 341

9.1 Procedures 341
9.1.1 sib_domains.authenticatedomain: Sendinblue Package for Domains operations that could not be mapped to a table.
9.1.2 sib_domains.deletedomain: Sendinblue Package for Domains operations that could not be mapped to a table.

10 Package: sib_ecommerce 342

10.1 Procedures 342
10.1.1 sib_ecommerce.createbatchorder: Sendinblue Package for Ecommerce operations that could not be mapped to a table.
10.1.2 sib_ecommerce.createorder: Sendinblue Package for Ecommerce operations that could not be mapped to a table.
10.1.3 sib_ecommerce.createproductalert: Sendinblue Package for Ecommerce operations that could not be mapped to a table.
10.1.4 sib_ecommerce.createupdatebatchcategory: Sendinblue Package for Ecommerce operations that could not be mapped to a table.
10.1.5 sib_ecommerce.createupdatebatchproducts: Sendinblue Package for Ecommerce operations that could not be mapped to a table.

11 Package: sib_email_campaigns 346

11.1 Procedures 346
11.1.1 sib_email_campaigns.emailexportrecipients: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.
11.1.2 sib_email_campaigns.sendemailcampaignnow : Sendinblue Package for Email Campaigns operations that could not be mapped to a tab
11.1.3 sib_email_campaigns.sendreport: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.
11.1.4 sib_email_campaigns.sendtestemail: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.
11.1.5 sib_email_campaigns.updatecampaignstatus: Sendinblue Package for Email Campaigns operations that could not be mapped to a table
11.1.6 sib_email_campaigns.uploadimagetogallery: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

12 Package: sib_event 350

12.1 Procedures 350
12.1.1 sib_event.createbatchevents: Sendinblue Package for Event operations that could not be mapped to a table.

13 Package: sib_master_account 350

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

13.1 Procedures 350
13.1.1 sib_master_account.inviteadminuser: Sendinblue Package for Master account operations that could not be mapped to a table.

14 Package: sib_payments 351

14.1 Procedures 351
14.1.1 sib_payments.createpaymentrequest: Sendinblue Package for Payments operations that could not be mapped to a table.
14.1.2 sib_payments.deletepaymentrequest: Sendinblue Package for Payments operations that could not be mapped to a table.

15 Package: sib_program 352

15.1 Procedures 352
15.1.1 sib_program.deletecontactmembers: Sendinblue Package for Program operations that could not be mapped to a table.
15.1.2 sib_program.deletecontactsubscription: Sendinblue Package for Program operations that could not be mapped to a table.
15.1.3 sib_program.partiallyupdateloyaltyprogram: Sendinblue Package for Program operations that could not be mapped to a table.
15.1.4 sib_program.publishloyaltyprogram: Sendinblue Package for Program operations that could not be mapped to a table.
15.1.5 sib_program.subscribemembertoasubscription: Sendinblue Package for Program operations that could not be mapped to a table.
15.1.6 sib_program.subscribetoloyaltyprogram: Sendinblue Package for Program operations that could not be mapped to a table.

16 Package: sib_reward 355

16.1 Procedures 355
16.1.1 sib_rew ard.completeredeemtransaction: Sendinblue Package for Rew ard operations that could not be mapped to a table.
16.1.2 sib_rew ard.createrew ard: Sendinblue Package for Rew ard operations that could not be mapped to a table.
16.1.3 sib_rew ard.createvoucher: Sendinblue Package for Rew ard operations that could not be mapped to a table.
16.1.4 sib_rew ard.redeemvoucher: Sendinblue Package for Rew ard operations that could not be mapped to a table.
16.1.5 sib_rew ard.revokevouchers: Sendinblue Package for Rew ard operations that could not be mapped to a table.
16.1.6 sib_rew ard.validaterew ard: Sendinblue Package for Rew ard operations that could not be mapped to a table.

17 Package: sib_sales 359

17.1 Procedures 359
17.1.1 sib_sales.insert_note: Sendinblue Brevo sales 359

18 Package: sib_senders 360

18.1 Procedures 360
18.1.1 sib_senders.validatesenderbyotp: Sendinblue Package for Senders operations that could not be mapped to a table.

19 Package: sib_sms_campaigns 360

19.1 Procedures 360
19.1.1 sib_sms_campaigns.requestsmsrecipientexport: Sendinblue Package for SMS Campaigns operations that could not be mapped to a ta
19.1.2 sib_sms_campaigns.sendsmscampaignnow : Sendinblue Package for SMS Campaigns operations that could not be mapped to a table
19.1.3 sib_sms_campaigns.sendsmsreport: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.
19.1.4 sib_sms_campaigns.sendtestsms: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.
19.1.5 sib_sms_campaigns.updatesmscampaignstatus: Sendinblue Package for SMS Campaigns operations that could not be mapped to a ta

20 Package: sib_tier 364

20.1 Procedures 364
20.1.1 sib_tier.addsubscriptiontotier: Sendinblue Package for Tier operations that could not be mapped to a table.
20.1.2 sib_tier.createtierfortiergroup: Sendinblue Package for Tier operations that could not be mapped to a table.
20.1.3 sib_tier.createtiergroup: Sendinblue Package for Tier operations that could not be mapped to a table. .. 365
20.1.4 sib_tier.deletetier: Sendinblue Package for Tier operations that could not be mapped to a table. 366
20.1.5 sib_tier.deletetiergroup: Sendinblue Package for Tier operations that could not be mapped to a table. ... 366
20.1.6 sib_tier.updatetier: Sendinblue Package for Tier operations that could not be mapped to a table. 367
20.1.7 sib_tier.updatetiergroup: Sendinblue Package for Tier operations that could not be mapped to a table. . 367

21 Package: sib_tracker 368

21.1 Procedures 368
21.1.1 sib_tracker.identify: Sendinblue Brevo tracker 368
21.1.2 sib_tracker.track_event: Sendinblue Brevo tracker 369

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

21.1.3 sib_tracker.track_link: Sendinblue Brevo tracker 370
21.1.4 sib_tracker.track_page: Sendinblue Brevo tracker 370

22 Package: sib_transactional_emails 371

22.1 Procedures 371
22.1.1 sib_transactional_emails.deleteblockeddomain: Sendinblue Package for Transactional emails operations that could not be mapped to a
22.1.2 sib_transactional_emails.deletehardbounces: Sendinblue Package for Transactional emails operations that could not be mapped to a t
22.1.3 sib_transactional_emails.deletescheduledemailbyid: Sendinblue Package for Transactional emails operations that could not be mapped
22.1.4 sib_transactional_emails.postpreview smtpemailtemplates: Sendinblue Package for Transactional emails operations that could not be r
22.1.5 sib_transactional_emails.sendtesttemplate: Sendinblue Package for Transactional emails operations that could not be mapped to a tab
22.1.6 sib_transactional_emails.sendtransacemail: Sendinblue Package for Transactional emails operations that could not be mapped to a tal

23 Package: sib_transactional_sms 375

23.1 Procedures 375
23.1.1 sib_transactional_sms.sendasynctransactionalsms: Sendinblue Package for Transactional SMS operations that could not be mapped
23.1.2 sib_transactional_sms.sendtransacsms: Sendinblue Package for Transactional SMS operations that could not be mapped to a table.

24 Package: sib_transactional_whatsapp 376

24.1 Procedures 376
24.1.1 sib_transactional_w hatsapp.sendw hatsappmessage: Sendinblue Package for Transactional WhatsApp operations that could not be

25 Package: sib_user 376

25.1 Procedures 376
25.1.1 sib_user.edituserpermission: Sendinblue Package for User operations that could not be mapped to a table.
25.1.2 sib_user.inviteuser: Sendinblue Package for User operations that could not be mapped to a table. 377
25.1.3 sib_user.putresendcancelinvitation: Sendinblue Package for User operations that could not be mapped to a table.
25.1.4 sib_user.putrevokeuserpermission: Sendinblue Package for User operations that could not be mapped to a table.

26 Package: sib_webhooks 379

26.1 Procedures 379
26.1.1 sib_w ebhooks.exportw ebhookshistory: Sendinblue Package for Webhooks operations that could not be mapped to a table.

27 Package: sib_whatsapp_campaigns 379

27.1 Procedures 379
27.1.1 sib_w hatsapp_campaigns.createw hatsapptemplate: Sendinblue Package for WhatsApp Campaigns operations that could not be map
27.1.2 sib_w hatsapp_campaigns.sendw hatsapptemplateapproval: Sendinblue Package for WhatsApp Campaigns operations that could not

Index 381

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

1 SQL Driver for Sendinblue API

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

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

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

The Sendinblue driver covers 236 tables and 2701 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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`

More technical documentation as provided by the supplier of Sendinblue on the native connection used can be found at https://api.brevo.com/.

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

2 {res:itgen_doc_sql_attr_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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		✓		✓	✓

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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.	Brevo		✓	✓	✓	✓
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	250	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when HTTP server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-403-forbidden-max-tries	Maximum number of tries when the website reports a HTTP status 403 (Forbidden).		2	✓	✓	✓	
download-error-403-forbidden-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 403 (Forbidden).		3000	✓	✓	✓	
download-error-403-forbidden-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 403 (Forbidden).		10000	✓	✓	✓	
download-error-403-forbidden-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 403 (Forbidden).		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	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 408.		300000	✓	✓	✓	
dow nload-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a HTTP status 408.		2	✓	✓	✓	
dow nload-error-422-bad-request-max-tries	Maximum number of tries w hen HTTP server reports unprocessable entity during retrieval of data.		30	✓	✓	✓	
dow nload-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen HTTP server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen HTTP server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep betw een retries HTTP server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
dow nload-error-429-too-many-requests-max-tries	Maximum number of tries w hen the w ebsite reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports that too many requests have been made during a timeslot of one minute or one day.		10000	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports that too many requests have been made during a timeslot of one minute or one day.		300000	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
dow nload-error-500-internal-server-error-max-tries	{res:itgen_pae_dow nload_error_500_internal_server_error_max_tries}		5	✓	✓	✓	
dow nload-error-500-internal-server-error-sleep-initial-ms	{res:itgen_pae_dow nload_error_500_internal_server_error_sleep_initial_ms}		1000	✓	✓	✓	
dow nload-error-500-internal-server-error-sleep-max-ms	{res:itgen_pae_dow nload_error_500_internal_server_error_sleep_max_ms}		5000	✓	✓	✓	
dow nload-error-500-internal-server-	{res:itgen_pae_dow nload_error_500_internal_server_er-		2	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
error-sleep-multiplic-ator	ror_sleep_multiplicator}						
dow nload-error-502-server-unavail-able-max-tries	Maximum number of tries w hen HTTP server reports a bad gatew ay during retrieval of data.		30	✓	✓	✓	
dow nload-error-502-server-unavail-able-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen HTTP server reports a bad gatew ay during retrieval of data.		10000	✓	✓	✓	
dow nload-error-502-server-unavail-able-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen HTTP server reports that a bad gatew ay during retrieval of data.		300000	✓	✓	✓	
dow nload-error-502-server-unavail-able-sleep-multiplic-ator	Multiplication factor for sleep betw een retries 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-	Maximum number of tries w hen the w ebsite reports a HTTP status 590.		10	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
download-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
download-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		10000	✓	✓	✓	
download-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error ocurrs during retrieval of data.		300000	✓	✓	✓	
download-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error ocurrs during retrieval of data.		2	✓	✓	✓	
download-error-socket-exception-max-tries	Maximum number of tries when the network connection is forcible dropped during retrieval of data.		10	✓	✓	✓	
download-error-socket-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		10000	✓	✓	✓	
download-error-socket-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when the network connection is forcible dropped during retrieval of data.		300000	✓	✓	✓	
download-error-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcible dropped during retrieval of data.		2	✓	✓	✓	
download-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-web-not-found-max-tries	{res:itgen_pae_download_error_web_not_found_max_tries}		1	✓	✓	✓	
download-error-web-not-found-sleep-initial-ms	{res:itgen_pae_download_error_web_not_found_sleep_initial_ms}		10000	✓	✓	✓	
download-error-web-not-found-sleep-max-ms	{res:itgen_pae_download_error_web_not_found_sleep_max_ms}		300000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQLStatement	Set from Drivers File	Set from Log On
download-error-web-not-found-sleep-multiplicator	{res:itgen_pae_download_error_web_not_found_sleep_multiplicator}		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-implemented-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports not implemented.		10000	✓	✓	✓	
download-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	
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	✓	✓	✓	
download-error-web-unknown-max-tries	{res:itgen_pae_download_error_web_unknown_max_tries}		10	✓	✓	✓	
download-error-web-unknown-sleep-initial-ms	{res:itgen_pae_download_error_web_unknown_sleep_initial_ms}		5000	✓	✓	✓	
download-error-web-unknown-sleep-max-ms	{res:itgen_pae_download_error_web_unknown_sleep_max_ms}		30000	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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-w eb-unknow n-sleep-multiplicator	{res:itgen_pae_dow nload_er-ror_w eb_unknow n_sleep_multiplic-ator}		2	✓	✓	✓	
duplicate-row -mode	Configure behaviour w hen multiple row s are discovered during a limited search based upon a fingerprint generated from the primary key value, or all column values w hen no primary key is know n.. The follow ing values can be used: "error" to raise error on duplicate fin-gerprint values in an API-page of row s, "keep" to ignore duplicate fin-gerprint values silently, "skip" like "keep" but w ith logging to trace of duplicate values or "null" for default behaviour (typically "error")..			✓	✓	✓	✓
force-case-sensit-ive-identifiers	Consider identifiers as case-sensit-ive independent of the platform cap-abilities.	Shared	False	✓	✓	✓	
forced-casing-identi-fiers	Forced casing of identifiers. Choose from: Unset, Low er, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (in-tense). Default is 5.		5	✓	✓	✓	
http-disk-cache-dir-ectory	Directory w here HTTP cache is stored.		C: \Users\guido\I n-vantive\Cach e\http\guido\s hared	✓	✓	✓	
http-disk-cache-ig-nore-w rite-errors	Whether to ignore w rite errors to disk cache.		False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.		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.		5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.		14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		57000	✓	✓	✓	
http-public-disk-cache-directory	{res:itgen_pae_http_pub-lic_disk_cache_directory}		C: \Users\guido\I	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
			n-vantive\Cache\public				
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
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	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
invantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-correct-invalid-date	Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753.	SQL Engine V1	False	✓	✓	✓	
invantive-sql-execution-profile-disk-path	{res:itgen_pae_invantive_sql_execution_profile_disk_path}	SQL Engine V1		✓	✓	✓	
invantive-sql-execution-profile-to-disk	{res:itgen_pae_invantive_sql_execution_profile_to_disk}	SQL Engine V1	False	✓	✓	✓	
invantive-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of 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		✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
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	✓	✓	✓	
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		✓	✓	✓	
max-odata-filters	Maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-odata-rewrite-in-count	{res:itgen_pae_max_odata_rewrite_in_count}	OData	500	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.		8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.		8000	✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.		2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.		10000	✓	✓	✓	
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.		1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.		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-maximum-usable-partitions	Requested Maximum Usable Partitions	Shared		✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective	Shared		✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
	on limited platforms such as AFAS Online.						
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 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-500-errors-percentage	Percentage of simulated HTTP 500 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-502-errors-percentage	Percentage of simulated HTTP 502 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	✓	✓	✓	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-504-errors	{res:itgen_pae_simulate_http_504_errors}		False	✓	✓	✓	
simulate-http-504-errors-percentage	{res:itgen_pae_simulate_http_504_errors_percentage}		0	✓	✓	✓	
simulate-http-522-errors	{res:itgen_pae_simulate_http_522_errors}		False	✓	✓	✓	
simulate-http-522-errors-percentage	{res:itgen_pae_simulate_http_522_errors_percentage}		0	✓	✓	✓	
simulate-http-524-errors	{res:itgen_pae_simulate_http_524_errors}		False	✓	✓	✓	
simulate-http-524-errors-percentage	{res:itgen_pae_simulate_http_524_errors_percentage}		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol 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	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	60000	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
standardize-identifiers	Rewrite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
standardize-identifiers-casing	Rewrite all identifiers to the recommended standard platform-specific casing when changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
swagger-specification-allow-fallback	Whether to allow fallback in case of an Internet connection issue on a possibly bundled Swagger specification.		True	✓		✓	✓
swagger-specification-download-tries	Specification Download Tries		5	✓	✓	✓	
swagger-specification-file	The Swagger file path, such as C:\temp\swagger.json.			✓	✓	✓	✓

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
swagger-specification-http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of Swagger specification data in the HTTP disk cache.		86400	✓	✓	✓	
swagger-specification-url	The Swagger URL such as https://example.org/rest/swagger.json.			✓	✓	✓	✓
totp-secret	Shared secret key to generate one-time password using TOTP RFC 6238. For improved security, manually enter the one-time password asked during login.	OData		✓		✓	✓
use-batch-insert	Whether to use batch insert.	OData	True	✓	✓	✓	
use-http-disk-cache	Combination of use-http-disk-cache-read and use-http-disk-cache-write.			✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.		False	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.		False	✓	✓	✓	
use-http-memory-cache	Combination of use-http-memory-cache-read and use-http-memory-cache-write.			✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.		True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.		True	✓	✓	✓	

3 Catalog: Brevo

3.1 Schemas

3.1.1 Schema: Account

3.1.1.1 Tables

AccountActivities

Get user activity logsRetrieves user activity logs from your organization for security monitoring and audit compliance.Use this to:- Monitor user login activities and access patterns- Track account modifications and configuration changes- Generate security audit reports and compliance documentation- Investigate suspicious activities and unauthorized access- Monitor team member actions and account usageKey information returned:- Complete user activity details and timestamps- User identification (email, IP address, browser)- Action types and activity descriptions- Security-relevant events and access logs- Historical activity data for audit trailsNote: Requires Enterprise plan for access to organization activity logs.

Catalog: Brevo

Schema: Account

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 AccountActivities. 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		☐		Enter the user's email address to filter their activity in the account.
endDate		☐		Mandatory if startDate is used. Enter end date in UTC date (YYYY-MM-DD)format to filter the activity in your account. Maximum time period thatcan be selected is one month.
startDate		☐		Mandatory if endDate is used. Enter start date in UTC date (YYYY-MM-DD)format to filter the activity in your account. Maximum time period thatcan be selected is one month. Additionally, you can retrieve activitylogs from the past 12 months from the date of your search.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
action	varchar2		☐	Type of activity in the account.
date	varchar2	Date	☐	Time of the activity.
user_agent	varchar2		☐	Browser details of the user who performed the activity.
user_email	varchar2		☐	Email address of the user who performed activity in theaccount.
user_ip	varchar2		☐	IP address of the user who performed activity in theaccount.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

SubscriptionAccount: Sendinblue Subscription Account

Get account detailsRetrieves details of your Brevo account.**Use this to:**- Get account information (email, name, company, address)- Check plan details (type, credits, expiration)- Get relay information (for transactional emails)- Check Marketing Automation status- Access organization and user identifiers**Key information returned:**- Complete account details (organization ID, user ID, company information)- Address and contact information- Plan configurations and credit allocations across different verticals- Marketing Automation settings and tracker key (when enabled)- SMTP relay configuration for transactional emails- Enterprise features availability status**Important considerations:**- Provides comprehensive account overview for billing and configuration management- Essential for understanding current plan limitations and feature availability- Marketing Automation key is only returned when Marketing Automation is enabled on the account- Plan verticals show detailed breakdown across Marketing, Chat, and CRM categories (only returned when plan verticals are available)- Relay configuration crucial for transactional email setup and deliverability- Enterprise status determines access to advanced features and sub-account management

Catalog: Brevo

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	varchar2	Address City	☐	City information
address_country	varchar2	Address Country	☐	Country information
address_street	varchar2	Address Street	☐	Street information
address_zipCode	varchar2	Address Zip Code	☐	Zip Code information
companyName	varchar2	Company Name	☐	Name of the company
email	varchar2	Email Address	☐	Login Email
enterprise	char		☐	Indicates if the account has enterprise features
firstName	varchar2	First Name	☐	First Name
lastName	varchar2	Last Name	☐	Last Name
marketingAutomation_enabled	char	Marketing Automation Enabled	☐	Status of Marketing Automation Plateform activationfor your account (true=enabled, false=disabled)
marketingAutomation_key	varchar2	Marketing Automation Key	☐	Marketing Automation Tracker ID
organization_id	varchar2		☐	Unique identifier for the organization
relay_data_port	int64	Relay Data Port	☐	Port used for SMTP Relay

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
relay_data_relay	varchar2	Relay Data Relay	☐	URL of the SMTP Relay
relay_data_userName	varchar2	Relay Data Username	☐	Email to use as login on transactional platform
relay_enabled	char	Relay Enabled	☐	Status of your transactional email Account(true=Enabled, false=Disabled)
user_id	int64		☐	Unique identifier for the user

3.1.2 Schema: Balance

3.1.2.1 Tables

BalanceDefinitionByPidAndBdid

Get balance definitionReturns balance definition

Catalog: Brevo

Schema: Balance

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 BalanceDefinitionByPidAndBdid. 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
bdid		☑		Balance Definition Id
pid		☑		Loyalty Program Id
version		☐		Version (Values: active, draft)

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
balanceAvailabilityDurationModifier	varchar2		☐	Modifier for balance availability duration. startOfPeriod causes expiry at the start of the period, endOfPeriod at the end. noModification uses the exact duration from the credit date.
balanceAvailabilityDurationUnit	varchar2		☐	Unit of time for the balance availability duration.
balanceAvailabilityDurationValue	int64		☐	Number of time units before the balance expires.
balanceExpirationDate	varchar2		☐	Fixed expiration date in dd/mm format. The balance expires when this date next appears in the calendar. Only one of balanceExpirationDate or balance availability duration fields can be used.
balanceOptionAmountOvertakingStrategy	varchar2		☐	Partial enables partial credit of balance if maximum balance limit is reached. Strict enables rejection of a transaction if it will breach the max balance limit.
balanceOptionCreditRounding	varchar2		☐	Rounding strategy for credit transactions.
balanceOptionDebitRounding	varchar2		☐	Rounding strategy for debit transactions.
createdAt	datetime	Creation Date	☐	Timestamp of balance definition creation.
deletedAt	datetime		☐	Timestamp of balance definition deletion (nullable).
description	varchar2	Description	☐	Short description of the balance definition.
id	guid	ID	☐	Unique identifier for the balance definition.
imageRef	varchar2		☐	Optional image reference URL.
maxAmount	decimal		☐	Maximum allowable balance.
maxCreditAmountLimit	decimal		☐	Maximum credit allowed per operation.
maxDebitAmountLimit	decimal		☐	Maximum debit allowed per operation.
meta	varchar2		☐	Additional metadata for the balance definition.
minAmount	decimal		☐	Minimum allowable balance.
name	varchar2	Name	☐	Name of the balance definition.
unit	varchar2		☐	Unit of balance measurement.
updatedAt	datetime		☐	Timestamp of the last update.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

BalanceDefinitionsByPid

Get balance definition listReturns balance definition page

Catalog: Brevo

Schema: Balance

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 BalanceDefinitionsByPid. 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
pid		☑		Loyalty Program Id
sortField		☐		Field to sort by (Values: name, created_at, updated_at)
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceAvailabilityDurationModifier	varchar2		☐	Modifier for balance availability duration. startOfPeriod causes expiry at the start of the period, endOfPeriod at the end. noModification uses the exact duration from the credit date.
balanceAvailabilityDurationUnit	varchar2		☐	Unit of time for the balance availability duration.
balanceAvailabilityDurationValue	int64		☐	Number of time units before the balance expires.
balanceExpirationDate	varchar2		☐	Fixed expiration date in dd/mm format. The balance expires w hen this date next appears in the calendar. Only one of bal-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				anceExpirationDate or balance availability duration fields can be used.
balanceOptionAmountOvertak-ingStrategy	varchar2		☐	Partial enables partial credit of balance if maximum balance limit is reached. Strict enables rejec-tion of a transaction if it w ill breach the max balance limit.
balanceOptionCreditRounding	varchar2		☐	Rounding strategy for credit transactions.
balanceOptionDebitRounding	varchar2		☐	Rounding strategy for debit transactions.
createdAt	datetime	Creation Date	☐	Timestamp of balance definition creation.
deletedAt	datetime		☐	Timestamp of balance definition deletion (nullable).
description	varchar2	Description	☐	Short description of the balance definition.
id	guid	ID	☐	Unique identifier for the balance definition.
imageRef	varchar2		☐	Optional image reference URL.
maxAmount	decimal		☐	Maximum allow able balance.
maxCreditAmountLimit	decimal		☐	Maximum credit allow ed per op-eration.
maxDebitAmountLimit	decimal		☐	Maximum debit allow ed per oper-ation.
meta	varchar2		☐	Additional metadata for the bal-ance definition.
minAmount	decimal		☐	Minimum allow able balance.
name	varchar2	Name	☐	Name of the balance definition.
unit	varchar2		☐	Unit of balance measurement.
updatedAt	datetime		☐	Timestamp of the last update.

BalanceLimitsByPidAndBdid

Get balance limitsFetches balance limits and send the created UUID along with the data

Catalog: Brevo

Schema: Balance

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 Bal-anceLimitsByPidAndBdid. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
bdid		☑		Balance Definition Id
blid		☑		Balance Limit Id
pid		☑		Loyalty Program Id
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	guid		☐	Balance definition ID.
constraintType	varchar2		☐	Defines the type of constraint (transaction count or amount).
createdAt	datetime	Creation Date	☐	Timestamp of when the balance limit was created.
durationUnit	varchar2		☐	Time unit for the balance limit.
durationValue	int64		☐	Number of time units the balance limit applies to.
id	guid	ID	☐	Unique identifier for the balance limit.
slidingSchedule	char		☐	Indicates if the limit resets periodically based on a sliding schedule.
transactionType	varchar2		☐	Specifies whether the limit applies to credit or debit transactions.
updatedAt	datetime		☐	Timestamp of the last update to the balance limit.
value	decimal	Value	☐	The maximum allowed value for the defined constraint.

ContactBalances

Get balance listReturns contact balances for a given balance definition across all subscriptions.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: Balance

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 ContactBalances. 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
balanceDefinitionId		☑		Balance Definition ID (required)
includeInternal		☐		Include balances tied to internal definitions.
pid		☑		Loyalty Program Id
sortField		☐		Field to sort by (Values: updatedAt, value)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	varchar2		☐	
count	int64	Count	☐	
loyaltyProgramId	varchar2		☐	

ContactBalances_Balances

Get balance listReturns contact balances for a given balance definition across all subscriptions.

Catalog: Brevo

Schema: Balance

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 ContactBalances_Balances. 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
balanceDefinitionId		☑		Balance Definition ID (required)
includeInternal		☐		Include balances tied to internal definitions.
pid		☑		Loyalty Program Id
sortField		☐		Field to sort by (Values: updatedAt, value)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	varchar2		☐	
contactId	int64		☐	
count	int64	Count	☐	
loyaltyProgramId	varchar2		☐	
loyaltySubscriptionId	varchar2		☐	
updatedAt	varchar2		☐	
value	decimal	Value	☐	

LoyaltyBalanceProgramsByPid_ActiveBalancesActiveBalance

Get Active Balances APIReturns Active Balances

Catalog: Brevo

Schema: Balance

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 LoyaltyBalanceProgramsByPid_ActiveBalancesActiveBalance. 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
balanceDefinitionId		☑		Balance Definition ID
contactId		☑		Contact ID
includeInternal		☐		Include balances tied to internal definitions.
pid		☑		Loyalty Program Id
sortField		☐		Sort Field (Values: createdAt)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
amount	double		☐	Current balance amount.
balanceDefinitionId	guid		☐	Unique identifier of the balance definition.
contactId	int64		☐	Unique identifier of the contact.
count	int64	Count	☐	Total number of active balances.
createdAt	datetime	Creation Date	☐	Timestamp when the balance was created.
expiresAt	datetime		☐	Timestamp when the balance expires (if applicable).
id	guid	ID	☐	Unique identifier of the balance entry.
loyaltyProgramId	guid		☐	Unique identifier of the loyalty program.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

LoyaltyBalanceProgramsByPid_TransactionHistoryTransactionHistory

Get Transaction History APIReturns transaction history

Catalog: Brevo

Schema: Balance

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 LoyaltyBalanceProgramsByPid_TransactionHistoryTransactionHistory. 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
balanceDefinitionId		☑		Balance Definition ID
contactId		☑		Contact ID
loyaltySubscriptionId		☐		Loyalty Subscription ID filter
pid		☑		Loyalty Program Id
sortField		☐		Field to sort by (Values: createdAt)
status		☐		Transaction status filter (Values: draft, completed, rejected, cancelled, expired)
transactionType		☐		Transaction type filter (Values: credit, debit)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
amount	double		☐	The transaction amount.
balanceDefinitionId	varchar2		☐	Unique identifier of the associated balance definition.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
balanceExpirationDate	varchar2		☐	Expiration date of the balance associated with this transaction.
cancelledAt	varchar2		☐	Timestamp when the transaction was canceled, if applicable.
completedAt	varchar2		☐	Timestamp when the transaction was successfully completed.
contactId	int64		☐	Unique identifier of the contact related to the transactions.
count	int64	Count	☐	Total number of transactions in the history.
createdAt	varchar2	Creation Date	☐	Timestamp when the transaction was initiated.
id	varchar2	ID	☐	Unique identifier of the transaction.
loyaltyProgramId	varchar2		☐	Unique identifier of the associated loyalty program.
meta	varchar2		☐	Optional metadata associated with the transaction.
rejectedAt	varchar2		☐	Timestamp when the transaction was rejected.
rejectReason	varchar2		☐	Reason for rejection, if the transaction was declined.
status	varchar2	Status	☐	Current status of the transaction (e.g., pending, completed, rejected).
transactionType	varchar2		☐	The type of the transaction.

LoyaltyBalanceProgramsByPidActiveBalance

Get Active Balances APIReturns Active Balances

Catalog: Brevo

Schema: Balance

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 LoyaltyBalanceProgramsByPidActiveBalance. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
balanceDefinitionId		☑		Balance Definition ID
contactId		☑		Contact ID
includeInternal		☐		Include balances tied to internal definitions.
pid		☑		Loyalty Program Id
sortField		☐		Sort Field (Values: createdAt)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	guid		☐	Unique identifier of the balance definition.
contactId	int64		☐	Unique identifier of the contact.
count	int64	Count	☐	Total number of active balances.
loyaltyProgramId	guid		☐	Unique identifier of the loyalty program.

LoyaltyBalanceProgramsByPidTransactionHistory

Get Transaction History APIReturns transaction history

Catalog: Brevo

Schema: Balance

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 LoyaltyBalanceProgramsByPidTransactionHistory. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
balanceDefinitionId		☑		Balance Definition ID
contactId		☑		Contact ID
loyaltySubscriptionId		☐		Loyalty Subscription ID filter
pid		☑		Loyalty Program Id
sortField		☐		Field to sort by (Values: createdAt)
status		☐		Transaction status filter (Values: draft, completed, rejected, cancelled, expired)
transactionType		☐		Transaction type filter (Values: credit, debit)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	varchar2		☐	Unique identifier of the associated balance definition.
contactId	int64		☐	Unique identifier of the contact related to the transactions.
count	int64	Count	☐	Total number of transactions in the history.
loyaltyProgramId	varchar2		☐	Unique identifier of the associated loyalty program.

SubscriptionBalances

Get subscription balancesReturns subscription balances

Catalog: Brevo

Schema: Balance

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 SubscriptionBalances. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
cid		☑		Contact Id
includeInternal		☐		Include balances tied to internal definitions.
pid		☑		Loyalty Program Id

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
contactId	int64		☐	Unique identifier of the contact.
loyaltyProgramId	guid		☐	Unique identifier of the loyalty program.

SubscriptionBalances_Balance

Get subscription balancesReturns subscription balances

Catalog: Brevo

Schema: Balance

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 SubscriptionBalances_Balance. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
cid		☑		Contact Id
includeInternal		☐		Include balances tied to internal definitions.
pid		☑		Loyalty Program Id

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	guid		☐	Balance definition ID.
contactId	int64		☐	Unique identifier of the contact.
loyaltyProgramId	guid		☐	Unique identifier of the loyalty program.
value	double	Value	☐	Aggregate balance value for this definition.

3.1.3 Schema: Companies

3.1.3.1 Tables

Companies

Get all CompaniesRetrieve a paginated list of companies with optional filtering, sorting, and search capabilities. Results are sorted by creation date in descending order by default, and can be filtered by attributes, linked contacts, linked deals, or modification/creation timestamps.

Catalog: Brevo

Schema: Companies

Primary Keys: id

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Companies. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		Filter (urlencoded) the companies 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.
filters[attributes.name]		☐		Filter by attributes. If you have a filter for the owner on your side please send it as filters[attributes.owner] and utilize the account email for the filtering.
linkedContactsIds		☐		Filter by linked contacts ids
linkedDealsIds		☐		Filter by linked Deals ids
modifiedSince		☐		Filter (urlencoded) the companies 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.
page		☐		Index of the first document of the page
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Company attributes with values
id	varchar2	ID	☐	Unique company id
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this company

Companies_ItemsLinkedContactsIds

Get all CompaniesRetrieve a paginated list of companies with optional filtering, sorting, and search capabilities. Results are sorted by creation date in descending order by default, and can be filtered by attributes, linked contacts, linked deals, or modification/creation timestamps.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: Companies

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 Companies_ItemsLinkedContactsIds. 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		☐		Filter (urlencoded) the companies 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.
filters[attributes.name]		☐		Filter by attributes. If you have a filter for the ow ner on your side please send it as filters[attributes.ow ner] and utilize the account email for the filtering.
linkedContactsIds		☐		Filter by linked contacts ids
linkedDealsIds		☐		Filter by linked Deals ids
modifiedSince		☐		Filter (urlencoded) the companies 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.
page		☐		Index of the first document of the page
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Company attributes with values
id	varchar2	ID	☐	Unique company id
TEXT	int64	Text	☐	

Companies_ItemsLinkedDealsIds

Get all CompaniesRetrieve a paginated list of companies with optional filtering, sorting, and search capabilities. Results are sorted by creation date in descending order by default, and can be filtered by attributes, linked contacts, linked deals, or modification/creation timestamps.

Catalog: Brevo

Schema: Companies

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 Companies_ItemsLinkedDealsIds. 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		☐		Filter (urlencoded) the companies 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.
filters[attributes.name]		☐		Filter by attributes. If you have a filter for the owner on your side please send it as filters[attributes.owner] and utilize the account email for the filtering.
linkedContactsIds		☐		Filter by linked contacts ids
linkedDealsIds		☐		Filter by linked Deals ids
modifiedSince		☐		Filter (urlencoded) the companies modified after a given UTC date-time (YYYY-MM-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				DDTHH:mm:ss.SSSZ). Prefer to pass your timezone in date-time format for accurate result.
page		☐		Index of the first document of the page
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Company attributes with values
id	varchar2	ID	☐	Unique company id
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this company
TEXT	varchar2	Text	☐	

Companies_LinkedContactsIdsById

Get a companyRetrieve the full details of a single company by its identifier, including its attributes, linked contacts, and linked deals. Returns a 404 error if the company does not exist, or a 403 error if the user lacks permission to view the company.

Catalog: Brevo

Schema: Companies

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 Companies_LinkedContactsIdsById. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
id		☑		Get Company Details

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Company attributes with values
id	varchar2	ID	☐	Unique company id
TEXT	int64	Text	☐	

Companies_LinkedDealsIdsById

Get a companyRetrieve the full details of a single company by its identifier, including its attributes, linked contacts, and linked deals. Returns a 404 error if the company does not exist, or a 403 error if the user lacks permission to view the company.

Catalog: Brevo

Schema: Companies

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Get Company Details

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Company attributes with values
id	varchar2	ID	☐	Unique company id
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this company
TEXT	varchar2	Text	☐	

CompaniesById

Get a companyRetrieve the full details of a single company by its identifier, including its attributes, linked contacts, and linked deals. Returns a 404 error if the company does not exist, or a 403 error if the user lacks permission to view the company.

Catalog: Brevo

Schema: Companies

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Get Company Details

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Company attributes with values
id	varchar2	ID	☐	Unique company id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this company

CrmAttributes_AttributeOptionsCompanies

Get company attributesRetrieve the list of all attributes defined for companies, including both system-default and custom attributes. Each attribute includes its label, internal name, type, required status, and available options for select-type attributes.

Catalog: Brevo

Schema: Companies

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 CrmAttributes_AttributeOptionsCompanies are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
attributeTypeName	varchar2		☐	Type of the attribute
internalName	varchar2		☐	Internal name of the attribute
isRequired	char		☐	Whether this attribute is required
isValueReadonly	char		☐	Whether this attribute value is read-only
key	varchar2	Key	☐	
label	varchar2	Label	☐	Display label of the attribute
value	varchar2	Value	☐	

CrmAttributesCompanies

Get company attributesRetrieve the list of all attributes defined for companies, including both system-default and custom attributes. Each attribute includes its label, internal name, type, required status, and available options for select-type attributes.

Catalog: Brevo

Schema: Companies

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 CrmAttributesCompanies are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
attributeTypeName	varchar2		☐	Type of the attribute
internalName	varchar2		☐	Internal name of the attribute
isRequired	char		☐	Whether this attribute is required
isValueReadonly	char		☐	Whether this attribute value is read-only
label	varchar2	Label	☐	Display label of the attribute

3.1.4 Schema: Contacts

3.1.4.1 Tables

AttributeMultiCategoryOptions

List all attributesRetrieve all contact attributes defined in your Brevo account, grouped by category (normal, transactional, category, calculated, global). Each attribute includes its name, type, and category, along with enumeration values for category-type attributes and options for multiple-choice-type attributes.

Catalog: Brevo

Schema: Contacts

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 AttributeMultiCategoryOptions are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
calculatedValue	varchar2	Calculated Value	☐	Calculated value formula
category	varchar2	Category	☐	Category of the attribute
name	varchar2	Name	☐	Name of the attribute
TEXT	varchar2	Text	☐	Options of the "multiple-choice" type attribute
type	varchar2	Type	☐	Type of the attribute

Attributes: Sendinblue Attributes

List all attributesRetrieve all contact attributes defined in your Brevo account, grouped by category (normal, transactional, category, calculated, global). Each attribute includes its name, type, and category, along with enumeration values for category-type attributes and options for multiple-choice-type attributes.

Catalog: Brevo

Schema: Contacts

Label: Attributes

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	Calculated Value	☐	Calculated value formula
category	varchar2	Category	☐	Category of the attribute
name	varchar2	Name	☐	Name of the attribute
type	varchar2	Type	☐	Type of the attribute

AttributeValues: Sendinblue Attribute Values

List all attributesRetrieve all contact attributes defined in your Brevo account, grouped by category (normal, transactional, category, calculated, global). Each attribute includes its name, type, and category, along with enumeration values for category-type attributes and options for multiple-choice-type attributes.

Catalog: Brevo

Schema: Contacts

Label: Attribute Values

This is a read-only table. The Sendinblue API may not support changing the data or the 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	varchar2	Calculated Value	☐	Calculated value formula
category	varchar2	Category	☐	Category of the attribute
label	varchar2	Label	☐	Label of the "category" type attribute
name	varchar2	Name	☐	Name of the attribute
type	varchar2	Type	☐	Type of the attribute
value	int64	Value	☐	Numeric ID of the "category" type attribute value. Set to 0 when the raw value cannot be converted to an integer (for example non-numeric values such as "en" or "fr"). Refer to `valueStr` for the original string representation.
valueStr	varchar2		☐	String representation of the "category" type attribute value. Always contains the original value as stored. Use this field when

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				the attribute value is non-numeric (e.g. "en", "fr") or w hen you need the exact string form along-side the numeric `value`.

ContactAdminUnsubscriptionsByEmail: Sendinblue Contact Unsubscriptions by Email (Administrative Mode)

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Unsubscriptions by Email (Administrative Mode)

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Event Time	☐	UTC date-time of the event
ip	varchar2	IP	☐	IP from w hich the user has been unsubscribed

ContactClickStatisticsByEmail

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

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

ContactComplaintStatisticsByEmail: Sendinblue Contact Complaint Statistics by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Complaint Statistics by Email

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

Parameters of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Event Time	☐	UTC date-time of the event

ContactDeliveryStatisticsByEmail

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactDeliveryStatisticsByEmail. 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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

Columns of Table Function

The columns of the table function ContactDeliveryStatisticsByEmail 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	varchar2	Event Time	☐	UTC date-time of the event

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

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxx +91xxxxxxxxx 0091xxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- ht-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

tps://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifier-Type is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, land-line_number_id for LANDLINE_NUMBER attribute Along with the contact details, this end-point will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/refer-ence/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Unsubscriptions by Email (Administrative Mode)

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

Parameters of Table Function

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

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four 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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events spe-cific to campaigns. Must be greater than equal to startDate. Must not be greater than the cur-rent date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlen-coded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, land-line_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, con-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				tact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	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	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailClickStatisticsByEmail

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

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
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich gener-ated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS cam-paigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailComplaintStatisticsByEmail: Sendinblue Contact Detail Complaint Statistics by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attrib-ute.Accepted Number Formats 91xxxxxxxxx +91xxxxxxxxx 0091xxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- ht-tps://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifier-Type is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, land-line_number_id for LANDLINE_NUMBER attribute Along with the contact details, this end-point will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/refer-ence/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Complaint Statistics by Email

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich gener-ated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
eventTime	varchar2	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS cam-paigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailDeliveryStatisticsByEmail

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attrib-ute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- ht-tps://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifier-Type is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, land-line_number_id for LANDLINE_NUMBER attribute Along with the contact details, this end-point will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/refer-ence/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function ContactDetailDeliveryStatisticsByEmail. 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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
eventTime	varchar2	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS cam-paigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailHardBounceStatisticsByEmail: Sendinblue Contact Detail Hard Bounce Statistics by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attrib-ute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- ht-tps://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifier-Type is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, land-line_number_id for LANDLINE_NUMBER attribute Along with the contact details, this end-point will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/refer-ence/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Hard Bounce Statistics by Email

This is a read-only table function. The Sendinblue API may not support changing the data or the 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 re-quired parameters, but optional parameters in general do not need to have a value and the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	varchar2	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailLinkClickStatisticsByEmail: Sendinblue Contact Detail Link Click Statistics by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Link Click Statistics by Email

This is a read-only table function. The Sendinblue API may not support changing the data or the 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 para-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailMessageSentStatisticsByEmail: Sendinblue Contact Detail Message Sent Statistics by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxx +91xxxxxxxxx 0091xxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Message Sent Statistics by Email

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

Parameters of Table Function

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

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

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	varchar2	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailOpenStatisticsByEmail: Sendinblue Contact Detail Open Statistics by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Open Statistics by Email

This is a read-only table function. The Sendinblue API may not support changing the data or the 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
count	int64	Count	☐	Number of openings for the campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	varchar2	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for w hich you requested the details
ip	varchar2	IP	☐	IP from w hich the user has opened the email
listIds	varchar2	List IDs	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
whatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailSoftBounceStatisticsByEmail: Sendinblue Contact Detail Soft Bounce Statistics by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Soft Bounce Statistics by Email

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

Parameters of Table Function

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

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

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
campaignId	int64	Campaign ID	☐	ID of the campaign w hich generated the event
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
eventTime	varchar2	Event Time	☐	UTC date-time of the event
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
whatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactDetailTransactionAttributesByEmail: Sendinblue Contact Detail Transaction Attributes by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxx +91xxxxxxxxx 0091xxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail Transaction Attributes by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events spe-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				cific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

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	varchar2	Attributes	☐	Set of attributes of the contact
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	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	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ContactDetailUserUnsubscriptionsByEmail: Sendinblue Contact Detail User Unsubscriptions by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Detail User Unsubscriptions by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, w hatsapp_id for WHATSAPP attribute, land-line_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, con-tact_id, ext_id, w hatsapp_id, landline_number_id)
startDate		☐		**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. Must not be greater than the current date.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Set of attributes of the contact
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
id	int64	ID	☐	ID of the contact for w hich you requested the details
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS cam-paigns (true=blacklisted, false=not blacklisted)
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactHardBounceStatisticsByEmail: Sendinblue Contact Hard Bounce Statistics by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attrib-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

utes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Hard Bounce Statistics by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
campaignId	int64	Campaign ID	☐	ID of the campaign which generated the event
eventTime	varchar2	Event Time	☐	UTC date-time of the event

ContactInfoByEmail: Sendinblue Contact Information by Email

Get a contact's detailsFollow this format when passing a "SMS" phone number as an attribute.Accepted Number Formats 91xxxxxxxxxx +91xxxxxxxxxx 0091xxxxxxxxxxThere are 2 ways to get a contact Option 1- https://api.brevo.com/v3/contacts/{identifier} Option 2- https://api.brevo.com/v3/contacts/{identifier}?identifierType={} Option 1 only works if identifierType is email_id (for EMAIL), phone_id (for SMS) or contact_id (for ID of the contact),where you can directly pass the value of EMAIL, SMS and ID of the contact. Option 2 works for all identifierType, use email_id for EMAIL attribute, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute Along with the contact details, this endpoint will show the statistics of contact for the recent 90 days by default. To fetch the earlier statistics, please use Get contact campaign stats ``https://developers.brevo.com/reference/contacts-7#getcontactstats`` endpoint with the appropriate date ranges.

Catalog: Brevo

Schema: Contacts

Label: Contact Information by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
identifier		☑		Email (urlencoded) OR ID of the contact OR its SMS attribute value OR EXT_ID attribute (urlencoded)
identifierType		☐		email_id for Email, phone_id for SMS attribute, contact_id for ID of the contact, ext_id for EXT_ID attribute, whatsapp_id for WHATSAPP attribute, landline_number_id for LANDLINE_NUMBER attribute (Values: email_id, phone_id, contact_id, ext_id, whatsapp_id, landline_number_id)
startDate		☐		**Mandatory if endDate is used.** Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be lower than equal to endDate. Must not be greater than the current date.

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	varchar2		☐	
ABONNEMENTEN_ACTIEF	varchar2		☐	
BLACKLIST	decimal	Blacklist	☐	
BUBS_GBR_ID_MASTER	decimal		☐	
BUBS_GBR_ID	decimal	Estate Person ID	☐	
CLICKERS	decimal	Clickers	☐	
COMPANY_NAME	varchar2	Company Name	☐	
CONTACT_TIMEZONE	varchar2		☐	
CONTRACT_CODE	varchar2	Contract Code	☐	
COUNTRY	varchar2	Country	☐	
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	varchar2		☐	
D31BILLINGIDS	varchar2		☐	
D31BUBSGBRIDS	varchar2		☐	
D31DATACONTAINERCNT	decimal		☐	
D31MBIO	decimal		☐	
D31PARTITIONCNT	decimal		☐	
D31PROVIDERCNT	decimal		☐	
D31ROWCNT	decimal		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
D31TABLECNT	decimal		☐	
DOUBLE_OPT-IN	varchar2		☐	
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
EXT_ID	varchar2		☐	
FIRSTNAME	varchar2	First Name	☐	
FORUMS_LOGIN	varchar2	Forums Login	☐	
id	int64	ID	☐	ID of the contact for w hich you requested the details
JOB_TITLE	varchar2		☐	
JOBTITLE	varchar2	Job Title	☐	
LANDLINE_NUMBER	varchar2		☐	
LANGUAGE_CODE	varchar2	Language Code	☐	
LASTNAME	varchar2	Last Name	☐	
LINKEDIN	varchar2		☐	
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
MOBILE_NUMBER	varchar2	Mobile Number	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
OPT_IN	char		☐	
PHONE_NUMBER	varchar2	Phone Number	☐	
PRIMARY_AUDIENCE	varchar2	Primary Audience	☐	
READERS	decimal	Readers	☐	
SMS	varchar2	SMS	☐	
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	varchar2	WhatsApp	☐	
w hatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactLinkClicksByEmail: Sendinblue Contact Link Clicks by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

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
count	int64	Count	☐	Number of clicks on this link for the campaign

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
eventTime	varchar2	Event Time	☐	UTC date-time of the event
ip	varchar2	IP	☐	IP from w hich the user has clicked on the link
url	varchar2	URL	☐	URL of the clicked link

ContactMessageSentStatisticsByEmail: Sendinblue Contact Message Sent Statistics by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Message Sent Statistics by Email

This is a read-only table function. The Sendinblue API may not support changing the data or the 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
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startDate		☐		**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. Must not be greater than the current date.

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 w hich gener-ated the event
eventTime	varchar2	Event Time	☐	UTC date-time of the event

ContactOpenStatisticsByEmail: Sendinblue Contact Open Statistics by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Open Statistics by Email

This is a read-only table function. The Sendinblue API may not support changing the data or the 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

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 w hich generated the event
count	int64	Count	☐	Number of openings of the campaign
eventTime	varchar2	Event Time	☐	UTC date-time of the event
ip	varchar2	IP	☐	IP from w hich the user has opened the campaign

Contacts: Sendinblue Contacts

Get all the contactsRetrieve all contacts from your Brevo account with support for pagination, filtering, and sorting. Results default to 50 contacts per page (maximum 1000) sorted in descending order of creation, and can be filtered by modification date, creation date, contact IDs (up to 20), list IDs, segment ID, or contact attributes using the equals operator. Note that either listIds or segmentId can be passed but not both simultaneously.

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Contacts

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

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		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.**
filter		☐		Filter the contacts on the basis of attributes. **Allowed operator: equals. For multiple-choice options, the filter will apply an AND condition between the options. For category attributes, the filter will work with both id and value. (e.g. filter=equals(FIRSTNAME,"Antoine"), filter=equals(B1, true), filter=equals(DOB, "1989-11-23"), filter=equals(GENDER, "1"), filter=equals(GENDER, "MALE"), filter=equals(COUNTRY,"USA, INDIA")**
modifiedSince		☐		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.**
segmentId		☐		Id of the segment. **Either listIds or segmentId can be passed.** Must be a positive integer (minimum value of 1).

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	varchar2		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
ABONNEMENTEN_ACTIEF	varchar2		☐	
BLACKLIST	decimal	Blacklist	☐	
BUBS_GBR_ID_MASTER	decimal		☐	
BUBS_GBR_ID	decimal	Estate Person ID	☐	
CLICKERS	decimal	Clickers	☐	
COMPANY_NAME	varchar2	Company Name	☐	
CONTACT_TIMEZONE	varchar2		☐	
CONTRACT_CODE	varchar2	Contract Code	☐	
COUNTRY	varchar2	Country	☐	
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	varchar2		☐	
D31BILLINGIDS	varchar2		☐	
D31BUBSGBRIDS	varchar2		☐	
D31DATACONTAINERCNT	decimal		☐	
D31MBIO	decimal		☐	
D31PARTITIONCNT	decimal		☐	
D31PROVIDERCNT	decimal		☐	
D31ROWCNT	decimal		☐	
D31TABLECNT	decimal		☐	
DOUBLE_OPT-IN	varchar2		☐	
email	varchar2	Email Address	☐	Email address of the contact for which you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email campaigns (true=blacklisted, false=not blacklisted)
EXT_ID	varchar2		☐	
FIRSTNAME	varchar2	First Name	☐	
FORUMS_LOGIN	varchar2	Forums Login	☐	
id	int64	ID	☐	ID of the contact for which you requested the details
JOB_TITLE	varchar2		☐	
JOBTITLE	varchar2	Job Title	☐	
LANDLINE_NUMBER	varchar2		☐	
LANGUAGE_CODE	varchar2	Language Code	☐	
LASTNAME	varchar2	Last Name	☐	
LINKEDIN	varchar2		☐	
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
MOBILE_NUMBER	varchar2	Mobile Number	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
OPT_IN	char		☐	
PHONE_NUMBER	varchar2	Phone Number	☐	
PRIMARY_AUDIENCE	varchar2	Primary Audience	☐	
READERS	decimal	Readers	☐	
SMS	varchar2	SMS	☐	
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	varchar2	WhatsApp	☐	
whatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactsByListId: Sendinblue Contacts by List ID

Get contacts in a listRetrieve all contacts belonging to a specific list, identified by its list ID. Results are paginated with a default of 50 contacts per page (maximum 500) and sorted in descending order of creation. You can optionally filter contacts by their modification date using the modifiedSince parameter.

Catalog: Brevo

Schema: Contacts

Label: Contacts by List ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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		☑		Id of the list
modifiedSince		☐		Filter (urlencoded) the contacts modified after a given UTC date-time (YYYY-MM-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				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	varchar2		☐	
ABONNEMENTEN_ACTIEF	varchar2		☐	
BLACKLIST	decimal	Blacklist	☐	
BUBS_GBR_ID_MASTER	decimal		☐	
BUBS_GBR_ID	decimal	Estate Person ID	☐	
CLICKERS	decimal	Clickers	☐	
COMPANY_NAME	varchar2	Company Name	☐	
CONTACT_TIMEZONE	varchar2		☐	
CONTRACT_CODE	varchar2	Contract Code	☐	
COUNTRY	varchar2	Country	☐	
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
D31APPLICATIONS	varchar2		☐	
D31BILLINGIDS	varchar2		☐	
D31BUBSGBRIDS	varchar2		☐	
D31DATACONTAINERCNT	decimal		☐	
D31MBIO	decimal		☐	
D31PARTITIONCNT	decimal		☐	
D31PROVIDERCNT	decimal		☐	
D31ROWCNT	decimal		☐	
D31TABLECNT	decimal		☐	
DOUBLE_OPT-IN	varchar2		☐	
email	varchar2	Email Address	☐	Email address of the contact for w hich you requested the details
emailBlacklisted	char	Email is Blacklisted	☐	Blacklist status for email cam-paigns (true=blacklisted, false=not blacklisted)
EXT_ID	varchar2		☐	
FIRSTNAME	varchar2	First Name	☐	
FORUMS_LOGIN	varchar2	Forums Login	☐	
id	int64	ID	☐	ID of the contact for w hich you requested the details
JOB_TITLE	varchar2		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
JOBTITLE	varchar2	Job Title	☐	
LANDLINE_NUMBER	varchar2		☐	
LANGUAGE_CODE	varchar2	Language Code	☐	
LASTNAME	varchar2	Last Name	☐	
LINKEDIN	varchar2		☐	
listIds	varchar2	List IDs	☐	
listUnsubscribed	varchar2	Unsubscribed List IDs	☐	
MOBILE_NUMBER	varchar2	Mobile Number	☐	
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the contact (YYYY-MM-DDTHH:mm:ss.SSSZ)
OPT_IN	char		☐	
PHONE_NUMBER	varchar2	Phone Number	☐	
PRIMARY_AUDIENCE	varchar2	Primary Audience	☐	
READERS	decimal	Readers	☐	
SMS	varchar2	SMS	☐	
smsBlacklisted	char	SMS is Blacklisted	☐	Blacklist status for SMS campaigns (true=blacklisted, false=not blacklisted)
WHATSAPP	varchar2	WhatsApp	☐	
whatsappBlacklisted	char		☐	Blacklist status for WhatsApp campaigns (true=blacklisted, false=not blacklisted)

ContactSoftBounceStatisticsByEmail: Sendinblue Contact Soft Bounce Statistics by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Soft Bounce Statistics by Email

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

Parameters of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

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 generated the event
eventTime	varchar2	Event Time	☐	UTC date-time of the event

ContactStats

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactStats. 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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2(1)	Dummy	☑	Default column added since the specification specifies that no data is returned.

ContactTransactionAttributesByEmail: Sendinblue Contact Transaction Attributes by Email

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: Contacts

Label: Contact Transaction Attributes by Email

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**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. Must not be greater than the current date.

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
orderPrice	double	Order Price	☐	Price of the order

ContactUserUnsubscriptionsByEmail: Sendinblue Contact Unsubscriptions by Email (User Mode)

Get email campaigns' statistics for a contactRetrieve email campaign statistics for a specific contact identified by email address or numeric ID. Statistics include messages sent, opens, clicks, hard/soft bounces, deliveries, unsubscriptions, complaints, and transactional attributes. By default, data covers the last 90 days; use startDate and endDate parameters (YYYY-MM-DD) to specify a custom range with a maximum span of 90 days.

Catalog: Brevo

Schema: Contacts

Label: Contact Unsubscriptions by Email (User Mode)

This is a read-only table function. The Sendinblue API may not support changing the data or the 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
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be greater than equal to startDate. Must not be greater than the current date. Maximum difference betw een startDate and endDate should not be greater than 90 days.
identifier		☑		Email (urlencoded) OR ID of the contact
startDate		☐		**Mandatory if endDate is used.** Starting date (YYYY-MM-DD) of the statistic events specific to campaigns. Must be low er than

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				equal to endDate. Must not be greater than the current date.

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 which generated the event
eventTime	varchar2	Event Time	☐	UTC date-time of the event
ip	varchar2	IP	☐	IP from which the user has unsubscribed

FolderById: Sendinblue Folder by ID

Returns a folder's detailsOngoing changes for this endpoint.We're dropping support for the response attributes totalSubscribers and totalBlacklisted.These are non breaking changes. The default value for the attributes will be 0.

Catalog: Brevo

Schema: Contacts

Label: Folder by ID

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

Parameters of Table Function

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

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

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

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	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

Folders: Sendinblue Folders

Get all foldersOngoing changes for this endpointWe're dropping support for the response attributes totalSubscribers and totalBlacklisted.These are non breaking changes.The default value for the attributes will be 0.The uniqueSubscribers field is deprecated

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Folders

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
id	int64	ID	☐	ID of the folder
name	varchar2	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

ListById: Sendinblue List by ID

Get a list's detailsRetrieve the details of a specific contact list by its ID, including its name, folder ID, creation date, subscriber counts, and campaign statistics. You can optionally filter campaign statistics by providing startDate and endDate parameters (both must be used together in YYYY-MM-DD format).

Catalog: Brevo

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Contacts

Label: List by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
endDate		☐		**Mandatory if startDate is used**. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to aggregate the sent email campaigns for a specific list id. **Prefer to pass your timezone in date-time format for accurate result**
listId		☑		Id of the list
startDate		☐		**Mandatory if endDate is used**. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to aggregate the sent email campaigns for a specific list id. **Prefer to pass your timezone in date-time format for accurate result**

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	varchar2	Creation Date	☐	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	char	Dynamic List	☐	Status telling if the list is dynamic or not (true=dynamic, false=not dynamic)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
folderId	int64	Folder ID	☐	ID of the folder
id	int64	ID	☐	ID of the list
name	varchar2	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
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the list

ListCampaignStatisticsByListId: Sendinblue List Campaign Statistics by List ID

Get a list's detailsRetrieve the details of a specific contact list by its ID, including its name, folder ID, creation date, subscriber counts, and campaign statistics. You can optionally filter campaign statistics by providing startDate and endDate parameters (both must be used together in YYYY-MM-DD format).

Catalog: Brevo

Schema: Contacts

Label: List Campaign Statistics by List ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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
endDate		☐		**Mandatory if startDate is used**. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to aggregate the sent email campaigns for a specific list id. **Prefer to pass your timezone in date-time format for accurate result**
listId		☑		Id of the list

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startDate		☐		**Mandatory if endDate is used**. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to aggregate the sent email campaigns for a specific list id. **Prefer to pass your timezone in date-time format for accurate result**

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	varchar2	Creation Date	☐	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	char	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	varchar2	Name	☐	Name of the list
stats_appleMppOpens	int64		☐	Numbers of times your email has been opened automatically through Apple MPP.
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_estimatedViews	int64		☐	Rate of recipients without any privacy protection option enabled in their email client, applied to all delivered emails
stats_hardBounces	int64	#Hard-bounces	☐	Number of hard bounces 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_opensRate	double		☐	Percentage of recipients who open the email out of your total number of recipients. Depending on your Campaign settings, they may include Apple MPP opens.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
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 email client
stats_trackableViewsRate	double		☐	Rate of 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
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the list

Lists: Sendinblue Lists

Get all the listsOngoing changes for this endpoint.We're dropping support for the response attributes totalSubscribers and totalBlacklisted.These are non breaking changes. The default value for the attributes will be 0.

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Lists

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

Table Columns

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

Name	Data Type	Label	Required	Documentation
folderId	int64	Folder ID	☐	ID of the folder
id	int64	ID	☐	ID of the list

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
name	varchar2	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
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the list

ListsByFolderId: Sendinblue Lists by Folder ID

Get lists in a folderOngoing changes for this endpoint.We're dropping support for the response attributes totalSubscribers and totalBlacklisted.These are non breaking changes. The default value for the attributes will be 0.

Catalog: Brevo

Schema: Contacts

Label: Lists by Folder ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
folderId		☑		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	varchar2	Name	☐	Name of the list

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Segments: Sendinblue Segments

Get all the segmentsRetrieve all contact segments defined in your Brevo account with support for pagination and sorting. Results default to 10 segments per page (maximum 50) sorted in descending order of creation. Each segment includes its ID, name, category name, and last update timestamp.

Catalog: Brevo

Schema: Contacts

Primary Keys: id

Label: Segments

This is a read-only table. The Sendinblue API may not support changing the data or the 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	varchar2		☐	Name of the Segment Category
id	int64	ID	☐	ID of the segment
segmentName	varchar2		☐	Name of the Segment
updatedAt	varchar2		☐	Last update UTC date-time of the segment (YYYY-MM-DDTHH:mm:ss.SSSZ)

3.1.5 Schema: Conversations

3.1.5.1 Tables

ConversationMessageBccsByConversationId

Get a messageRetrieve a single message by its ID.

Catalog: Brevo

Schema: Conversations

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
email	varchar2	Email Address	☐	
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
name	varchar2	Name	☐	
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be marked with receivedFrom property and then filtered out when received in a webhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationMessageCcsByConversationId

Get a messageRetrieve a single message by its ID.

Catalog: Brevo

Schema: Conversations

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on w hose behalf the message w as sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
name	varchar2	Name	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be marked with receivedFrom property and then filtered out when received in a webhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationMessageTosByConversationId

Get a messageRetrieve a single message by its ID.

Catalog: Brevo

Schema: Conversations

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 ConversationMessageTosByConversationId. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
email	varchar2	Email Address	☐	
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	`true` if the message is a forwarded message.'

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
name	varchar2	Name	☐	
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be marked with receivedFrom property and then filtered out when received in a webhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationPushedMessageBccsByConversationId

Get an automated messageRetrieve a single automated (pushed) message by its ID.

Catalog: Brevo

Schema: Conversations

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message sent previously

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
email	varchar2	Email Address	☐	
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
name	varchar2	Name	☐	
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be marked with receivedFrom property and then filtered out when received in a webhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationPushedMessageCcsByConversationId

Get an automated messageRetrieve a single automated (pushed) message by its ID.

Catalog: Brevo

Schema: Conversations

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message sent previously

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
email	varchar2	Email Address	☐	
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				message types).
name	varchar2	Name	☐	
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be marked with receivedFrom property and then filtered out when received in a webhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationPushedMessageTosByConversationId

Get an automated messageRetrieve a single automated (pushed) message by its ID.

Catalog: Brevo

Schema: Conversations

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 ConversationPushedMessageTosByConversationId. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message sent previously

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
email	varchar2	Email Address	☐	
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForw ard	char		☐	‘`true` if the message is a for-w arded message.’
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent’s messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	‘`true` if the message w as sent via JavaScript API.’
isTrigger	char		☐	`true` for automatic messages from “Targeted chats & triggers” and API (https://de-velopers.brevo.com/docs/javas-cript-api-refer-ence#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
name	varchar2	Name	☐	
raw UnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In tw o-w ay integrations, messages sent via REST API can be marked w ith receivedFrom property and then filtered out w hen received in a w ebhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents’ messages, `"visitor"` for visitors’ messages.
visitorId	varchar2		☐	visitor’s ID

ConversationsMessages_AttachmentsById

Get a messageRetrieve a single message by its ID.

Catalog: Brevo

Schema: Conversations

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
imageInfo_height	int64		☐	Height of the image in pixels.
imageInfo_previewLink	varchar2		☐	URL of the image preview.
imageInfo_width	int64		☐	Width of the image in pixels.
inlineId	varchar2		☐	The ID of the inline file.
integrations	varchar2		☐	Integration details.
isAllowedFileType	char		☐	Whether the file extension is allowed for download.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'
isImage	char		☐	`true` for images.
isInline	char		☐	`true` for inline files.
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isSticker	char		☐	`true` for sticker files.
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
link	varchar2	Link	☐	The URL of the file.
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
mimeType	varchar2		☐	The MIME type of the file.
name	varchar2	Name	☐	The name of the file.
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
receivedFrom	varchar2		☐	In tw o-w ay integrations, mes-sages sent via REST API can be marked w ith receivedFrom prop-erty and then filtered out w hen received in a w ebhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
size	int64		☐	The size of the file in bytes.
source	varchar2		☐	Source information for the mes-sage.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email mes-sage (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the at-tached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationsMessagesById

Get a messageRetrieve a single message by its ID.

Catalog: Brevo

Schema: Conversations

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 Conversa-tionsMessagesById. 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 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message was sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be marked with receivedFrom property and then filtered out when received in a webhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationsPushedMessages_AttachmentsById

Get an automated messageRetrieve a single automated (pushed) message by its ID.

Catalog: Brevo

Schema: Conversations

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message sent previously

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
imageInfo_height	int64		☐	Height of the image in pixels.
imageInfo_previewLink	varchar2		☐	URL of the image preview.
imageInfo_width	int64		☐	Width of the image in pixels.
inlineId	varchar2		☐	The ID of the inline file.
integrations	varchar2		☐	Integration details.
isAllowedFileType	char		☐	Whether the file extension is allowed for download.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	‘`true` if the message is a forwarded message.’
isImage	char		☐	`true` for images.
isInline	char		☐	`true` for inline files.
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent’s messages in finished chats.
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	‘`true` if the message was sent via JavaScript API.’
isSticker	char		☐	`true` for sticker files.
isTrigger	char		☐	`true` for automatic messages from “Targeted chats & triggers” and API (https://developers.brevo.com/docs/javascript-api-reference#sendautomessage)
link	varchar2	Link	☐	The URL of the file.
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
mimeType	varchar2		☐	The MIME type of the file.
name	varchar2	Name	☐	The name of the file.
rawUnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In two-way integrations, messages sent via REST API can be

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				marked w ith receivedFrom property and then filtered out w hen received in a w ebhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
size	int64		☐	The size of the file in bytes.
source	varchar2		☐	Source information for the message.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email message (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the attached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

ConversationsPushedMessagesById

Get an automated messageRetrieve a single automated (pushed) message by its ID.

Catalog: Brevo

Schema: Conversations

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		ID of the message sent previously

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
agentId	varchar2		☐	ID of the agent on whose behalf the message was sent (only in messages sent by an agent).
agentName	varchar2		☐	Agent's name as displayed to the visitor. Only in the messages sent by an agent.
agentUserpic	varchar2		☐	Only set if the agent has uploaded a profile picture.
createdAt	int64	Creation Date	☐	Timestamp in milliseconds.
file_imageInfo_height	int64		☐	Height of the image in pixels.
file_imageInfo_previewLink	varchar2		☐	URL of the image preview.
file_imageInfo_width	int64		☐	Width of the image in pixels.
file_inlineId	varchar2		☐	The ID of the inline file.
file_isAllowedFileType	char		☐	Whether the file extension is allowed for download.
file_isImage	char		☐	Whether the file is an image.
file_isInline	char		☐	Whether the file is inline.
file_isSticker	char		☐	Whether the file is a sticker.
file_link	varchar2		☐	URL of the file.
file_mimeType	varchar2		☐	MIME type of the file.
file_name	varchar2		☐	Name of the file.
file_size	int64		☐	Size in bytes.
forwardedToSourceStatus_error	varchar2		☐	
forwardedToSourceStatus_isSuccess	char		☐	
from_email	varchar2		☐	
from_name	varchar2		☐	
html	varchar2		☐	The HTML content of the message.
id	varchar2	ID	☐	Message ID. It can be used for further manipulations with the message.
integrations	varchar2		☐	Integration details.
isBot	char		☐	`true` for automated messages generated by an AI bot.
isForward	char		☐	'`true` if the message is a forwarded message.'
isMissed	char		☐	`true` for missed and offline messages.
isMissedByVisitor	char		☐	`true` for unread agent's messages in finished chats.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
isPushed	char		☐	`true` for pushed messages
isSentViaJsApi	char		☐	'`true` if the message w as sent via JavaScript API.'
isTrigger	char		☐	`true` for automatic messages from "Targeted chats & triggers" and API (https://de-velopers.brevo.com/docs/javas-cript-api-refer-ence#sendautomessage)
messageType	varchar2		☐	The type of message content (e.g. for integration-specific message types).
raw UnsafeHtml	varchar2		☐	Unescaped HTML content of the message (may include unsafe HTML).
receivedFrom	varchar2		☐	In tw o-w ay integrations, mes-sages sent via REST API can be marked w ith receivedFrom prop-erty and then filtered out w hen received in a w ebhook to avoid infinite loop.
replyTo_email	varchar2		☐	
replyTo_name	varchar2		☐	
source	varchar2		☐	Source information for the mes-sage.
sourceMessageId	varchar2		☐	The ID of the message assigned by the integration source.
subject	varchar2	Subject	☐	The subject line of the email mes-sage (only for messages sent to email threads).
text	varchar2	Text	☐	Message text or name of the at-tached file
type	varchar2	Type	☐	`"agent"` for agents' messages, `"visitor"` for visitors' messages.
visitorId	varchar2		☐	visitor's ID

3.1.6 Schema: Coupons

3.1.6.1 Tables

CouponCollections

Get all your coupon collectionsRetrieve a paginated list of all coupon collections in your Brevo account. Results can be sorted by creation date, remaining coupons count, or expira-tion date, in ascending or descending order. Pagination defaults to 50 collections per page (maximum 100).

Catalog: Brevo

Schema: Coupons

Primary Keys: id

Retrieve: true

Insert: true

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Update: false

Delete: false

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CouponCollections. 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
sortBy		☐		The field used to sort coupon collections (Values: createdAt, remainingCoupons, expirationDate)

Columns of Table Function

The columns of the table function CouponCollections 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
createdAt	datetime	Creation Date	☐	Datetime on which the collection was created.
defaultCoupon	varchar2		☐	The default coupon of the collection.
expirationDate	datetime		☐	Expiration date for the coupon collection in RFC3339 format.
id	varchar2	ID	☐	The id of the collection.
name	varchar2	Name	☐	The name of the collection.
remainingCoupons	int64		☐	Number of coupons that have not been sent yet.
remainingCouponsAlert	int64		☐	If present, an email notification is going to be sent when the total number of available coupons falls below the defined threshold.
remainingDaysAlert	int64		☐	If present, an email notification is going to be sent the defined amount of days before to the expiration date.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
totalCoupons	int64		☐	Total number of coupons in the collection.

CouponsById

Get a coupon collection by idRetrieve the details of a single coupon collection by its UUID. The response includes the collection name, default coupon value, total and remaining coupon counts, and creation timestamp. Returns a `404` error if no collection matches the provided ID.

Catalog: Brevo

Schema: Coupons

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Id of the collection to return

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Datetime on w hich the collection w as created.
defaultCoupon	varchar2		☐	The default coupon of the collection.
expirationDate	datetime		☐	Expiration date for the coupon collection in RFC3339 format.
id	varchar2	ID	☐	The id of the collection.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
name	varchar2	Name	☐	The name of the collection.
remainingCoupons	int64		☐	Number of coupons that have not been sent yet.
remainingCouponsAlert	int64		☐	If present, an email notification is going to be sent w hen the total number of available coupons falls below the defined threshold.
remainingDaysAlert	int64		☐	If present, an email notification is going to be sent the defined amount of days before to the expiration date.
totalCoupons	int64		☐	Total number of coupons in the collection.

3.1.7 Schema: CustomObjects

3.1.7.1 Tables

records

Get the list of object records and total records count for an object.Custom objects are only available to Enterprise plans.This feature is in beta. These are subject to change.This API retrieves a list of object records along with their associated records and provides the total count of records for the specified object. **Note**: Contact as object type is not supported in this endpoint.

Catalog: Brevo

Schema: CustomObjects

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 records. 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
association		☐		Whether to include associations, must be 'true' or 'false'. Default to

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				'false' if not provided. (Values: true, false)
object_type		☑		Object type for the records to retrieve. Must be a previously created custom object type. Contact as object type is not supported in this endpoint.
page_num		☑		Page number for pagination. It's used to fetch the object records on a provided page number. Must be a valid positive integer.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Key-value pairs of attribute data for the object record. Each key is the attribute **key** (e.g., `company_name`, `engine_type`), not the attribute label (e.g., "Company Name", "Engine Type"). Only attributes that already exist in the object schema will be present.For `category` or `multiple_category` attributes, the value is the option **key** (not the option label or option ID).
createdAt	datetime	Creation Date	☐	Timestamp when the object record was created
identifiers_ext_id	varchar2		☐	External ID of the record in your system.
identifiers_id	int64		☐	Internal ID of the object record generated by Brevo.
updatedAt	datetime		☐	Timestamp when the object record was last updated

records_RecordsAssociations

Get the list of object records and total records count for an object.Custom objects are only available to Enterprise plans.This feature is in beta. These are subject to change.This API retrieves a list of object records along with their associated records and provides the total count of records for the specified object. **Note**: Contact as object type is not supported in this endpoint.

Catalog: Brevo

Schema: CustomObjects

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function records_RecordsAssociations. 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
association		☐		Whether to include associations, must be 'true' or 'false'. Default to 'false' if not provided. (Values: true, false)
object_type		☑		Object type for the records to retrieve. Must be a previously created custom object type. Contact as object type is not supported in this endpoint.
page_num		☑		Page number for pagination. It's used to fetch the object records on a provided page number. Must be a valid positive integer.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Key-value pairs of attribute data for the object record. Each key is the attribute **key** (e.g., `company_name`, `engine_type`), not the attribute label (e.g., "Company Name", "Engine Type"). Only attributes that already exist in the object schema w ill be present.For `category` or `multiple_category` attributes, the value is the option **key** (not the option label or option ID).
createdAt	datetime	Creation Date	☐	Timestamp w hen the object record w as created
identifiers_ext_id	varchar2		☐	External ID of the record in your system.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
identifiers_id	int64		☐	Internal ID of the object record generated by Brevo.
object_type	varchar2		☐	Type of the associated object
updatedAt	datetime		☐	Timestamp when the object record was last updated

3.1.8 Schema: Deals

3.1.8.1 Tables

Crm_ItemsLinkedCompaniesIdsDeals

Get all dealsRetrieve a paginated list of deals with optional filtering, sorting, and search capabilities. Results can be filtered by attributes such as deal name or owner, linked companies, linked contacts, or modification/creation timestamps. Default sort order is descending by creation date.

Catalog: Brevo

Schema: Deals

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 Crm_ItemsLinkedCompaniesIdsDeals. 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		☐		Filter (urlencoded) the deals 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.
filters[attributes.deal_name]		☐		Filter by attributes. If you have a filter for the owner on your end, please send it as filters[attributes.deal_owner] and utilize the account email for the filtering.
filters[linkedCompaniesIds]		☐		Filter by linked companies ids

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
filters[linkedContactsIds]		☐		Filter by linked contacts ids
modifiedSince		☐		Filter (urlencoded) the deals 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.
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Deal attributes with values
id	varchar2	ID	☐	Unique deal id
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this deal
TEXT	varchar2	Text	☐	

Crm_ItemsLinkedContactsIdsDeals

Get all dealsRetrieve a paginated list of deals with optional filtering, sorting, and search capabilities. Results can be filtered by attributes such as deal name or owner, linked companies, linked contacts, or modification/creation timestamps. Default sort order is descending by creation date.

Catalog: Brevo

Schema: Deals

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 Crm_ItemsLinkedContactsIdsDeals. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		Filter (urlencoded) the deals 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.
filters[attributes.deal_name]		☐		Filter by attributes. If you have a filter for the owner on your end, please send it as filters[attributes.deal_owner] and utilize the account email for the filtering.
filters[linkedCompaniesIds]		☐		Filter by linked companies ids
filters[linkedContactsIds]		☐		Filter by linked contacts ids
modifiedSince		☐		Filter (urlencoded) the deals 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.
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Deal attributes with values
id	varchar2	ID	☐	Unique deal id
TEXT	int64	Text	☐	

CrmAttributes_AttributeOptionsDeals

Get deal attributesRetrieve the list of all attributes defined for deals, including both system-default and custom attributes. Each attribute includes its label, internal name, type, required status, and available options for select-type attributes.

Catalog: Brevo

Schema: Deals

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 CrmAttributes_AttributeOptionsDeals are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
attributeTypeName	varchar2		☐	Type of the attribute
internalName	varchar2		☐	Internal name of the attribute
isRequired	char		☐	Whether this attribute is required
isValueReadonly	char		☐	Whether this attribute value is read-only
key	varchar2	Key	☐	
label	varchar2	Label	☐	Display label of the attribute
value	varchar2	Value	☐	

CrmAttributesDeals

Get deal attributesRetrieve the list of all attributes defined for deals, including both system-default and custom attributes. Each attribute includes its label, internal name, type, required status, and available options for select-type attributes.

Catalog: Brevo

Schema: Deals

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 CrmAttributesDeals are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
attributeTypeName	varchar2		☐	Type of the attribute
internalName	varchar2		☐	Internal name of the attribute
isRequired	char		☐	Whether this attribute is required
isValueReadonly	char		☐	Whether this attribute value is read-only
label	varchar2	Label	☐	Display label of the attribute

CrmDeals

Get all dealsRetrieve a paginated list of deals with optional filtering, sorting, and search capabilities. Results can be filtered by attributes such as deal name or owner, linked companies, linked contacts, or modification/creation timestamps. Default sort order is descending by creation date.

Catalog: Brevo

Schema: Deals

Primary Keys: id

Retrieve: true

Insert: true

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CrmDeals. 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		☐		Filter (urlencoded) the deals 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.
filters[attributes.deal_name]		☐		Filter by attributes. If you have a filter for the ow ner on your end, please send it as filters[attributes.deal_ow ner] and utilize the account email for the filtering.
filters[linkedCompaniesIds]		☐		Filter by linked companies ids
filters[linkedContactsIds]		☐		Filter by linked contacts ids
modifiedSince		☐		Filter (urlencoded) the deals 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.
sortBy		☐		The field used to sort field names.

Columns of Table Function

The columns of the table function CrmDeals 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
attributes	varchar2	Attributes	☐	Deal attributes w ith values
id	varchar2	ID	☐	Unique deal id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this deal

CrmDeals_LinkedCompaniesIdsById

Get a dealRetrieve the full details of a single deal by its identifier, including its attributes, pipeline stage, linked contacts, and linked companies. Returns a 404 error if the deal does not exist.

Catalog: Brevo

Schema: Deals

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Deal attributes w ith values
id	varchar2	ID	☐	Unique deal id
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this deal
TEXT	varchar2	Text	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

CrmDeals_LinkedContactsIdsById

Get a dealRetrieve the full details of a single deal by its identifier, including its attributes, pipeline stage, linked contacts, and linked companies. Returns a 404 error if the deal does not exist.

Catalog: Brevo

Schema: Deals

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Deal attributes w ith values
id	varchar2	ID	☐	Unique deal id
TEXT	int64	Text	☐	

CrmDealsById

Get a dealRetrieve the full details of a single deal by its identifier, including its attributes, pipeline stage, linked contacts, and linked companies. Returns a 404 error if the deal does not exist.

Catalog: Brevo

Schema: Deals

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributes	varchar2	Attributes	☐	Deal attributes w ith values
id	varchar2	ID	☐	Unique deal id
linkedContactsIds	varchar2		☐	Contact ids for contacts linked to this deal

CrmPipeline_StagesDetails

Get pipeline stagesThis endpoint is deprecated. Prefer /crm/pipeline/details/{pipelineID} instead.

Catalog: Brevo

Schema: Deals

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 CrmPipeline_StagesDetails are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
id	varchar2	ID	☐	Stage id
name	varchar2	Name	☐	Stage name
pipeline_name	varchar2		☐	Pipeline Name
pipeline	varchar2		☐	Pipeline id

CrmPipelineDetails

Get pipeline stagesThis endpoint is deprecated. Prefer /crm/pipeline/details/{pipelineID} instead.

Catalog: Brevo

Schema: Deals

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 CrmPipelineDetails are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
pipeline_name	varchar2		☐	Pipeline Name
pipeline	varchar2		☐	Pipeline id

CrmPipelineDetails_StagesByPipelineID

Get a pipelineRetrieve the details of a specific deal pipeline by its identifier, including its stages, stage ordering, and configuration. Use this endpoint to obtain the pipeline and stage IDs needed when creating or updating deals.

Catalog: Brevo

Schema: Deals

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 CrmPipelineDetails_StagesByPipelineID. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
pipelineID		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
id	varchar2	ID	☐	Stage id
name	varchar2	Name	☐	Stage name
pipeline_name	varchar2		☐	Pipeline Name
pipeline	varchar2		☐	Pipeline id

CrmPipelineDetailsByPipelineID

Get a pipelineRetrieve the details of a specific deal pipeline by its identifier, including its stages, stage ordering, and configuration. Use this endpoint to obtain the pipeline and stage IDs needed when creating or updating deals.

Catalog: Brevo

Schema: Deals

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 CrmPipelineDetailsByPipelineID. 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
pipelineID		☑		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
pipeline_name	varchar2		☐	Pipeline Name
pipeline	varchar2		☐	Pipeline id

Pipelines

Get all pipelinesRetrieve the list of all deal pipelines configured for your account, including each pipeline''s stages and settings. If no pipelines have been configured yet, a default pipeline is automatically created and returned.

Catalog: Brevo

Schema: Deals

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 Pipelines are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
pipeline_name	varchar2		☐	Pipeline Name
pipeline	varchar2		☐	Pipeline id

PipelineStages

Get all pipelinesRetrieve the list of all deal pipelines configured for your account, including each pipeline''s stages and settings. If no pipelines have been configured yet, a default pipeline is automatically created and returned.

Catalog: Brevo

Schema: Deals

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 PipelineStages are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	varchar2	ID	☐	Stage id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
name	varchar2	Name	☐	Stage name
pipeline_name	varchar2		☐	Pipeline Name
pipeline	varchar2		☐	Pipeline id

3.1.9 Schema: Domains

3.1.9.1 Tables

DomainByName

Get a sender domain configurationRetrieves configuration of a specific domain, to know if the domain is valid or not.Use this to:- Check domain configuration- Validate a domain configuration- Monitor DNS record status- Troubleshoot authentication issuesKey information returned:- Domain verification and authentication status- DNS records configuration and validation status- Detailed authentication requirements

Catalog: Brevo

Schema: Domains

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 DomainByName. 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
domainName		☑		Domain name

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authenticated	char		☐	Status of domain authentication (true=authenticated, false=no-nauthenticated)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
dns_records_brevo_code_host_-name	varchar2		☐	DNS record hostname
dns_records_brevo_code_status	char		☐	Whether the record is properly configured
dns_records_brevo_code_type	varchar2		☐	DNS record type
dns_records_brevo_code_value	varchar2		☐	DNS record value
dns_records_dkim_record_host_-name	varchar2		☐	DNS record hostname
dns_records_dkim_record_status	char		☐	Whether the record is properly configured
dns_records_dkim_record_type	varchar2		☐	DNS record type
dns_records_dkim_record_value	varchar2		☐	DNS record value
dns_records_dmarc_re-cord_host_name	varchar2		☐	DNS record hostname
dns_records_dmarc_re-cord_status	char		☐	Whether the record is properly configured
dns_records_dmarc_record_type	varchar2		☐	DNS record type
dns_records_dmarc_record_value	varchar2		☐	DNS record value
domain	varchar2	Domain	☐	Domain name
verified	char		☐	Status of domain verification (true=verified, false=nonverified)

Domains

Get the list of all sender domainsRetrieves all domains associated with the account.Use this to:- List all domains- Verify domain existence- Check domain authentication and verification status- Monitor domain configuration and provider information- Review domain creation history and ownershipKey information returned:- Domain details (ID, name, authentication status)- Verification and authentication states- Associated IP addresses and DNS providers- Creator information and creation timestamps- Pagination information for large domain lists

Catalog: Brevo

Schema: Domains

Retrieve: true

Insert: true

Update: false

Delete: false

Table Columns

The columns of the table Domains 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
count	int64	Count	☐	Total number of domains
current_page	int64		☐	Current page number

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
total_pages	int64		☐	Total number of pages

Domains_Domains

Get the list of all sender domainsRetrieves all domains associated with the account.Use this to:- List all domains- Verify domain existence- Check domain authentication and verification status- Monitor domain configuration and provider information- Review domain creation history and ownershipKey information returned:- Domain details (ID, name, authentication status)- Verification and authentication states- Associated IP addresses and DNS providers- Creator information and creation timestamps- Pagination information for large domain lists

Catalog: Brevo

Schema: Domains

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 Domains_Domains are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
authenticated	char		☐	Status of domain authentication (true=authenticated,false=non authenticated)
count	int64	Count	☐	Total number of domains
creator_creationDate	datetime		☐	Domain creation date
creator_email	varchar2		☐	Creator's email address
creator_id	varchar2		☐	Creator's user ID
current_page	int64		☐	Current page number
domain_name	varchar2		☐	Domain name
id	varchar2	ID	☐	Unique identifier of the domain
ip	varchar2	IP	☐	Dedicated IP associated w ith domain (null if none)
provider	varchar2		☐	DNS provider for the domain
total_pages	int64		☐	Total number of pages
verified	char		☐	Status of domain verification (true=verified, false=nonverified)

3.1.10 Schema: Ecommerce

3.1.10.1 Tables

Categories

Return all your categoriesRetrieve a paginated list of all ecommerce categories stored in your Brevo account. Results are sorted by creation date in descending order by default, and can be filtered by category IDs, name, modification date, creation date, or deletion status.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The response includes a `count` field with the total number of matching categories, and pagination defaults to 50 categories per page (maximum 100).

Catalog: Brevo

Schema: Ecommerce

Primary Keys: 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 Categories. 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		☐		Filter (urlencoded) the categories 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.**
isDeleted		☐		Filter categories by their deletion status. If `false` is passed, only categories that are not deleted will be returned.
modifiedSince		☐		Filter (urlencoded) the categories 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.**
name		☐		Filter by category name

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function Categories 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
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the category (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	varchar2	ID	☐	Category ID for which you requested the details
isDeleted	char		☐	category deleted from the shop's database
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the category (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the category for which you requested the details
url	varchar2	URL	☐	URL to the category

CategoryById

Get a category detailsRetrieve the full details of a single ecommerce category by its unique ID. The response includes the category name, URL, creation and modification timestamps, and deletion status. Returns a `404` error if no category matches the provided ID.

Catalog: Brevo

Schema: Ecommerce

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Category ID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the category (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	varchar2	ID	☐	Category ID for which you requested the details
isDeleted	char		☐	category deleted from the shop's database
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the category (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the category for which you requested the details
url	varchar2	URL	☐	URL to the category

EcommerceAttribution_ResultsMetrics

Get attribution metrics for one or more Brevo campaigns or workflowsRetrieve aggregated ecommerce attribution metrics for one or more Brevo email campaigns, SMS campaigns, or automation workflows. You can optionally filter by a date range using `periodFrom` and `periodTo` in RFC3339 format. The response includes per-source metrics (orders count, revenue, and average basket) as well as aggregated totals across all requested sources.

Catalog: Brevo

Schema: Ecommerce

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 EcommerceAttribution_ResultsMetrics. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
periodFrom		☐		When getting metrics for a specific period, define the starting datetime in RFC3339 format
periodTo		☐		When getting metrics for a specific period, define the end datetime in RFC3339 format

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
averageBasket	double		☐	
conversionSource	varchar2		☐	
id	decimal	ID	☐	
ordersCount	decimal		☐	
revenue	double		☐	
totals_averageBasket	double		☐	
totals_ordersCount	decimal		☐	
totals_revenue	double		☐	

EcommerceAttributionMetrics

Get attribution metrics for one or more Brevo campaigns or workflowsRetrieve aggregated ecommerce attribution metrics for one or more Brevo email campaigns, SMS campaigns, or automation workflows. You can optionally filter by a date range using `periodFrom` and `periodTo` in RFC3339 format. The response includes per-source metrics (orders count, revenue, and average basket) as well as aggregated totals across all requested sources.

Catalog: Brevo

Schema: Ecommerce

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 EcommerceAttributionMetrics. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
periodFrom		☐		When getting metrics for a specific period, define the starting datetime in RFC3339 format
periodTo		☐		When getting metrics for a specific period, define the end datetime in RFC3339 format

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
totals_averageBasket	double		☐	
totals_ordersCount	decimal		☐	
totals_revenue	double		☐	

EcommerceAttributionMetricsByConversionSourceConversionSourceId

Get detailed attribution metrics for a single Brevo campaign or workflowRetrieve detailed attribution metrics for a single Brevo campaign or automation workflow, identified by its conversion source type and ID. The response includes orders count, revenue, average basket value, and the number of new customers attributed to that specific campaign or workflow.

Catalog: Brevo

Schema: Ecommerce

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 EcommerceAttributionMetricsByConversionSourceConversionSourceId. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
conversionSource		☑		The Brevo campaign type or w orkflow type for w hich data w ill be retrieved (Values: email_campaign, sms_campaign, automation_w orkflow _email, automation_w orkflow _sms)
conversionSourceId		☑		The Brevo campaign or automation w orkflow id for w hich data w ill be retrieved

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
averageBasket	double		☐	
conversionSource	varchar2		☐	
id	decimal	ID	☐	
new CustomersCount	decimal		☐	
ordersCount	decimal		☐	
revenue	double		☐	

EcommerceAttributionProductsByConversionSourceConversionSourceId

Get attributed product sales for a single Brevo campaign or workflowRetrieve the list of products whose sales have been attributed to a specific Brevo campaign or automation workflow. Each product entry includes its ID, name, SKU, image URL, product URL, price, revenue, and orders count. The conversion source type must be one of `email_campaign`, `sms_campaign`, `automation_workflow_email`, or `automation_workflow_sms`.

Catalog: Brevo

Schema: Ecommerce

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 EcommerceAttributionProductsByConversionSourceConversionSourceId. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
conversionSource		☑		The Brevo campaign or automation workflow type for which data will be retrieved (Values: email_campaign, sms_campaign, automation_workflow_email, automation_workflow_sms)
conversionSourceId		☑		The Brevo campaign or automation workflow id for which data will be retrieved

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
id	varchar2	ID	☐	
imageUrl	varchar2		☐	
name	varchar2	Name	☐	
ordersCount	int64		☐	
price	double		☐	
revenue	double		☐	
sku	varchar2		☐	
url	varchar2	URL	☐	

EcommerceDisplayCurrencies

Get the ISO 4217 compliant display currency code for your Brevo accountRetrieve the ISO 4217 display currency code currently configured for your Brevo ecommerce account. This currency is used to display monetary values across the ecommerce dashboard and reports. Returns a `403` error if ecommerce is not activated on the account.

Catalog: Brevo

Schema: Ecommerce

Retrieve: true

Insert: true

Update: false

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Delete: false

Table Columns

The columns of the table EcommerceDisplayCurrencies 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
code	varchar2		☐	ISO 4217 compliant display currency code

Orders

Get order detailsGet all the orders

Catalog: Brevo

Schema: Ecommerce

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 Orders. 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		☐		Filter (urlencoded) the orders 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		☐		Filter (urlencoded) the orders 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.**

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DUMMY	varchar2(1)	Dummy	☑	Default column added since the specification specifies that no data is returned.

ProductById

Get a product's detailsRetrieve the full details of a single ecommerce product by its unique ID. The response includes the product name, price, SKU, URL, image URLs (original and thumbnails), categories, stock level, meta information, creation and modification timestamps, and deletion status. Returns a `404` error if no product matches the provided ID.

Catalog: Brevo

Schema: Ecommerce

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Product ID

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
alternativePrice	double		☐	Alternative price of the product

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
brand	varchar2(128)		☐	Brand of the product
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the product (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2(3000)	Description	☐	Description of the product
id	varchar2	ID	☐	Product ID for which you requested the details
imageUrl	varchar2		☐	Absolute URL to the cover image of the product
isDeleted	char		☐	Product deleted from the shop's database but not from Brevo
metaInfo	varchar2		☐	Meta data of product such as description, vendor, producer, stock level, etc.
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the product (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the product for which you requested the details
parentId	varchar2		☐	Parent product id of the product
price	double		☐	Price of the product
s3Original	varchar2		☐	S3 url of original image
s3ThumbAnalytics	varchar2		☐	S3 thumbnail url of original image in 120x120 dimension for analytics section
s3ThumbEditor	varchar2		☐	S3 thumbnail url of original image in 600x400 dimension for editor section
sku	varchar2		☐	Product identifier from the shop
stock	decimal		☐	Current stock value of the product from the shop's database
url	varchar2	URL	☐	URL to the product

ProductInfo_Categories

Get a product's detailsRetrieve the full details of a single ecommerce product by its unique ID. The response includes the product name, price, SKU, URL, image URLs (original and thumbnails), categories, stock level, meta information, creation and modification timestamps, and deletion status. Returns a `404` error if no product matches the provided ID.

Catalog: Brevo

Schema: Ecommerce

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Product ID

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
alternativePrice	double		☐	Alternative price of the product
brand	varchar2(128)		☐	Brand of the product
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the product (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2(3000)	Description	☐	Description of the product
id	varchar2	ID	☐	Product ID for which you requested the details
imageUrl	varchar2		☐	Absolute URL to the cover image of the product
isDeleted	char		☐	Product deleted from the shop's database but not from Brevo
metaInfo	varchar2		☐	Meta data of product such as description, vendor, producer, stock level, etc.
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the product (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the product for which you requested the details
parentId	varchar2		☐	Parent product id of the product
price	double		☐	Price of the product
s3Original	varchar2		☐	S3 url of original image
s3ThumbAnalytics	varchar2		☐	S3 thumbnail url of original image in 120x120 dimension for analytics section

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
s3ThumbEditor	varchar2		☐	S3 thumbnail url of original image in 600x400 dimension for editor section
sku	varchar2		☐	Product identifier from the shop
stock	decimal		☐	Current stock value of the product from the shop's database
TEXT	varchar2	Text	☐	
url	varchar2	URL	☐	URL to the product

Products

Return all your productsRetrieve a paginated list of all ecommerce products stored in your Brevo account. Results are sorted by creation date in descending order by default, and can be filtered by product IDs, name (minimum 3 characters), price range, category IDs, modification date, creation date, or deletion status. Use the `search` parameter to query across SKU, name, and ID simultaneously — results are prioritized as exact SKU match > SKU prefix match > name match > ID match. Pagination defaults to 50 products per page (maximum 1000), and the response includes a `count` field with the total number of matching products.

Catalog: Brevo

Schema: Ecommerce

Primary Keys: 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 Products. 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
alternativePrice[eq]		☐		Alternative price filter for products equals to particular

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				amount
alternativePrice[gt]		☐		Alternative price filter for products greater than particular amount
alternativePrice[gte]		☐		Alternative price filter for products greater than and equals to particular amount
alternativePrice[lt]		☐		Alternative price filter for products less than particular amount
alternativePrice[lte]		☐		Alternative price filter for products less than and equals to particular amount
alternativePrice[ne]		☐		Alternative price filter for products not equals to particular amount
createdSince		☐		Filter (urlencoded) the products 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.**
isDeleted		☐		Filter products by their deletion status. If `false` is passed, only products that are not deleted w ill be returned.
modifiedSince		☐		Filter (urlencoded) the products 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.**
name		☐		Filter by product name, minimum 3 characters should be present for search.
price[eq]		☐		Price filter for products equals to particular amount
price[gt]		☐		Price filter for products greater than particular amount
price[gte]		☐		Price filter for products greater than and equals to particular amount
price[lt]		☐		Price filter for products less than particular amount
price[lte]		☐		Price filter for products less than and equals to particular amount
price[ne]		☐		Price filter for products not equals to particular amount
search		☐		Search products simultaneously across SKU, name, and ID fields. Results are returned in the fol-low ing priority order: **exact SKU match** > **SKU prefix match** > **name match** > **ID match**. For example, `?

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				search=123` on products with `{sku: "123"}` and `{sku: "123456"}` returns the exact SKU match first.
sortByField		☐		Sort the results by a specific field. Default sort field is `created_at` when not passed.(Values: created_at, name, price, id)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
alternativePrice	double		☐	Alternative price of the product
brand	varchar2(128)		☐	Brand of the product
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the product (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2(3000)	Description	☐	Description of the product
id	varchar2	ID	☐	Product ID for which you requested the details
imageUrl	varchar2		☐	Absolute URL to the cover image of the product
isDeleted	char		☐	Product deleted from the shop's database but not from Brevo
metaInfo	varchar2		☐	Meta data of product such as description, vendor, producer, stock level, etc.
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the product (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the product for which you requested the details
parentId	varchar2		☐	Parent product id of the product
price	double		☐	Price of the product
s3Original	varchar2		☐	S3 url of original image
s3ThumbAnalytics	varchar2		☐	S3 thumbnail url of original image in 120x120 dimension for analytics section
s3ThumbEditor	varchar2		☐	S3 thumbnail url of original image in 600x400 dimension for editor section
sku	varchar2		☐	Product identifier from the shop
stock	decimal		☐	Current stock value of the product from the shop's database

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
url	varchar2	URL	☐	URL to the product

3.1.11 Schema: EmailCampaigns

3.1.11.1 Tables

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: Brevo

Schema: EmailCampaigns

Label: A/B Test Campaign Result by ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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		☑		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	varchar2	Clickrate	☐	Click rate for current w inning version
openRate	varchar2	Open Rate	☐	Open rate for current w inning version
statistics_clicks_Version_A	varchar2		☐	percentage of an event for version A

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
statistics_clicks_Version_B	varchar2		☐	percentage of an event for version B
statistics_complaints_Version_A	varchar2		☐	percentage of an event for version A
statistics_complaints_Version_B	varchar2		☐	percentage of an event for version B
statistics_hardBounces_Version_A	varchar2		☐	percentage of an event for version A
statistics_hardBounces_Version_B	varchar2		☐	percentage of an event for version B
statistics_openers_Version_A	varchar2		☐	percentage of an event for version A
statistics_openers_Version_B	varchar2		☐	percentage of an event for version B
statistics_softBounces_Version_A	varchar2		☐	percentage of an event for version A
statistics_softBounces_Version_B	varchar2		☐	percentage of an event for version B
statistics_unsubscribed_Version_A	varchar2		☐	percentage of an event for version A
statistics_unsubscribed_Version_B	varchar2		☐	percentage of an event for version B
winningCriteria	varchar2	Winning Criteria	☐	Criteria chosen for winning version (Open/Click)
winningSubjectLine	varchar2	Winning Subject Line	☐	Subject Line of current winning version
winningVersion	varchar2	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	varchar2	Winning Version Rate	☐	Open/Click rate for the winner version

EmailCampaignById: Sendinblue Email Campaign by ID

Get an email campaign reportRetrieve detailed information about a specific email campaign by its ID, including recipients, statistics, and HTML content. Use the statistics query parameter to select which statistics to include (globalStats, linksStats, statsByDomain, statsByDevice, or statsByBrowser); statsByDevice and statsByBrowser are only available on this single-campaign endpoint. You can exclude HTML content from the response by setting excludeHtmlContent to true.

Catalog: Brevo

Schema: EmailCampaigns

Label: Email Campaign by ID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		Id of the campaign
excludeHtmlContent		☐		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 (Values: true, false)
statistics		☐		Filter on the type of statistics required. Example: **globalStats** value will only fetch globalStats info of the campaign in the returned response. `statsByDevice` and `statsByBrowser` are only available when retrieving a single campaign (not in the list endpoint). (Values: 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	char	A/B-Testing	☐	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
attachmentFile	varchar2		☐	Url of the attachment file. Only available if the campaign has an attachment.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
attachmentUrl	varchar2		☐	URL of the attachment file associated w ith the campaign. Empty string if no attachment is present.
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
emailExpirationDate_duration	int64		☐	Duration of the email expiry
emailExpirationDate_unit	varchar2		☐	Unit of the duration
footer	varchar2	Footer	☐	Footer of the campaign
header	varchar2	Header	☐	Header of the campaign
htmlContent	varchar2	HTML Content	☐	HTML content of the campaign
id	int64	ID	☐	ID of the campaign
inlineImageActivation	char	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	char	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	varchar2	Modified Date	☐	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the campaign
preview Text	varchar2		☐	Preview text or preheader of the email campaign
recipients	varchar2	Recipients	☐	
recurring	char	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	varchar2	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	varchar2	Schedule Date	☐	UTC date-time on w hich campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	char	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	varchar2	Sender Email Address	☐	Sender email of the campaign
sender_id	int64	Sender ID	☐	Sender id of the campaign

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
sender_name	varchar2	Sender Name	☐	Sender name of the campaign
sentDate	varchar2	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	varchar2	Share Link	☐	Link to share the campaign on social media. For trigger campaigns, this returns a descriptive message instead of a URL. For classic campaigns that have not been sent, this also returns a descriptive message.
splitRule	int64	Split Rule	☐	The size of your ab-test groups. Only available if `abTesting` flag of the campaign is `true`
statistics	varchar2	Statistics	☐	
status	varchar2	Status	☐	Status of the campaign
subject	varchar2	Subject	☐	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
subjectA	varchar2	Subject A	☐	Subject A of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
subjectB	varchar2	Subject B	☐	Subject B of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
tag	varchar2	Tag	☐	Tag of the campaign
testSent	char	Test Sent	☐	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	varchar2	To Field	☐	Customisation of the "to" field of the campaign
type	varchar2	Type	☐	Type of campaign
utmCampaignValue	varchar2		☐	The utm_campaign value associated with the campaign. Only present if a UTM campaign value was set.
utmID	int64		☐	The campaign ID used as utm_id parameter. Only present if UTM campaign tracking with ID is enabled.
utmMedium	varchar2		☐	The utm_medium value. Set to "EMAIL" when UTM campaign tracking is enabled.
utmSource	varchar2		☐	The utm_source value. Set to "Brevo" when UTM campaign tracking is enabled.
winnerCriteria	varchar2	Winner Criteria	☐	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
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`

EmailCampaigns: Sendinblue Email Campaigns

Return all your created email campaignsThe response payload for this endpoint has changedYou now need to specify which type of statistics you would like to retrieve. For more information visit [this page](https://developers.brevo.com/changelog/get-all-marketing-campaigns).

Catalog: Brevo

Schema: EmailCampaigns

Primary Keys: id

Label: Email Campaigns

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function EmailCampaigns. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a 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		☐		**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 `status` is not passed or is set to `sent`. The date range between `startDate` and `endDate` must not exceed 2 years. `endDate` must not be in the future.
excludeHtmlContent		☐		Use this flag to exclude htmlContent from the response body. If

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				set to **true**, the htmlContent field w ill be returned as an empty string in the response body. (Values: true, false)
excludePdfAttachment		☐		Use this flag to filter out campaigns that have a PDF attachment. If set to **true**, only campaigns w ithout a PDF attachment (or w ith no attachment at all) w ill be returned. (Values: true, false)
startDate		☐		**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 `status` is not passed or is set to `sent`. The date range betw een `startDate` and `endDate` must not exceed 2 years. `startDate` must not be in the future.
statistics		☐		Filter on the type of statistics required. Example: **globalStats** value w ill only fetch globalStats info of the campaign in the returned response. This option only returns data for events that occurred in the last 6 months. For older campaigns, it is advisable to use the **Get Campaign Report** endpoint. (Values: globalStats, linksStats, statsByDomain)
status		☐		Filter on the status of the campaign (Values: suspended, archive, sent, queued, draft, inProcess, inReview , cancelling, cancelled)
type		☐		Filter on the type of the campaigns (Values: 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	char	A/B-Testing	☐	Status of A/B Test for the campaign. abTesting = false means it is disabled, & abTesting = true means it is enabled.
attachmentFile	varchar2		☐	Url of the attachment file. Only available if the campaign has an

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				attachment.
attachmentUrl	varchar2		☐	URL of the attachment file associated w ith the campaign. Empty string if no attachment is present.
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
emailExpirationDate_duration	int64		☐	Duration of the email expiry
emailExpirationDate_unit	varchar2		☐	Unit of the duration
footer	varchar2	Footer	☐	Footer of the campaign
header	varchar2	Header	☐	Header of the campaign
htmlContent	varchar2	HTML Content	☐	HTML content of the campaign
id	int64	ID	☐	ID of the campaign
inlineImageActivation	char	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	char	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	varchar2	Modified Date	☐	UTC date-time of last modification of the campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the campaign
preview Text	varchar2		☐	Preview text or preheader of the email campaign
recipients	varchar2	Recipients	☐	
recurring	char	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	varchar2	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	varchar2	Schedule Date	☐	UTC date-time on w hich campaign is scheduled (YYYY-MM-DDTHH:mm:ss.SSSZ)
sendAtBestTime	char	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	varchar2	Sender Email Address	☐	Sender email of the campaign

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
sender_id	int64	Sender ID	☐	Sender id of the campaign
sender_name	varchar2	Sender Name	☐	Sender name of the campaign
sentDate	varchar2	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	varchar2	Share Link	☐	Link to share the campaign on social media. For trigger campaigns, this returns a descriptive message instead of a URL. For classic campaigns that have not been sent, this also returns a descriptive message.
splitRule	int64	Split Rule	☐	The size of your ab-test groups. Only available if `abTesting` flag of the campaign is `true`
statistics	varchar2	Statistics	☐	
status	varchar2	Status	☐	Status of the campaign
subject	varchar2	Subject	☐	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
subjectA	varchar2	Subject A	☐	Subject A of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
subjectB	varchar2	Subject B	☐	Subject B of the ab-test campaign. Only available if `abTesting` flag of the campaign is `true`
tag	varchar2	Tag	☐	Tag of the campaign
testSent	char	Test Sent	☐	Retrieved the status of test email sending. (true=Test email has been sent false=Test email has not been sent)
toField	varchar2	To Field	☐	Customisation of the "to" field of the campaign
type	varchar2	Type	☐	Type of campaign
utmCampaignValue	varchar2		☐	The utm_campaign value associated with the campaign. Only present if a UTM campaign value was set.
utmID	int64		☐	The campaign ID used as utm_id parameter. Only present if UTM campaign tracking with ID is enabled.
utmMedium	varchar2		☐	The utm_medium value. Set to "EMAIL" when UTM campaign tracking is enabled.
utmSource	varchar2		☐	The utm_source value. Set to "Brevo" when UTM campaign tracking is enabled.
winnerCriteria	varchar2	Winner Criteria	☐	Criteria for the winning version. Only available if `abTesting` flag of the campaign is `true`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
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`

SharedTemplateUrlByCampaignId: Sendinblue Shared URL Template by Campaign ID

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

Catalog: Brevo

Schema: EmailCampaigns

Label: Shared URL Template by Campaign ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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		☑		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	varchar2	Shared URL	☐	A unique URL for the email campaign or transactional template. This URL can be shared with other Brevo users.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

3.1.12 Schema: Event

3.1.12.1 Tables

Events

Get eventsRetrieve a list of events filtered by various criteria.

Catalog: Brevo

Schema: Event

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 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
endDate		☐		Mandatory if startDate is used. End of date range (YYYY-MM-DD or RFC3339). Must be ≥ startDate.
startDate		☐		Mandatory if endDate is used. Start of date range (YYYY-MM-DD or RFC3339). Defaults to 6 months ago w hen omitted alongside endDate. Must be ≤ endDate.

Columns of Table Function

The columns of the table function Events 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
contact_id	int64		☐	Contact ID associated w ith the event

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
contact_properties	varchar2		☐	Contact-level properties at time of event
event_date	datetime		☐	Date and time of the event
event_filter_id	varchar2		☐	Filter ID of the event
event_name	varchar2		☐	Name of the event
event_properties	varchar2		☐	Event-level custom properties
object_type	varchar2		☐	Object type associated w ith the event

3.1.13 Schema: ExternalFeeds

3.1.13.1 Tables

ExternalFeedByUUID

Get external feed detailsRetrieves details of a specific external feed by its UUID.**Use this to:**- Get complete configuration of an external feed- Check feed authentication settings- Review feed personalization options- Verify feed URL and parameters- Monitor feed modification history**Key information returned:**- Complete feed configuration and settings- Authentication credentials and headers- Personalization and fallback settings- Creation and modification timestamps- Cache and retry configurations**Important considerations:**- UUID must exist in your account- Provides complete feed information for troubleshooting- Essential before making modifications- Shows current feed health status- Useful for debugging feed issues

Catalog: Brevo

Schema: ExternalFeeds

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 ExternalFeedByUUID. 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
uuid		☑		UUID of the feed to fetch

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authType	varchar2		☐	Authentication type for accessing the feed
cache	char		☐	Whether to cache the feed response
createdAt	datetime	Creation Date	☐	Feed creation timestamp
id	varchar2	ID	☐	Unique identifier of the feed
maxRetries	int64		☐	Maximum number of retry attempts for failed requests
modifiedAt	datetime	Modified Date	☐	Feed last modification timestamp
name	varchar2	Name	☐	Name of the feed
password	varchar2	Password	☐	Password for basic authentication. Only returned when authType is 'basic'. Excluded from response when authType is 'token'.
token	varchar2	Token	☐	Token for token-based authentication. Only returned when authType is 'token'. Excluded from response when authType is 'basic' or 'noAuth'.
url	varchar2	URL	☐	URL of the external data source
username	varchar2		☐	Username for basic authentication. Only returned when authType is 'basic'. Excluded from response when authType is 'token'.

ExternalFeedByUUID_Headers

Get external feed detailsRetrieves details of a specific external feed by its UUID.**Use this to:**- Get complete configuration of an external feed- Check feed authentication settings- Review feed personalization options- Verify feed URL and parameters- Monitor feed modification history**Key information returned:**- Complete feed configuration and settings- Authentication credentials and headers- Personalization and fallback settings- Creation and modification timestamps- Cache and retry configurations**Important considerations:**- UUID must exist in your account- Provides complete feed information for troubleshooting- Essential before making modifications- Shows current feed health status- Useful for debugging feed issues

Catalog: Brevo

Schema: ExternalFeeds

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function ExternalFeedByUUID_Headers. 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		☑		UUID of the feed to fetch

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authType	varchar2		☐	Authentication type for accessing the feed
cache	char		☐	Whether to cache the feed response
createdAt	datetime	Creation Date	☐	Feed creation timestamp
id	varchar2	ID	☐	Unique identifier of the feed
maxRetries	int64		☐	Maximum number of retry attempts for failed requests
modifiedAt	datetime	Modified Date	☐	Feed last modification timestamp
name_2	varchar2		☐	Name of the feed
name	varchar2	Name	☐	Header name
password	varchar2	Password	☐	Password for basic authentication. Only returned when authType is 'basic'. Excluded from response when authType is 'token'.
token	varchar2	Token	☐	Token for token-based authentication. Only returned when authType is 'token'. Excluded from response when authType is 'basic' or 'noAuth'.
url	varchar2	URL	☐	URL of the external data source
username	varchar2		☐	Username for basic authentication. Only returned when authType is 'basic'. Excluded from response when authType is 'token'.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
value	varchar2	Value	☐	Header value

ExternalFeeds

Get all external feedsRetrieves all external feeds from your Brevo account with filtering and pagination.**Use this to:**- Get an overview of all external data feeds- Find feeds by name using search functionality- Filter feeds by creation date range- Browse feeds by authentication type- Monitor feed library organization and usage**Key information returned:**- Feed details (UUID, name, URL, authentication type)- Feed configuration and settings- Creation and modification timestamps- Feed status and error information- Authentication and header configurations**Important considerations:**- External feeds enable dynamic content in email campaigns- Feeds must be accessible from Brevo servers- Authentication credentials are securely stored- Feed performance affects campaign delivery- Use pagination for accounts with many feeds- Date range filtering limited to 30 days maximum- Search functionality works on feed name matching- Internal feeds are system-managed and cannot be modified

Catalog: Brevo

Schema: ExternalFeeds

Primary Keys: id

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 ExternalFeeds. 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
authType		☐		Filter the records by `authType` of the feed. (Values: basic, token, noAuth)
endDate		☐		Mandatory if `startDate` is used. Ending date (YYYY-MM-DD) till w hichyou w ant to fetch the list. Maximum time period that can be selected isone month.
search		☐		Can be used to filter records by search keyw ord on feed name
startDate		☐		Mandatory if `endDate` is used. Starting date (YYYY-MM-DD)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				from w hichyou w ant to fetch the list. Can be maximum 30 days older than currentdate.

Columns of Table Function

The columns of the table function ExternalFeeds 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
authType	varchar2		☐	Authentication type for access-ing the feed
cache	char		☐	Whether to cache the feed re-sponse
createdAt	datetime	Creation Date	☐	Feed creation timestamp
id	varchar2	ID	☐	Unique identifier of the feed
maxRetries	int64		☐	Maximum number of retry at-tempts for failed requests
modifiedAt	datetime	Modified Date	☐	Feed last modification timestamp
name	varchar2	Name	☐	Name of the feed
passw ord	varchar2	Passw ord	☐	Passw ord for basic authentica-tion. Only returned w hen au-thType is 'basic'. Excluded w hen authType is 'token'.
token	varchar2	Token	☐	Token for token-based authentic-ation. Only returned w hen au-thType is 'token'. Excluded w hen authType is 'basic' or 'noAuth'.
url	varchar2	URL	☐	URL of the external data source
username	varchar2		☐	Username for basic authentica-tion. Only returned w hen au-thType is 'basic'. Excluded w hen authType is 'token'.

ExternalFeeds_FeedsHeaders

Get all external feedsRetrieves all external feeds from your Brevo account with filtering and pagination.**Use this to:**- Get an overview of all external data feeds- Find feeds by name using search functionality- Filter feeds by creation date range- Browse feeds by authentica-tion type- Monitor feed library organization and usage**Key information returned:**- Feed de-tails (UUID, name, URL, authentication type)- Feed configuration and settings- Creation and modification timestamps- Feed status and error information- Authentication and header con-figurations**Important considerations:**- External feeds enable dynamic content in email campaigns- Feeds must be accessible from Brevo servers- Authentication credentials are securely stored- Feed performance affects campaign delivery- Use pagination for accounts with many feeds- Date range filtering limited to 30 days maximum- Search functionality works on feed name matching- Internal feeds are system-managed and cannot be modified

Catalog: Brevo

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: ExternalFeeds

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 ExternalFeeds_FeedsHeaders. 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
authType		☐		Filter the records by `authType` of the feed. (Values: basic, token, noAuth)
endDate		☐		Mandatory if `startDate` is used. Ending date (YYYY-MM-DD) till w hichyou w ant to fetch the list. Maximum time period that can be selected isone month.
search		☐		Can be used to filter records by search keyw ord on feed name
startDate		☐		Mandatory if `endDate` is used. Starting date (YYYY-MM-DD) from w hichyou w ant to fetch the list. Can be maximum 30 days older than currentdate.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authType	varchar2		☐	Authentication type for accessing the feed
cache	char		☐	Whether to cache the feed response
createdAt	datetime	Creation Date	☐	Feed creation timestamp
id	varchar2	ID	☐	Unique identifier of the feed

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
maxRetries	int64		☐	Maximum number of retry attempts for failed requests
modifiedAt	datetime	Modified Date	☐	Feed last modification timestamp
name_2	varchar2		☐	Name of the feed
name	varchar2	Name	☐	Header name
password	varchar2	Password	☐	Password for basic authentication. Only returned when authType is 'basic'. Excluded when authType is 'token'.
token	varchar2	Token	☐	Token for token-based authentication. Only returned when authType is 'token'. Excluded when authType is 'basic' or 'noAuth'.
url	varchar2	URL	☐	URL of the external data source
username	varchar2		☐	Username for basic authentication. Only returned when authType is 'basic'. Excluded when authType is 'token'.
value	varchar2	Value	☐	Header value

3.1.14 Schema: Files

3.1.14.1 Tables

CrmFiles

Get all filesRetrieve a paginated list of CRM files with optional filtering by entity type, entity IDs, and date range. Results are sorted by creation date in descending order by default, with a default limit of 50 files per page.

Catalog: Brevo

Schema: Files

Primary Keys: id

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CrmFiles. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
entity		☐		Filter by file entity type (Values: companies, deals, contacts)
entityIds		☐		Filter by file entity IDs

Columns of Table Function

The columns of the table function CrmFiles 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
authorId	varchar2		☐	Account id of user which created the file
companyId	varchar2		☐	Company id linked to a file
contactId	int64		☐	Contact id of contact on which file is uploaded
createdAt	datetime	Creation Date	☐	File created date/time
dealId	varchar2		☐	Deal id linked to a file
id	varchar2	ID	☐	Unique file id
name	varchar2	Name	☐	Name of uploaded file
size	int64		☐	Size of file in bytes

CrmFilesById

Download a fileGet a temporary download URL for a CRM file by its identifier. The returned URL is valid for 5 minutes only and provides direct access to the file content.

Catalog: Brevo

Schema: Files

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 CrmFilesById. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		File id to dow nload.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
fileUrl	varchar2		☐	A unique link to dow nload the requested file.

CrmFilesByIdData

Get file detailsRetrieve the metadata and details of a specific CRM file by its identifier. This returns information such as the file name, size, type, creation date, and associated entities, but does not include the file content itself.

Catalog: Brevo

Schema: Files

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 CrmFilesByIdData. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
id		☑		File id to get file data.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account id of user which created the file
companyId	varchar2		☐	Company id linked to a file
contactId	int64		☐	Contact id of contact on which file is uploaded
createdAt	datetime	Creation Date	☐	File created date/time
dealId	varchar2		☐	Deal id linked to a file
id	varchar2	ID	☐	Unique file id
name	varchar2	Name	☐	Name of uploaded file
size	int64		☐	Size of file in bytes

3.1.15 Schema: InboundParsing

3.1.15.1 Tables

InboundEmailAttachmentByToken

Retrieve inbound attachment with download token.This endpoint will retrieve inbound attachment with download token.

Catalog: Brevo

Schema: InboundParsing

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 InboundEmailAttachmentByToken. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
dow nloadToken		☑		Token to fetch a particular attachment

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
DOCUMENT	blob		☐	

InboundEmailEvents

Get the list of all the events for the received emails.This endpoint will show the list of all the events for the received emails. When no date range is provided, the last 30 days of events are returned by default.

Catalog: Brevo

Schema: InboundParsing

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 InboundEmailEvents. 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		☐		Mandatory if startDate is used. Ending date (YYYY-MM-DD or YYYY-MM-DDTHH:mm:ss.SSSZ) till w hich you w ant to fetch the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				list. Maximum time period that can be selected is 30 days. Must not be in the future.
sender		☐		Email address of the sender.
startDate		☐		Mandatory if endDate is used. Starting date (YYYY-MM-DD or YYYY-MM-DDTHH:mm:ss.SSSZ) from which you want to fetch the list. Maximum time period that can be selected is 30 days. Must not be in the future.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	☐	Date when email was received on SMTP relay
recipient	varchar2		☐	Recipient's email address
sender	varchar2	Sender	☐	Sender's email address
uuid	guid	UUID	☐	UUID that can be used to fetch additional data

InboundEmailEventsByUuid

Fetch all events history for one particular received email.This endpoint will show the list of all events history for one particular received email.

Catalog: Brevo

Schema: InboundParsing

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 InboundEmailEventsByUuid. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		UUID to fetch events specific to received email

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
deliveredAt	datetime		☐	Date when email was delivered successfully to the client's web-hook
messageId	varchar2	Message ID	☐	Value of the Message-ID header. This will be present only after the processing is done.
receivedAt	datetime		☐	Date when email was received on SMTP relay
recipient	varchar2		☐	Recipient's email address
sender	varchar2	Sender	☐	Sender's email address
subject	varchar2	Subject	☐	Value of the Subject header. This will be present only after the processing is done.

InboundEmailEventsByUuid_Attachments

Fetch all events history for one particular received email.This endpoint will show the list of all events history for one particular received email.

Catalog: Brevo

Schema: InboundParsing

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 InboundEmailEventsByUuid_Attachments. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		UUID to fetch events specific to received email

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
contentId	varchar2		☐	value of the Content-ID header of the attachment.
contentLength	int64		☐	size of the attachment in bytes
contentType	varchar2		☐	value of the Content-Type header of the attachment
deliveredAt	datetime		☐	Date when email was delivered successfully to the client's webhook
messageId	varchar2	Message ID	☐	Value of the Message-ID header. This will be present only after the processing is done.
name	varchar2	Name	☐	filename specified in the Content-Disposition header of the attachment
receivedAt	datetime		☐	Date when email was received on SMTP relay
recipient	varchar2		☐	Recipient's email address
sender	varchar2	Sender	☐	Sender's email address
subject	varchar2	Subject	☐	Value of the Subject header. This will be present only after the processing is done.

InboundEmailEventsByUuid_Logs

Fetch all events history for one particular received email.This endpoint will show the list of all events history for one particular received email.

Catalog: Brevo

Schema: InboundParsing

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function InboundEmailEventsByUuid_Logs. 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		☑		UUID to fetch events specific to received email

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	datetime	Date	☐	Date of the event
deliveredAt	datetime		☐	Date when email was delivered successfully to the client's webhook
messageId	varchar2	Message ID	☐	Value of the Message-ID header. This will be present only after the processing is done.
receivedAt	datetime		☐	Date when email was received on SMTP relay
recipient	varchar2		☐	Recipient's email address
sender	varchar2	Sender	☐	Sender's email address
subject	varchar2	Subject	☐	Value of the Subject header. This will be present only after the processing is done.
type	varchar2	Type	☐	Type of the event

3.1.16 Schema: MasterAccount

3.1.16.1 Tables

CorporateGroup_Sub-accountsById

GET a group detailsThis endpoint allows you to retrieve a specific group's information such asthe list of sub-organizations and the user associated with the group.

Catalog: Brevo

Schema: MasterAccount

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Id of the group of sub-organization

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
companyName	varchar2	Company Name	☐	Name of the sub-account organzation
createdAt	varchar2	Creation Date	☐	Creation date of the sub-account organzation
group_createdAt	varchar2		☐	Group creation date
group_groupName	varchar2		☐	Name of the group
group_id	varchar2		☐	Group id
id	int64	ID	☐	Id of the sub-account organzation

CorporateGroup_UsersById

GET a group detailsThis endpoint allows you to retrieve a specific group’s information such asthe list of sub-organizations and the user associated with the group.

Catalog: Brevo

Schema: MasterAccount

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Id of the group of sub-organization

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user
firstName	varchar2	First Name	☐	First name of the user
group_createdAt	varchar2		☐	Group creation date
group_groupName	varchar2		☐	Name of the group
group_id	varchar2		☐	Group id
lastName	varchar2	Last Name	☐	Last name of the user

CorporateGroupById

GET a group detailsThis endpoint allows you to retrieve a specific group’s information such asthe list of sub-organizations and the user associated with the group.

Catalog: Brevo

Schema: MasterAccount

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CorporateGroupById. 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
id		☑		Id of the group of sub-organization

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
group_createdAt	varchar2		□	Group creation date
group_groupName	varchar2		□	Name of the group
group_id	varchar2		□	Group id

CorporateInvitedUserAppsManagement

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUserAppsManagement. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUserMyPlans

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUserMyPlans. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

Name	Data Type	Required	Default Value	Documentation
type		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUsers

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUsers. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_ow ner	varchar2		☐	Flag for indicating is user ow ner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUsersList_UsersFeature_accessAnalytics

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUsersList_UsersFeature_accessAnalytics. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUsersList_UsersFeature_accessApi_keys

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUsersList_UsersFeature_accessApi_keys. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
type		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUsersList_UsersFeature_accessCreate_sub_organizations

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUsersList_UsersFeature_accessCreate_sub_organizations. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
type		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUsersList_UsersFeature_accessManage_sub_organizations

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUsersList_UsersFeature_accessManage_sub_organizations. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
type		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUsersList_UsersFeature_accessSub_organization_groups

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUsersList_UsersFeature_accessSub_organization_groups. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
type		☐		User type (active | pending). This is required if offset is provided forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateInvitedUserUserManagement

Get the list of all admin usersThis endpoint allows you to list all Admin users of your Admin account. Youcan filter users by type (active or pending) and paginate results usingoffset and limit.

Catalog: Brevo

Schema: MasterAccount

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 CorporateInvitedUserUserManagement. 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		☐		User type (active | pending). This is required if offset is provided

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				forlimited result.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
groups_id	varchar2		☐	group id
groups_name	varchar2		☐	group name
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

CorporateIpAddresses

List of all IPsThis endpoint allows you to retrieve the list of active IPs on your Adminaccount

Catalog: Brevo

Schema: MasterAccount

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 CorporateIpAddresses are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
domain	varchar2	Domain	☐	
ip	varchar2	IP	☐	
transactional	char		☐	

CorporateMasterAccounts

Get the details of requested master accountThis endpoint will provide the details of the master account.

Catalog: Brevo

Schema: MasterAccount

Primary Keys: id

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Table Columns

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

Name	Data Type	Label	Required	Documentation
billingInfo_address_countryCode	varchar2		☐	Country code
billingInfo_address_locality	varchar2		☐	Locality
billingInfo_address_postalCode	varchar2		☐	Postal code
billingInfo_address_stateCode	varchar2		☐	State code
billingInfo_address_streetAddress	varchar2		☐	Street address
billingInfo_companyName	varchar2		☐	Company name of master account
billingInfo_email	varchar2		☐	Billing email id of master account
billingInfo_name_familyName	varchar2		☐	Last name for billing
billingInfo_name_givenName	varchar2		☐	First name for billing
companyName	varchar2	Company Name	☐	Company name of master account organization
currencyCode	varchar2		☐	Currency code of the master account organization
email	varchar2	Email Address	☐	Email id of master account
id	int64	ID	☐	Unique identifier of the master account organization
planInfo_currencyCode	varchar2		☐	Plan currency
planInfo_nextBillingAt	int64		☐	Timestamp of next billing date
planInfo_planPeriod	varchar2		☐	Plan period type
planInfo_price	decimal		☐	Plan amount
planInfo_subAccounts	int64		☐	Number of sub-accounts
timezone	varchar2		☐	Timezone of the master account organization

CorporatePlanMasterAccounts

Get the details of requested master accountThis endpoint will provide the details of the master account.

Catalog: Brevo

Schema: MasterAccount

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 CorporatePlanMasterAccounts are shown below. Each column has an SQL data type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
billingInfo_address_countryCode	varchar2		☐	Country code
billingInfo_address_locality	varchar2		☐	Locality
billingInfo_address_postalCode	varchar2		☐	Postal code
billingInfo_address_stateCode	varchar2		☐	State code
billingInfo_address_streetAddress	varchar2		☐	Street address
billingInfo_companyName	varchar2		☐	Company name of master account
billingInfo_email	varchar2		☐	Billing email id of master account
billingInfo_name_familyName	varchar2		☐	Last name for billing
billingInfo_name_givenName	varchar2		☐	First name for billing
companyName	varchar2	Company Name	☐	Company name of master account organization
currencyCode_2	varchar2		☐	Currency code of the master account organization
currencyCode	varchar2		☐	Plan currency
email	varchar2	Email Address	☐	Email id of master account
id	int64	ID	☐	Unique identifier of the master account organization
name	varchar2	Name	☐	Name of the feature
nextBillingAt	int64		☐	Timestamp of next billing date
planPeriod	varchar2		☐	Plan period type
price	decimal		☐	Plan amount
quantity	int64		☐	Quantity provided in the plan
quantityWithOverages	int64		☐	Quantity with overages provided in the plan (onlyapplicable on ENTv2)
remaining	int64	Remaining	☐	Quantity remaining in the plan
subAccounts	int64		☐	Number of sub-accounts
timezone	varchar2		☐	Timezone of the master account organization
unitValue	varchar2		☐	Unit value of the feature
used	int64		☐	Quantity consumed by master
usedOverages	int64		☐	Quantity consumed by sub-organizations over theadmin plan limit (only applicable on ENTv2)

CorporateSubAccountById

Get sub-account detailsThis endpoint will provide the details for the specified sub-account company

Catalog: Brevo

Schema: MasterAccount

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Id of the sub-account organization

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
companyName	varchar2	Company Name	☐	Sub-account company name
email	varchar2	Email Address	☐	Email id of the sub-account organization
name	varchar2	Name	☐	Name of the sub-account user
planInfo_planType	varchar2		☐	type of the plan

CorporateSubAccountGroups

Get the list of all the sub-accounts of the master account.This endpoint will provide the list all the sub-accounts of the masteraccount.

Catalog: Brevo

Schema: MasterAccount

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
active	char	Active	☐	Whether the sub-account is act-ive or not
companyName	varchar2	Company Name	☐	Name of the sub-account com-pany
createdAt	int64	Creation Date	☐	Timestamp w hen the sub-ac-count w as created
id_2	int64		☐	id of the sub-account
id	varchar2	ID	☐	Group identifier
name	varchar2	Name	☐	Name of the group

CorporateSubAccountGroupsBySubAccountId

Get sub-account detailsThis endpoint will provide the details for the specified sub-account company

Catalog: Brevo

Schema: MasterAccount

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 Corpor-ateSubAccountGroupsBySubAccountId. 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 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
id		☑		Id of the sub-account organiza-tion

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
companyName	varchar2	Company Name	☐	Sub-account company name
email	varchar2	Email Address	☐	Email id of the sub-account organization
id	varchar2	ID	☐	Group id
name_2	varchar2		☐	Name of the sub-account user
name	varchar2	Name	☐	Name of the group
planInfo_planType	varchar2		☐	type of the plan

CorporateSubAccounts

Get the list of all the sub-accounts of the master account.This endpoint will provide the list all the sub-accounts of the masteraccount.

Catalog: Brevo

Schema: MasterAccount

Primary Keys: id

Retrieve: true

Insert: true

Update: false

Delete: true

Table Columns

The columns of the table CorporateSubAccounts 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	char	Active	☐	Whether the sub-account is active or not
companyName	varchar2	Company Name	☐	Name of the sub-account company
createdAt	int64	Creation Date	☐	Timestamp when the sub-account was created
id	int64	ID	☐	id of the sub-account

CorporateUserPermissionAnalyticsByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CorporateUserPermissionAnalyticsByEmail. 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		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionApiKeysByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionApiKeysByEmail. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

Name	Data Type	Required	Default Value	Documentation
email		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionAppsManagementByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionAppsManagementByEmail. 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		☑		Email of the invited user.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionCreateSubOrganizationsByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionCreateSubOrganizationsByEmail. 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		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionGroupsByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionGroupsByEmail. 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		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
id	varchar2	ID	☐	group identifier
name	varchar2	Name	☐	Group name
status	varchar2	Status	☐	Status of the invited user.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

CorporateUserPermissionManageSubOrganizationsByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionManageSubOrganizationsByEmail. 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		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionMyPlansByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CorporateUserPermissionMyPlansByEmail. 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		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionsByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionsByEmail. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

Name	Data Type	Required	Default Value	Documentation
email		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.

CorporateUserPermissionSubOrganizationGroupsByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionSubOrganizationGroupsByEmail. 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		☑		Email of the invited user.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	Permission details

CorporateUserPermissionUserManagementByEmail

Check admin user permissionsThis endpoint will provide the list of admin user permissions

Catalog: Brevo

Schema: MasterAccount

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 CorporateUserPermissionUserManagementByEmail. 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		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
status	varchar2	Status	☐	Status of the invited user.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
TEXT	varchar2	Text	☐	Permission details

SubAccountGroups

Get the list of groupsThis endpoint allows you to list all groups created on your Admin account.

Catalog: Brevo

Schema: MasterAccount

Primary Keys: id

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 SubAccountGroups are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
groupName	varchar2		☐	The name of the group of sub-accounts
id	varchar2	ID	☐	Unique id of the group

3.1.17 Schema: Native

3.1.17.1 Tables

NATIVEPLATFORMSCALARREQUESTS: Sendinblue Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: Brevo

Schema: Native

Alias: `npt`

Label: Native Platform Scalar Requests

Documentation:

The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the Brevo API server. It will contain a new row for every row inserted with a native API request in PAYLOAD_TEXT with the results of unaltered forwarding of the payload to the Brevo API server.

Retrieve: true

Insert: true

Update: false

Delete: false

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	char	BLOB Preferred	☑	Indicator whether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	varchar2(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☑	
DATE_STARTED	datetime	Start Date	☑	
DRY_RUN	char	Run without Actions	☑	
DURATION_MS	int64	Duration (ms)	☑	
ERROR_MESSAGE_CODE	varchar2(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	varchar2(32000)	Error Message Text	☐	
FAIL_ON_ERROR	char	Fail on Error	☑	Whether to raise an exception when processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	char	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	char	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	char	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	char	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	varchar2(30)	HTTP Method	☐	
HTTP_STATUS_CODE	numeric	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	varchar2(4000)	Original System Group	☐	
ORIG_SYSTEM_REFERENCE	varchar2(4000)	Original System Reference	☐	
PAYLOAD_TEXT	varchar2	Payload	☐	
RESULT_BLOB	blob	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	☐	
RESULT_NUMBER	number	Result Number	☐	
RESULT_TEXT	varchar2	Result Text	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
SUCCESSFUL	char	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int64	Transaction ID	☑	Incrementing ID of the transaction.
URL	varchar2(4000)	URL	☐	

3.1.18 Schema: Notes

3.1.18.1 Tables

Crm_CompanyIdsNotes

Get all notesRetrieve a paginated list of CRM notes with optional filtering by entity type, entity IDs, and date range. Results are sorted by creation date in descending order by default, with a default limit of 50 notes per page.

Catalog: Brevo

Schema: Notes

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 Crm_CompanyIdsNotes. 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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
entity		☐		Filter by note entity type (Values: companies, deals, contacts)
entityIds		☐		Filter by note entity IDs

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user w hich created the note
contactIds	varchar2		☐	Contact ids linked to a note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text_2	varchar2(10000)		☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Note updated date/time

Crm_ContactIdsNotes

Get all notesRetrieve a paginated list of CRM notes with optional filtering by entity type, entity IDs, and date range. Results are sorted by creation date in descending order by default, with a default limit of 50 notes per page.

Catalog: Brevo

Schema: Notes

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 Crm_ContactIdsNotes. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
entity		☐		Filter by note entity type (Values: companies, deals, contacts)
entityIds		☐		Filter by note entity IDs

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user which created the note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text_2	varchar2(10000)		☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
TEXT	int64	Text	☐	
updatedAt	datetime		☐	Note updated date/time

Crm_DealIdsNotes

Get all notesRetrieve a paginated list of CRM notes with optional filtering by entity type, entity IDs, and date range. Results are sorted by creation date in descending order by default, with a default limit of 50 notes per page.

Catalog: Brevo

Schema: Notes

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 Crm_DealIdsNotes. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
entity		☐		Filter by note entity type (Values: companies, deals, contacts)
entityIds		☐		Filter by note entity IDs

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user w hich created the note
contactIds	varchar2		☐	Contact ids linked to a note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text_2	varchar2(10000)		☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Note updated date/time

CrmNotes

Get all notesRetrieve a paginated list of CRM notes with optional filtering by entity type, entity IDs, and date range. Results are sorted by creation date in descending order by default, with a default limit of 50 notes per page.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: Notes

Primary Keys: id

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CrmNotes. 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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
entity		☐		Filter by note entity type (Values: companies, deals, contacts)
entityIds		☐		Filter by note entity IDs

Columns of Table Function

The columns of the table function CrmNotes 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
authorId	varchar2		☐	Account details of user which created the note
contactIds	varchar2		☐	Contact ids linked to a note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text	varchar2(10000)	Text	☐	Content of the note. Supports HTML for rich text formatting.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
updatedAt	datetime		☐	Note updated date/time

CrmNotes_CompanyIdsById

Get a noteRetrieve the full details of a single CRM note by its identifier. The response includes the note''s text content, creation date, author, and any associated contacts, companies, or deals.

Catalog: Brevo

Schema: Notes

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Note ID to get

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user which created the note

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
contactIds	varchar2		☐	Contact ids linked to a note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text_2	varchar2(10000)		☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Note updated date/time

CrmNotes_ContactIdsById

Get a noteRetrieve the full details of a single CRM note by its identifier. The response includes the note''s text content, creation date, author, and any associated contacts, companies, or deals.

Catalog: Brevo

Schema: Notes

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Note ID to get

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user w hich created the note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text_2	varchar2(10000)		☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
TEXT	int64	Text	☐	
updatedAt	datetime		☐	Note updated date/time

CrmNotes_DealIdsById

Get a noteRetrieve the full details of a single CRM note by its identifier. The response includes the note''s text content, creation date, author, and any associated contacts, companies, or deals.

Catalog: Brevo

Schema: Notes

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Note ID to get

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user which created the note
contactIds	varchar2		☐	Contact ids linked to a note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text_2	varchar2(10000)		☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Note updated date/time

CrmNotesById

Get a noteRetrieve the full details of a single CRM note by its identifier. The response includes the note''s text content, creation date, author, and any associated contacts, companies, or deals.

Catalog: Brevo

Schema: Notes

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 CrmNotesById. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Note ID to get

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
authorId	varchar2		☐	Account details of user which created the note
contactIds	varchar2		☐	Contact ids linked to a note
createdAt	datetime	Creation Date	☐	Note created date/time
id	varchar2	ID	☐	Unique note Id
text	varchar2(10000)	Text	☐	Content of the note. Supports HTML for rich text formatting. Supported tags include: `<p>` (paragraph), `<b>` / `<strong>` (bold), `<i>` / `<em>` (italic), `<u>` (underline), ` ` (line break), `<a href="...">` (labelled hyperlink). Example labelled link: `<a href="https://example.com">Link text</a>`.
updatedAt	datetime		☐	Note updated date/time

3.1.19 Schema: Payments

3.1.19.1 Tables

PaymentRequestsById

Get payment request detailsRetrieve the details of a specific payment request by its ID. The response includes the reference, status (created, sent, reminderSent, or paid), cart details, notification configuration, contact ID, and the number of reminders sent. Returns a `404` error if no payment request matches the provided ID.

Catalog: Brevo

Schema: Payments

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 PaymentRequestsById. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		Id of the payment Request

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
cart_currency	varchar2		☐	Currency code for the payment amount.
cart_specificAmount	int64		☐	Payment amount, in cents. e.g. if you want to request €12.00, then the amount in cents is 1200.
configuration_customSuccessUrl	varchar2		☐	Absolute URL of the custom success page.
contactId	int64		☐	Brevo ID of the contact requested to pay.
notification_channel	varchar2		☐	Channel used to send the notifications.
notification_text	varchar2		☐	Use this field if you want to give more context to your contact about the payment request.
numberOfRemindersSent	int64		☐	number of reminders sent.
reference	varchar2	Reference	☐	Reference of the payment request, it will appear on the payment page.
status	varchar2	Status	☐	Status of the payment request.

3.1.20 Schema: Process

3.1.20.1 Tables

BackgroundProcessById: Sendinblue Background Process by ID

Get process detailsRetrieves detailed information about a specific background process.**Use this to:**- Get detailed status of a specific process- Monitor process progress and completion- Download results from completed export processes- Check error details for failed processes- Track process execution times**Key information returned:**- Complete process details (ID, name, status)- Download URLs for completed export processes- Import details with CSV report URLs for completed import processes**Important considerations:**- Process ID must exist in your account and not be deleted- Completed export processes in-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

clude download URLs- Completed import processes include CSV report URLs with details about problematic records- Different process types return different result structures

Catalog: Brevo

Schema: Process

Label: Background Process by ID

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
processId		☑		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	varchar2	Export URL	☐	Download URL for completed export processes (returned for SEARCH_EXPORT_USERS, SEARCH_EXPORT_USERS_API, CAMPAIGN_USER_DETAILS, and EXPORT_WEBHOOK process types)
id	int64	ID	☐	Unique identifier of the process
info_import_duplicate_contact_id	varchar2		☐	URL to CSV file containing duplicate contact IDs, or null if none
info_import_duplicate_email_id	varchar2		☐	URL to CSV file containing duplicate email IDs, or null if none
info_import_duplicate_ext_id	varchar2		☐	URL to CSV file containing duplicate external IDs, or null if none

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
info_import_duplicate_land-line_number_id	varchar2		☐	URL to CSV file containing duplicate landline numbers, or null if none
info_import_duplicate_phone_id	varchar2		☐	URL to CSV file containing duplicate phone numbers, or null if none
info_import_duplicate_w hatsap-p_id	varchar2		☐	URL to CSV file containing duplicate WhatsApp numbers, or null if none
info_import_invalid_emails	varchar2		☐	URL to CSV file containing invalid email addresses, or null if none
name	varchar2	Name	☐	Name/type of the process
status	varchar2	Status	☐	Current status of the process

BackgroundProcesses: Sendinblue Background Processes

Get background processesRetrieves a list of background processes from your Brevo account with filtering and pagination.**Use this to:**- Monitor background process activity and status- Track long-running operations and tasks- Find process IDs for detailed status checking- Review process history and performance- Identify failed or stuck processes for troubleshooting**Key information returned:**- Process details (ID, name, status)- Export download URLs for completed export processes- Import details with CSV report URLs for completed import processes- Total count of processes for pagination**Important considerations:**- Background processes handle long-running operations like imports and exports- Process status indicates current state (queued, processing, completed)- Export processes provide download URLs when completed- Import processes provide CSV report URLs with details about problematic records- Use pagination for accounts with many historical processes- Sort options available for creation order (ascending or descending)- Default limit is 10 results per page, maximum is 50

Catalog: Brevo

Schema: Process

Primary Keys: id

Label: Background Processes

This is a read-only table. The Sendinblue API may not support changing the data or the 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	varchar2	Export URL	☐	Dow nload URL for completed export processes (returned for SEARCH_EXPORT_USERS, SEARCH_EXPORT_USERS_API, CAMPAIGN_USER_DETAILS, and EXPORT_WEBHOOK process types)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
id	int64	ID	☐	Unique identifier of the process
name	varchar2	Name	☐	Name/type of the process
status	varchar2	Status	☐	Current status of the process

3.1.21 Schema: Program

3.1.21.1 Tables

LoyaltyPrograms

Get loyalty program listReturns list of loyalty programs

Catalog: Brevo

Schema: Program

Primary Keys: id

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 LoyaltyPrograms. 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
sort_field		☐		Sort documents by field (Values: name, created_at, updated_at)

Columns of Table Function

The columns of the table function LoyaltyPrograms 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
birthdayAttribute	varchar2		☐	Contact attribute name used for birthday tracking.
codeCount	int64		☐	Number of available subscription codes.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Timestamp when the loyalty program was created.
description	varchar2	Description	☐	Loyalty Program description.
documentId	guid		☐	Associated document identifier.
id	guid	ID	☐	Loyalty Program unique identifier.
meta	varchar2		☐	Loyalty Program metadata.
name	varchar2	Name	☐	Loyalty Program name.
pattern	varchar2		☐	Subscription code pattern.
state	varchar2		☐	Current state of the loyalty program.
subscriptionGeneratorId	guid		☐	Loyalty Program subscription generator ID.
subscriptionPoolId	guid		☐	Loyalty Program subscription pool ID.
updatedAt	datetime		☐	Timestamp when the loyalty program was last modified.

LoyaltyProgramsByPid

Get loyalty program InfoReturns loyalty program

Catalog: Brevo

Schema: Program

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 LoyaltyProgramsByPid. 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
pid		☑		Loyalty Program ID. A unique identifier for the loyalty program.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
birthdayAttribute	varchar2		☐	Contact attribute name used for birthday tracking.
codeCount	int64		☐	Number of available subscription codes.
createdAt	datetime	Creation Date	☐	Timestamp when the loyalty program was created.
description	varchar2	Description	☐	Loyalty Program description.
documentId	guid		☐	Associated document identifier.
id	guid	ID	☐	Loyalty Program unique identifier.
meta	varchar2		☐	Loyalty Program metadata.
name	varchar2	Name	☐	Loyalty Program name.
pattern	varchar2		☐	Subscription code pattern.
state	varchar2		☐	Current state of the loyalty program.
subscriptionGeneratorId	guid		☐	Loyalty Program subscription generator ID.
subscriptionPoolId	guid		☐	Loyalty Program subscription pool ID.
updatedAt	datetime		☐	Timestamp when the loyalty program was last modified.

ParameterSubscriptionById

Get Subscription DataGet Information of balances, tiers, rewards and subscription members for a subscription

Catalog: Brevo

Schema: Program

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 ParameterSubscriptionById. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
contactId		☐		The contact ID to filter by.
includeInternal		☐		Include balances tied to internal definitions.
loyaltySubscriptionId		☐		The loyalty subscription ID to filter by.
params		☐		A list of filter parameters for querying the subscription info.
pid		☑		Loyalty Program ID. A unique identifier for the loyalty program.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balance_contactId	int64		☐	Unique identifier of the contact.
balance_loyaltyProgramId	varchar2		☐	Unique identifier of the loyalty program.

ParameterSubscriptionInfo_Members

Get Subscription DataGet Information of balances, tiers, rewards and subscription members for a subscription

Catalog: Brevo

Schema: Program

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 ParameterSubscriptionInfo_Members. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
contactId		☐		The contact ID to filter by.
includeInternal		☐		Include balances tied to internal definitions.
loyaltySubscriptionId		☐		The loyalty subscription ID to filter by.
params		☐		A list of filter parameters for querying the subscription info.
pid		☑		Loyalty Program ID. A unique identifier for the loyalty program.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balance_contactId	int64		☐	Unique identifier of the contact.
balance_loyaltyProgramId	varchar2		☐	Unique identifier of the loyalty program.
createdAt	varchar2	Creation Date	☐	Timestamp when the member was created.
memberContactId	int64		☐	Unique identifier of the member.
updatedAt	varchar2		☐	Timestamp when the member was last updated.

ParameterSubscriptionRewardsByPid

Get Subscription DataGet Information of balances, tiers, rewards and subscription members for a subscription

Catalog: Brevo

Schema: Program

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 ParameterSubscriptionRewardsByPid. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
contactId		☐		The contact ID to filter by.
includeInternal		☐		Include balances tied to internal definitions.
loyaltySubscriptionId		☐		The loyalty subscription ID to filter by.
params		☐		A list of filter parameters for querying the subscription info.
pid		☑		Loyalty Program ID. A unique identifier for the loyalty program.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balance_contactId	int64		☐	Unique identifier of the contact.
balance_loyaltyProgramId	varchar2		☐	Unique identifier of the loyalty program.
code	varchar2		☐	Reward code assigned to the contact.
contactId	int64		☐	Unique identifier of the contact.
createdAt	varchar2	Creation Date	☐	Timestamp when the reward was created.
expirationDate	varchar2		☐	Expiration date of the reward.
id	varchar2	ID	☐	Unique identifier of the reward.
loyaltyProgramId	varchar2		☐	Unique identifier of the loyalty program.
meta	varchar2		☐	Additional metadata related to the reward.
rewardId	varchar2		☐	Unique identifier of the reward definition.
updatedAt	datetime		☐	Timestamp when the reward was last updated.
validFrom	datetime		☐	Date from which the voucher becomes valid.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ParameterSubscriptionTiersByPid

Get Subscription DataGet Information of balances, tiers, rewards and subscription members for a subscription

Catalog: Brevo

Schema: Program

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 ParameterSubscriptionTiersByPid. 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
contactId		☐		The contact ID to filter by.
includeInternal		☐		Include balances tied to internal definitions.
loyaltySubscriptionId		☐		The loyalty subscription ID to filter by.
params		☐		A list of filter parameters for querying the subscription info.
pid		☑		Loyalty Program ID. A unique identifier for the loyalty program.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balance_contactId	int64		☐	Unique identifier of the contact.
balance_loyaltyProgramId	varchar2		☐	Unique identifier of the loyalty program.
contactId	int64		☐	Unique identifier of the contact.
createdAt	varchar2	Creation Date	☐	Timestamp when the tier was assigned.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
groupId	varchar2		☐	Unique identifier of the group associated with the tier.
loyaltyProgramId	varchar2		☐	Unique identifier of the loyalty program.
meta	varchar2		☐	Additional metadata related to the tier.
tierId	varchar2		☐	Unique identifier of the tier.
updatedAt	varchar2		☐	Timestamp when the tier was last updated

3.1.22 Schema: Reward

3.1.22.1 Tables

CodeCountsByPidAndCpid

Get code countGet code count

Catalog: Brevo

Schema: Reward

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 CodeCountsByPidAndCpid. 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
cpid		☑		Code Pool ID
pid		☑		Loyalty Program ID

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
count	int64	Count	☐	Number of codes

LoyaltyOfferProgramsByPid_ContactRewardsVouchers

Get voucher for a contactGet voucher for a contact

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPid_ContactRewardsVouchers. 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
contactId		☑		Contact ID
metadata_key_value		☐		Metadata value for a Key filter
pid		☑		Loyalty Program ID
rewardId		☐		Reward ID
sortField		☐		Sort field (Values: updatedAt, createdAt)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
code	varchar2		☐	Generated code
consumedAt	varchar2		☐	Timestamp when the reward limit was consumed

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
contactId	int64		☐	Contact id associated w ith the current rew ard
count	int64	Count	☐	Count of the rew ards associated w ith the current contact
createdAt	varchar2	Creation Date	☐	Timestamp w hen the rew ard limit w as created
expirationDate	varchar2		☐	Rew ard expiration date
id	varchar2	ID	☐	Unique identifier for the contact
loyaltyProgramId	varchar2		☐	Loyalty Program Id for the con- tact
loyaltySubscriptionId	varchar2		☐	Loyalty Subscription Id for the contact
meta	varchar2		☐	Additional data for the rew ard
rew ardId	varchar2		☐	Unique identifier for the rew ard
unit	varchar2		☐	Unit of the balance selected in the rew ard definition
updatedAt	varchar2		☐	Timestamp w hen the rew ard limit w as updated
validFrom	datetime		☐	Date from w hich the voucher be- comes valid
value	double	Value	☐	Value of the selected rew ard config

LoyaltyOfferProgramsByPid_ItemsOffers

Get Reward Page APIReturns a reward page

Catalog: Brevo

Schema: Reward

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 Loy- altyOfferProgramsByPid_ItemsOffers. 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 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
pid		☑		Loyalty Program ID
state		☐		State of the reward
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Timestamp for when this reward was created
endDate	datetime	End Date	☐	Timestamp for when this reward attribution ended
id	guid	ID	☐	Unique identifier for the reward
loyaltyProgramId	guid		☐	Loyalty Program to which current reward is associated
name	varchar2	Name	☐	Name of the reward
publicImage	varchar2		☐	Public image of the reward
startDate	datetime	Start Date	☐	Timestamp for when this reward attribution started
state	varchar2		☐	State of the reward
totalCount	int64		☐	Count of the associated parameter in current reward
updatedAt	datetime		☐	Timestamp for when this reward attribution was updated

LoyaltyOfferProgramsByPidOffers

Get Reward Page APIReturns a reward page

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPidOffers. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
pid		☑		Loyalty Program ID
state		☐		State of the reward
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
totalCount	int64		☐	Count of the associated parameter in current reward

LoyaltyOfferProgramsByPidRewards_LimitsRid

Get reward informationReturns reward information.

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPidRewards_LimitsRid. 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
pid		☑		Loyalty Program ID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
rid		☑		Reward ID
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributionPerConsumer	int64		☐	Maximum number of times a consumer can be attributed this reward
balanceDefinitionId	guid		☐	Unique identifier for the balance definition
code	varchar2		☐	Unique code for the reward
codeCount	int64		☐	Total number of codes generated
codeGeneratorId	guid		☐	Unique identifier for the code generator
codePoolId	guid		☐	Unique identifier for the code pool
config	varchar2		☐	Configuration settings for the reward
createdAt_2	datetime		☐	Timestamp when the reward was created
createdAt	datetime	Creation Date	☐	Timestamp when the reward limit was created
disabledAt	datetime		☐	Disabled date of the reward
durationUnit	varchar2		☐	Unit of time for the reward limit's availability (e.g., day/week/month/year).
durationValue	int64		☐	Number of days/weeks/month/year for reward limit
endDate	datetime	End Date	☐	End date of the reward validity
expirationDate	datetime		☐	Expiration date of the reward
expirationModifier	varchar2		☐	Select startOfPeriod to configure rewards expiry on start of day/week/month/year. Select endOfPeriod to configure reward expiry on end of day/week/month/year, else select noModification
expirationUnit	varchar2		☐	Unit of time for the rewards's availability (e.g., day/week/month/year).
expirationValue	int64		☐	Number of days/weeks/month/year for reward expiry
generator_createdAt	varchar2		☐	Timestamp when the reward was created

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
generator_description	varchar2		☐	Public facing description of the reward
generator_id	guid		☐	Unique identifier for the reward
generator_name	varchar2		☐	Name of the reward
generator_pattern	varchar2		☐	Generated pattern
generator_updatedAt	datetime		☐	Timestamp when the reward was created
id	guid	ID	☐	Unique identifier for the reward
limitValue	int64		☐	Value of the limit
loyaltyProgramId	guid		☐	Id of the loyalty program to which the current reward belongs to
meta	varchar2		☐	Additional data for reward definition
name	varchar2	Name	☐	Name of the reward
publicDescription	varchar2		☐	Public description for the reward
publicImage	varchar2		☐	Public Image for the reward
publicName	varchar2		☐	Public name for the reward
redeemPerConsumer	int64		☐	Defines the redeem limit for the consumer
rewardConfigs_attribution	varchar2		☐	Attribution config of the reward
rewardConfigs_code	varchar2		☐	Code config of the reward
rewardConfigs_value	varchar2		☐	Value config of the reward
rewardLimitId	varchar2		☐	Unique identifier for the reward limit
rule_condition_op	varchar2		☐	Selected operator for the rule
rule_createdAt	varchar2		☐	Timestamp when the rule was created
rule_description	varchar2		☐	Description of the rule
rule_event_name	varchar2		☐	Name of the event
rule_event_source	varchar2		☐	Source of the event
rule_isInternal	char		☐	Metric to identify if it's an internal rule or not
rule_loyaltyProgramId	varchar2		☐	Loyalty Program id to which current rule is associated
rule_loyaltyVersionId	int64		☐	Loyalty Version id to which current rule is associated
rule_meta	varchar2		☐	Additional data to define the rule
rule_name	varchar2		☐	Rule name
rule_ruleId	varchar2		☐	Unique identifier for the rule
rule_ruleType	varchar2		☐	Type of the rule
rule_updatedAt	varchar2		☐	Timestamp when the rule was updated
slidingSchedule	char		☐	Select true to calculate all redeems/attributions from the pre-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				vious value of selected durationUnit to the current time
startDate	datetime	Start Date	☐	Start date of attribution of the reward
subtractBalanceDefinitionId	varchar2		☐	Id of the selected balance while redeeming / attributing a reward
subtractBalanceStrategy	varchar2		☐	Strategy of the Balance while redeeming / attributing a reward
subtractBalanceValue	int64		☐	Amount of balance to be selected while redeeming / attributing a reward
subtractTotalBalance	char		☐	Value to indicate to subtract full balance or not
totalAttribution	int64		☐	Defines the limit to which a consumer can attribute a reward
totalRedeem	int64		☐	Defines the limit to which a consumer can redeem a reward
triggerId	varchar2		☐	Id of the Rule to be updated for that reward
type	varchar2	Type	☐	Type of reward
unit	varchar2		☐	Selected unit of the balance
updatedAt_2	varchar2		☐	Timestamp for when this reward was last updated.
updatedAt	datetime		☐	Timestamp when the reward limit was created
value	decimal	Value	☐	Value of metric in selected config for reward definition
valueType	varchar2		☐	Type of config selected for reward definition

LoyaltyOfferProgramsByPidRewards_ProductsRid

Get reward informationReturns reward information.

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPidRewards_ProductsRid. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
pid		☑		Loyalty Program ID
rid		☑		Reward ID
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributionPerConsumer	int64		☐	Maximum number of times a consumer can be attributed this reward
balanceDefinitionId	guid		☐	Unique identifier for the balance definition
code	varchar2		☐	Unique code for the reward
codeCount	int64		☐	Total number of codes generated
codeGeneratorId	guid		☐	Unique identifier for the code generator
codePoolId	guid		☐	Unique identifier for the code pool
config	varchar2		☐	Configuration settings for the reward
createdAt_2	datetime		☐	Timestamp when the reward was created
createdAt	varchar2	Creation Date	☐	Timestamp when the product was created
disabledAt	datetime		☐	Disabled date of the reward
endDate	datetime	End Date	☐	End date of the reward validity
expirationDate	datetime		☐	Expiration date of the reward
expirationModifier	varchar2		☐	Select startOfPeriod to configure rewards expiry on start of day/week/month/year. Select endOfPeriod to configure reward expiry on end of day/week/month/year, else select noModification
expirationUnit	varchar2		☐	Unit of time for the rewards's availability (e.g., day/week/month/year).
expirationValue	int64		☐	Number of days/weeks/month/year for re-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				w ard expiry
generator_createdAt	varchar2		☐	Timestamp w hen the rew ard w as created
generator_description	varchar2		☐	Public facing description of the rew ard
generator_id	guid		☐	Unique identifier for the rew ard
generator_name	varchar2		☐	Name of the rew ard
generator_pattern	varchar2		☐	Generated pattern
generator_updatedAt	datetime		☐	Timestamp w hen the rew ard w as created
id	guid	ID	☐	Unique identifier for the rew ard
imageRef	varchar2		☐	URL of the product image
loyaltyProgramId	guid		☐	Id of the loyalty program to w hich the current rew ard be-longs to
meta	varchar2		☐	Additional data for rew ard defini-tion
name	varchar2	Name	☐	Name of the rew ard
productId	varchar2		☐	Unique identifier for the product
publicDescription	varchar2		☐	Public description for the rew ard
publicImage	varchar2		☐	Public Image for the rew ard
publicName	varchar2		☐	Public name for the rew ard
redeemPerConsumer	int64		☐	Defines the redeem limit for the consumer
rew ardConfigs_attribution	varchar2		☐	Attribution config of the rew ard
rew ardConfigs_code	varchar2		☐	Code config of the rew ard
rew ardConfigs_value	varchar2		☐	Value config of the rew ard
rule_condition_op	varchar2		☐	Selected operator for the rule
rule_createdAt	varchar2		☐	Timestamp w hen the rule w as created
rule_description	varchar2		☐	Description of the rule
rule_event_name	varchar2		☐	Name of the event
rule_event_source	varchar2		☐	Source of the event
rule_isInternal	char		☐	Metric to identify if it's an internal rule or not
rule_loyaltyProgramId	varchar2		☐	Loyalty Program id to w hich cur-rent rule is associated
rule_loyaltyVersionId	int64		☐	Loyalty Version id to w hich cur-rent rule is associated
rule_meta	varchar2		☐	Additional data to define the rule
rule_name	varchar2		☐	Rule name
rule_ruleId	varchar2		☐	Unique identifier for the rule
rule_ruleType	varchar2		☐	Type of the rule
rule_updatedAt	varchar2		☐	Timestamp w hen the rule w as updated

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
startDate	datetime	Start Date	☐	Start date of attribution of the reward
subtractBalanceDefinitionId	varchar2		☐	Id of the selected balance while redeeming / attributing a reward
subtractBalanceStrategy	varchar2		☐	Strategy of the Balance while redeeming / attributing a reward
subtractBalanceValue	int64		☐	Amount of balance to be selected while redeeming / attributing a reward
subtractTotalBalance	char		☐	Value to indicate to subtract full balance or not
totalAttribution	int64		☐	Defines the limit to which a consumer can attribute a reward
totalRedeem	int64		☐	Defines the limit to which a consumer can redeem a reward
triggerId	varchar2		☐	Id of the Rule to be updated for that reward
unit	varchar2		☐	Selected unit of the balance
updatedAt	varchar2		☐	Timestamp for when this reward was last updated.
value_2	decimal		☐	Value of metric in selected config for reward definition
value	varchar2	Value	☐	string
valueType	varchar2		☐	Type of config selected for reward definition

LoyaltyOfferProgramsByPidRewards_RedeemRulesRid

Get reward informationReturns reward information.

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPidRewards_RedeemRulesRid. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
pid		☑		Loyalty Program ID
rid		☑		Reward ID
version		☐		Version (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributionPerConsumer	int64		☐	Maximum number of times a consumer can be attributed this reward
balanceDefinitionId	guid		☐	Unique identifier for the balance definition
code	varchar2		☐	Unique code for the reward
codeCount	int64		☐	Total number of codes generated
codeGeneratorId	guid		☐	Unique identifier for the code generator
codePoolId	guid		☐	Unique identifier for the code pool
config	varchar2		☐	Configuration settings for the reward
createdAt	datetime	Creation Date	☐	Timestamp when the reward was created
disabledAt	datetime		☐	Disabled date of the reward
endDate	datetime	End Date	☐	End date of the reward validity
expirationDate	datetime		☐	Expiration date of the reward
expirationModifier	varchar2		☐	Select startOfPeriod to configure rewards expiry on start of day/week/month/year. Select endOfPeriod to configure reward expiry on end of day/week/month/year, else select noModification
expirationUnit	varchar2		☐	Unit of time for the rewards's availability (e.g., day/week/month/year).
expirationValue	int64		☐	Number of days/weeks/month/year for reward expiry
generator_createdAt	varchar2		☐	Timestamp when the reward was created
generator_description	varchar2		☐	Public facing description of the reward

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
generator_id	guid		☐	Unique identifier for the reward
generator_name	varchar2		☐	Name of the reward
generator_pattern	varchar2		☐	Generated pattern
generator_updatedAt	datetime		☐	Timestamp when the reward was created
id	guid	ID	☐	Unique identifier for the reward
loyaltyProgramId	guid		☐	Id of the loyalty program to which the current reward belongs to
meta	varchar2		☐	Additional data for reward definition
name	varchar2	Name	☐	Name of the reward
publicDescription	varchar2		☐	Public description for the reward
publicImage	varchar2		☐	Public Image for the reward
publicName	varchar2		☐	Public name for the reward
redeemPerConsumer	int64		☐	Defines the redeem limit for the consumer
rewardConfigs_attribution	varchar2		☐	Attribution config of the reward
rewardConfigs_code	varchar2		☐	Code config of the reward
rewardConfigs_value	varchar2		☐	Value config of the reward
rule_condition_op	varchar2		☐	Selected operator for the rule
rule_createdAt	varchar2		☐	Timestamp when the rule was created
rule_description	varchar2		☐	Description of the rule
rule_event_name	varchar2		☐	Name of the event
rule_event_source	varchar2		☐	Source of the event
rule_isInternal	char		☐	Metric to identify if it's an internal rule or not
rule_loyaltyProgramId	varchar2		☐	Loyalty Program id to which current rule is associated
rule_loyaltyVersionId	int64		☐	Loyalty Version id to which current rule is associated
rule_meta	varchar2		☐	Additional data to define the rule
rule_name	varchar2		☐	Rule name
rule_ruleId	varchar2		☐	Unique identifier for the rule
rule_ruleType	varchar2		☐	Type of the rule
rule_updatedAt	varchar2		☐	Timestamp when the rule was updated
startDate	datetime	Start Date	☐	Start date of attribution of the reward
subtractBalanceDefinitionId	varchar2		☐	Id of the selected balance while redeeming / attributing a reward
subtractBalanceStrategy	varchar2		☐	Strategy of the Balance while redeeming / attributing a reward
subtractBalanceValue	int64		☐	Amount of balance to be selected while redeeming / attributing

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				a reward
subtractTotalBalance	char		☐	Value to indicate to subtract full balance or not
TEXT	varchar2	Text	☐	
totalAttribution	int64		☐	Defines the limit to which a consumer can attribute a reward
totalRedeem	int64		☐	Defines the limit to which a consumer can redeem a reward
triggerId	varchar2		☐	Id of the Rule to be updated for that reward
unit	varchar2		☐	Selected unit of the balance
updatedAt	varchar2		☐	Timestamp for when this reward was last updated.
value	decimal	Value	☐	Value of metric in selected config for reward definition
valueType	varchar2		☐	Type of config selected for reward definition

LoyaltyOfferProgramsByPidRewardsRid

Get reward informationReturns reward information.

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPidRewardsRid. 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
pid		☑		Loyalty Program ID
rid		☑		Reward ID
version		☐		Version (Values: active, draft)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
attributionPerConsumer	int64		☐	Maximum number of times a consumer can be attributed this reward
balanceDefinitionId	guid		☐	Unique identifier for the balance definition
code	varchar2		☐	Unique code for the reward
codeCount	int64		☐	Total number of codes generated
codeGeneratorId	guid		☐	Unique identifier for the code generator
codePoolId	guid		☐	Unique identifier for the code pool
config	varchar2		☐	Configuration settings for the reward
createdAt	datetime	Creation Date	☐	Timestamp when the reward was created
disabledAt	datetime		☐	Disabled date of the reward
endDate	datetime	End Date	☐	End date of the reward validity
expirationDate	datetime		☐	Expiration date of the reward
expirationModifier	varchar2		☐	Select startOfPeriod to configure rewards expiry on start of day/week/month/year. Select endOfPeriod to configure reward expiry on end of day/week/month/year, else select noModification
expirationUnit	varchar2		☐	Unit of time for the rewards's availability (e.g., day/week/month/year).
expirationValue	int64		☐	Number of days/weeks/month/year for reward expiry
generator_createdAt	varchar2		☐	Timestamp when the reward was created
generator_description	varchar2		☐	Public facing description of the reward
generator_id	guid		☐	Unique identifier for the reward
generator_name	varchar2		☐	Name of the reward
generator_pattern	varchar2		☐	Generated pattern
generator_updatedAt	datetime		☐	Timestamp when the reward was created
id	guid	ID	☐	Unique identifier for the reward
loyaltyProgramId	guid		☐	Id of the loyalty program to which the current reward belongs to

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
meta	varchar2		☐	Additional data for reward definition
name	varchar2	Name	☐	Name of the reward
publicDescription	varchar2		☐	Public description for the reward
publicImage	varchar2		☐	Public Image for the reward
publicName	varchar2		☐	Public name for the reward
redeemPerConsumer	int64		☐	Defines the redeem limit for the consumer
rewardConfigs_attribution	varchar2		☐	Attribution config of the reward
rewardConfigs_code	varchar2		☐	Code config of the reward
rewardConfigs_value	varchar2		☐	Value config of the reward
rule_condition_op	varchar2		☐	Selected operator for the rule
rule_createdAt	varchar2		☐	Timestamp when the rule was created
rule_description	varchar2		☐	Description of the rule
rule_event_name	varchar2		☐	Name of the event
rule_event_source	varchar2		☐	Source of the event
rule_isInternal	char		☐	Metric to identify if it's an internal rule or not
rule_loyaltyProgramId	varchar2		☐	Loyalty Program id to which current rule is associated
rule_loyaltyVersionId	int64		☐	Loyalty Version id to which current rule is associated
rule_meta	varchar2		☐	Additional data to define the rule
rule_name	varchar2		☐	Rule name
rule_ruleId	varchar2		☐	Unique identifier for the rule
rule_ruleType	varchar2		☐	Type of the rule
rule_updatedAt	varchar2		☐	Timestamp when the rule was updated
startDate	datetime	Start Date	☐	Start date of attribution of the reward
subtractBalanceDefinitionId	varchar2		☐	Id of the selected balance while redeeming / attributing a reward
subtractBalanceStrategy	varchar2		☐	Strategy of the Balance while redeeming / attributing a reward
subtractBalanceValue	int64		☐	Amount of balance to be selected while redeeming / attributing a reward
subtractTotalBalance	char		☐	Value to indicate to subtract full balance or not
totalAttribution	int64		☐	Defines the limit to which a consumer can attribute a reward
totalRedeem	int64		☐	Defines the limit to which a consumer can redeem a reward
triggerId	varchar2		☐	Id of the Rule to be updated for that reward
unit	varchar2		☐	Selected unit of the balance

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
updatedAt	varchar2		☐	Timestamp for when this reward was last updated.
value	decimal	Value	☐	Value of metric in selected config for reward definition
valueType	varchar2		☐	Type of config selected for reward definition

LoyaltyOfferProgramsByPidVouchers

Get voucher for a contactGet voucher for a contact

Catalog: Brevo

Schema: Reward

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 LoyaltyOfferProgramsByPidVouchers. 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
contactId		☑		Contact ID
metadata_key_value		☐		Metadata value for a Key filter
pid		☑		Loyalty Program ID
rewardId		☐		Reward ID
sortField		☐		Sort field (Values: updatedAt, createdAt)

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
contactId	int64		☐	Contact id associated with the current reward
count	int64	Count	☐	Count of the rewards associated with the current contact
loyaltyProgramId	varchar2		☐	Loyalty Program Id for the contact
loyaltySubscriptionId	varchar2		☐	Loyalty Subscription Id for the contact

3.1.23 Schema: Senders

3.1.23.1 Tables

IpAddresses: Sendinblue IP Addresses

Get all the dedicated IPs for your accountRetrieves all dedicated IPs associated with your Brevo account.Use this to:- List all your dedicated IPs- Check the status of your dedicated IPs (active/inactive)- Find IP addresses and associated domains for configuration purposes- Monitor your IP reputation and deliverability- Verify available IPs for sender configurationKey information returned:- IP ID and address- Associated domain- Active status- IP configuration details

Catalog: Brevo

Schema: Senders

Primary Keys: id

Label: IP Addresses

This is a read-only table. The Sendinblue API may not support changing the data or the 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	char	Active	☐	Status of the IP (true=active, false=inactive)
domain	varchar2	Domain	☐	Domain associated to the IP
id	int64	ID	☐	ID of the dedicated IP
ip	varchar2	IP	☐	Dedicated IP address

IpAddressesBySenderId: Sendinblue IP Addresses by Send ID

Get all the dedicated IPs for a specific senderRetrieves the dedicated IPs associated with a specific sender.Use this to:- Check IP configuration for a sender- Verify dedicated IP associations- Get IP details for troubleshooting- Monitor sender IP configurationKey information returned:- List of associated dedicated IPs- IP addresses and domain configurations- IP status and settings

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: Senders

Label: IP Addresses by Send ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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		☑		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	varchar2	Domain	☐	Domain associated to the IP
id	int64	ID	☐	ID of the dedicated IP
ip	varchar2	IP	☐	Dedicated IP address
weight	int64	Weight	☐	Weight of the IP for this sender

SenderIpAddresses: Sendinblue Sender IP Addresses

Get email sendersRetrieves a list of all email senders from your Brevo account with optional filtering.**Use this to:**- Get all available senders for email campaign setup- Find sender details including ID, name, and email address- Filter senders by IP address for dedicated IP users- Filter senders by domain for domain-specific configurations- Monitor sender configuration and status**Key information returned:**- Sender details (ID, name, email address)- Sender status and verification information- Associated IP addresses and domains (for dedicated IP accounts)- Sender configuration settings**Important considerations:**- Standard accounts show empty IP arrays, dedicated IP accounts show IP assignments- Filtering by IP only available for accounts with dedicated IPs- Domain filtering helps organize senders by

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

business units or brands- Sender status indicates if sender is active and ready for campaign use- Email verification required before sender can be used in campaigns

Catalog: Brevo

Schema: Senders

Label: Sender IP Addresses

This is a read-only table function. The Sendinblue API may not support changing the data or the 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		☐		Filter your senders for a specific domain
ip		☐		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	char	Active	☐	Status of sender (true=activated, false=deactivated)
domain	varchar2	Domain	☐	Domain of the IP
email	varchar2	Email Address	☐	From Email associated to the sender
id	int64	ID	☐	Id of the sender
ip	varchar2	IP	☐	Dedicated IP available in your account
name	varchar2	Name	☐	From Name associated to the sender

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
w eight	int64	Weight	□	Weight of the IP for this sender

Senders: Sendinblue Senders

Get email sendersRetrieves a list of all email senders from your Brevo account with optional filtering.**Use this to:**- Get all available senders for email campaign setup- Find sender details including ID, name, and email address- Filter senders by IP address for dedicated IP users- Filter senders by domain for domain-specific configurations- Monitor sender configuration and status**Key information returned:**- Sender details (ID, name, email address)- Sender status and verification information- Associated IP addresses and domains (for dedicated IP accounts)- Sender configuration settings**Important considerations:**- Standard accounts show empty IP arrays, dedicated IP accounts show IP assignments- Filtering by IP only available for accounts with dedicated IPs- Domain filtering helps organize senders by business units or brands- Sender status indicates if sender is active and ready for campaign use- Email verification required before sender can be used in campaigns

Catalog: Brevo

Schema: Senders

Primary Keys: id

Label: Senders

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

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
domain		□		Filter your senders for a specific domain
ip		□		Filter your senders for a specific ip. **Available for dedicated IP usage only**

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	char	Active	☐	Status of sender (true=activated, false=deactivated)
email	varchar2	Email Address	☐	From Email associated to the sender
id	int64	ID	☐	Id of the sender
name	varchar2	Name	☐	From Name associated to the sender

3.1.24 Schema: SMSCampaigns

3.1.24.1 Tables

SmsCampaign_RecipientsExclusionLists

Get an SMS campaignRetrieve detailed information about a specific SMS campaign by its ID, including campaign content, sender, recipients with list names, statistics (delivered, sent, bounces, unsubscriptions, answered), and tags. Unlike the list endpoint, recipients are returned as objects with id and name fields rather than plain IDs.

Catalog: Brevo

Schema: SMSCampaigns

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 SmsCampaign_RecipientsExclusionLists. 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		☑		id of the SMS campaign

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
content	varchar2	Content	☐	Content of the SMS Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id_2	int64		☐	ID of the exclusion list
id	int64	ID	☐	ID of the SMS Campaign
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name_2	varchar2		☐	Name of the exclusion list
name	varchar2	Name	☐	Name of the SMS Campaign
organisationPrefix	varchar2		☐	A recognizable prefix added as the Brand Name before the message content. Empty string if not set.
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format. Empty string if not scheduled.
sender	varchar2	Sender	☐	Sender of the SMS Campaign
sentDate	varchar2	Sent Date	☐	UTC date-time on which the SMS campaign was sent (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if the campaign status is 'sent'.
statistics	varchar2	Statistics	☐	
status	varchar2	Status	☐	Status of the SMS Campaign
unsubscribeInstruction	varchar2		☐	Instructions to unsubscribe from future communications. Empty string if not set.

SmsCampaign_RecipientsLists

Get an SMS campaignRetrieve detailed information about a specific SMS campaign by its ID, including campaign content, sender, recipients with list names, statistics (delivered, sent, bounces, unsubscriptions, answered), and tags. Unlike the list endpoint, recipients are returned as objects with id and name fields rather than plain IDs.

Catalog: Brevo

Schema: SMSCampaigns

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the table function SmsCampaign_RecipientsLists. 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		☑		id of the SMS campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	varchar2	Content	☐	Content of the SMS Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id_2	int64		☐	ID of the list
id	int64	ID	☐	ID of the SMS Campaign
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name_2	varchar2		☐	Name of the list
name	varchar2	Name	☐	Name of the SMS Campaign
organisationPrefix	varchar2		☐	A recognizable prefix added as the Brand Name before the message content. Empty string if not set.
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format. Empty string if not scheduled.
sender	varchar2	Sender	☐	Sender of the SMS Campaign
sentDate	varchar2	Sent Date	☐	UTC date-time on which the SMS campaign was sent (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if the campaign status is 'sent'.
statistics	varchar2	Statistics	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
status	varchar2	Status	☐	Status of the SMS Campaign
unsubscribeInstruction	varchar2		☐	Instructions to unsubscribe from future communications. Empty string if not set.

SmsCampaign_Tags

Get an SMS campaignRetrieve detailed information about a specific SMS campaign by its ID, including campaign content, sender, recipients with list names, statistics (delivered, sent, bounces, unsubscriptions, answered), and tags. Unlike the list endpoint, recipients are returned as objects with id and name fields rather than plain IDs.

Catalog: Brevo

Schema: SMSCampaigns

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 SmsCampaign_Tags. 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		☑		id of the SMS campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
content	varchar2	Content	☐	Content of the SMS Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	☐	ID of the SMS Campaign

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the SMS Campaign
organisationPrefix	varchar2		☐	A recognizable prefix added as the Brand Name before the message content. Empty string if not set.
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format. Empty string if not scheduled.
sender	varchar2	Sender	☐	Sender of the SMS Campaign
sentDate	varchar2	Sent Date	☐	UTC date-time on which the SMS campaign was sent (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if the campaign status is 'sent'.
statistics	varchar2	Statistics	☐	
status	varchar2	Status	☐	Status of the SMS Campaign
unsubscribeInstruction	varchar2		☐	Instructions to unsubscribe from future communications. Empty string if not set.
value	varchar2	Value	☐	Tag name

SmsCampaignById: Sendinblue SMS Campaign by ID

Get an SMS campaignRetrieve detailed information about a specific SMS campaign by its ID, including campaign content, sender, recipients with list names, statistics (delivered, sent, bounces, unsubscriptions, answered), and tags. Unlike the list endpoint, recipients are returned as objects with id and name fields rather than plain IDs.

Catalog: Brevo

Schema: SMSCampaigns

Label: SMS Campaign by ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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 para-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		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	varchar2	Content	☐	Content of the SMS Campaign
createdAt	varchar2	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	varchar2	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the SMS Campaign
organisationPrefix	varchar2		☐	A recognizable prefix added as the Brand Name before the message content. Empty string if not set.
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format. Empty string if not scheduled.
sender	varchar2	Sender	☐	Sender of the SMS Campaign
sentDate	varchar2	Sent Date	☐	UTC date-time on which the SMS campaign was sent (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if the campaign status is 'sent'.
statistics	varchar2	Statistics	☐	
status	varchar2	Status	☐	Status of the SMS Campaign
unsubscribeInstruction	varchar2		☐	Instructions to unsubscribe from future communications. Empty string if not set.

SmsCampaigns: Sendinblue SMS Campaigns

Returns the information for all your created SMS campaignsRetrieve a paginated list of all your SMS campaigns with their statistics and recipient information. Results can be filtered by status and date range, with a default limit of 500 and maximum of 1000 per page. The sort

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

order defaults to descending by creation date; date filters are only available when status is not passed or is set to sent.

Catalog: Brevo

Schema: SMSCampaigns

Primary Keys: id

Label: SMS Campaigns

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

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function 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		☐		**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 `status` is not passed or is set to `sent`. `endDate` must not be in the future.
startDate		☐		**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 `status` is not passed or is set to `sent`. `startDate` must not be in the future.
status		☐		Status of campaign. (Values: suspended, archive, sent, queued, draft, inProcess)

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	Content	☐	Content of the SMS Campaign
createdAt	varchar2	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	varchar2	Modified Date	☐	UTC date-time of last modification of the SMS campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the SMS Campaign
organisationPrefix	varchar2		☐	A recognizable prefix added as the Brand Name before the message content. Empty string if not set.
recipients	varchar2	Recipients	☐	
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on which SMS campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format. Empty string if not scheduled.
sender	varchar2	Sender	☐	Sender of the SMS Campaign
sentDate	varchar2	Sent Date	☐	UTC date-time on which the SMS campaign was sent (YYYY-MM-DDTHH:mm:ss.SSSZ). Only available if the campaign status is 'sent'.
statistics	varchar2	Statistics	☐	
status	varchar2	Status	☐	Status of the SMS Campaign
unsubscribeInstruction	varchar2		☐	Instructions to unsubscribe from future communications. Empty string if not set.

3.1.25 Schema: SMSTemplates

3.1.25.1 Tables

SMSTemplates

Return all your created SMS templates

Catalog: Brevo

Schema: SMSTemplates

Primary Keys: id

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
compliance_compliance_toggle	char		☐	US compliance is enabled.
compliance_organisation_prefix	varchar2		☐	A recognizable prefix will ensure your audience knows who you are. Recommended by the U.S. carriers.
compliance_stop_keyword	varchar2		☐	Instructions to unsubscribe from future communications. Must include STOP keyword.
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
encryption	char		☐	Enable Unicode encryption.
id	int64	ID	☐	ID of the template
media_file_size	varchar2		☐	file size in bytes.
media_file	varchar2		☐	file name of the media.
media_url	varchar2		☐	URL of the image.
message	varchar2		☐	message content for the sms
name	varchar2	Name	☐	Name of the template
short_url_enabled	char		☐	URL shortening is enabled
unicode_sms	char		☐	Support unicode characters in the sms.
updatedAt	varchar2		☐	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)

3.1.26 Schema: Tasks

3.1.26.1 Tables

Crm_ItemsCompaniesIdsTasks

Get all tasksRetrieve a paginated list of CRM tasks with optional filtering by task type, status, date range, assignee, and linked entities (contacts, deals, companies). Results are sorted by creation date in descending order by default, with a default limit of 50 tasks per page.

Catalog: Brevo

Schema: Tasks

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 Crm_ItemsCompaniesIdsTasks. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
filter[assignTo]		☐		Filter by the "assignTo" ID. You can utilize account emails for the "assignTo" attribute.
filter[companies]		☐		Filter by companies ids
filter[contacts]		☐		Filter by contact ids
filter[date]		☐		Filter by date (Values: overdue, today, tomorrow , w eek, range)
filter[deals]		☐		Filter by deals ids
filter[status]		☐		Filter by task status (Values: done, undone)
filter[type]		☐		Filter by task type (ID)
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
contactsIds	varchar2		☐	Contact ids for contacts linked to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
TEXT	varchar2	Text	☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
updatedAt	datetime		☐	Task last update date/time

Crm_ItemsContactsIdsTasks

Get all tasksRetrieve a paginated list of CRM tasks with optional filtering by task type, status, date range, assignee, and linked entities (contacts, deals, companies). Results are sorted by creation date in descending order by default, with a default limit of 50 tasks per page.

Catalog: Brevo

Schema: Tasks

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 Crm_ItemsContactsIdsTasks. 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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
filter[assignTo]		☐		Filter by the "assignTo" ID. You can utilize account emails for the "assignTo" attribute.
filter[companies]		☐		Filter by companies ids
filter[contacts]		☐		Filter by contact ids
filter[date]		☐		Filter by date (Values: overdue, today, tomorrow , w eek, range)
filter[deals]		☐		Filter by deals ids
filter[status]		☐		Filter by task status (Values: done, undone)
filter[type]		☐		Filter by task type (ID)
sortBy		☐		The field used to sort field names.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
TEXT	int64	Text	☐	
updatedAt	datetime		☐	Task last update date/time

Crm_ItemsDealsIdsTasks

Get all tasksRetrieve a paginated list of CRM tasks with optional filtering by task type, status, date range, assignee, and linked entities (contacts, deals, companies). Results are sorted by creation date in descending order by default, with a default limit of 50 tasks per page.

Catalog: Brevo

Schema: Tasks

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 Crm_ItemsDealsIdsTasks. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
filter[assignTo]		☐		Filter by the "assignTo" ID. You can utilize account emails for the "assignTo" attribute.
filter[companies]		☐		Filter by companies ids
filter[contacts]		☐		Filter by contact ids
filter[date]		☐		Filter by date (Values: overdue, today, tomorrow , w eek, range)
filter[deals]		☐		Filter by deals ids
filter[status]		☐		Filter by task status (Values: done, undone)
filter[type]		☐		Filter by task type (ID)
sortBy		☐		The field used to sort field names.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
contactsIds	varchar2		☐	Contact ids for contacts linked to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Task last update date/time

CrmTasks

Get all tasksRetrieve a paginated list of CRM tasks with optional filtering by task type, status, date range, assignee, and linked entities (contacts, deals, companies). Results are sorted by creation date in descending order by default, with a default limit of 50 tasks per page.

Catalog: Brevo

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Tasks

Primary Keys: id

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function CrmTasks. 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
dateFrom		☐		dateFrom to date range filter type (timestamp in milliseconds)
dateTo		☐		dateTo to date range filter type (timestamp in milliseconds)
filter[assignTo]		☐		Filter by the "assignTo" ID. You can utilize account emails for the "assignTo" attribute.
filter[companies]		☐		Filter by companies ids
filter[contacts]		☐		Filter by contact ids
filter[date]		☐		Filter by date (Values: overdue, today, tomorrow , w eek, range)
filter[deals]		☐		Filter by deals ids
filter[status]		☐		Filter by task status (Values: done, undone)
filter[type]		☐		Filter by task type (ID)
sortBy		☐		The field used to sort field names.

Columns of Table Function

The columns of the table function CrmTasks 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
contactsIds	varchar2		☐	Contact ids for contacts linked to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
updatedAt	datetime		☐	Task last update date/time

CrmTasks_CompaniesIdsById

Get a taskRetrieve the full details of a single CRM task by its identifier. The response includes the task''s name, type, status, due date, duration, notes, assignee, reminder settings, and linked contacts, companies, or deals.

Catalog: Brevo

Schema: Tasks

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

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
contactsIds	varchar2		☐	Contact ids for contacts linked to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Task last update date/time

CrmTasks_ContactsIdsById

Get a taskRetrieve the full details of a single CRM task by its identifier. The response includes the task''s name, type, status, due date, duration, notes, assignee, reminder settings, and linked contacts, companies, or deals.

Catalog: Brevo

Schema: Tasks

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 CrmTasks_ContactsIdsById. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
id		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
TEXT	int64	Text	☐	
updatedAt	datetime		☐	Task last update date/time

CrmTasks_DealsIdsById

Get a taskRetrieve the full details of a single CRM task by its identifier. The response includes the task''s name, type, status, due date, duration, notes, assignee, reminder settings, and linked contacts, companies, or deals.

Catalog: Brevo

Schema: Tasks

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 CrmTasks_DealsIdsById. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
id		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
contactsIds	varchar2		☐	Contact ids for contacts linked to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
TEXT	varchar2	Text	☐	
updatedAt	datetime		☐	Task last update date/time

CrmTasksById

Get a taskRetrieve the full details of a single CRM task by its identifier. The response includes the task''s name, type, status, due date, duration, notes, assignee, reminder settings, and linked contacts, companies, or deals.

Catalog: Brevo

Schema: Tasks

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 CrmTasksById. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

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

Name	Data Type	Required	Default Value	Documentation
id		☑		

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
assignToId	varchar2		☐	Account id of the user assigned to this task
contactsIds	varchar2		☐	Contact ids for contacts linked to this task
createdAt	datetime	Creation Date	☐	Task creation date/time
date	datetime	Date	☐	Task due date and time
done	char		☐	Whether the task is marked as done
id	varchar2	ID	☐	Unique task id
name	varchar2	Name	☐	Name of task
notes	varchar2		☐	Notes added to a task
taskTypeId	varchar2		☐	Id for type of task e.g Call / Email / Meeting etc.
updatedAt	datetime		☐	Task last update date/time

CrmTasktypes

Get all task typesRetrieve the list of all available task types, such as Email, Call, Meeting, Todo, Lunch, Deadline, and LinkedIn. If no task types exist yet, the default set is automatically created and returned. Use the task type ID when creating or updating tasks.

Catalog: Brevo

Schema: Tasks

Primary Keys: id

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
id	varchar2	ID	☐	Id of task type
title	varchar2		☐	Title of task type

3.1.27 Schema: Tier

3.1.27.1 Tables

ListOfTierGroups

List tier groupsReturns the list of tier groups defined within the loyalty program.

Catalog: Brevo

Schema: Tier

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 ListOfTierGroups. 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
pid		☑		Loyalty Program ID
version		☐		Select 'active' to retrieve list of all tier groups which are live for clients. Select draft to retrieve list of all non deleted tier groups. (Values: active, draft)

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Timestamp when the tier group was created
downgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
downgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
downgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
downgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
downgradeSchedule_strategy	varchar2		☐	Schedule strategy type.
downgradeStrategy	varchar2		☐	Select real_time to downgrade tier on real time balance updates. Select membership_anniversary to downgrade tier on subscription anniversary. Select tier_anniversary to downgrade tier on tier anniversary.
id	guid	ID	☐	Tier group unique identifier
loyaltyProgramId	guid		☐	Associated loyalty program Id
name	varchar2	Name	☐	Tier group name
updatedAt	datetime		☐	Timestamp when the tier group was last updated
upgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
upgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
upgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
upgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
upgradeSchedule_strategy	varchar2		☐	Schedule strategy type.
upgradeStrategy	varchar2		☐	Select real_time to upgrade tier on real time balance updates. Select membership_anniversary to upgrade tier on subscription anniversary. Select tier_anniversary to upgrade tier on tier anniversary.

LoyaltyProgramTier_ItemsAccessConditions

List tiersReturns the list of tiers defined within the loyalty program.

Catalog: Brevo

Schema: Tier

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 LoyaltyProgramTier_ItemsAccessConditions. A value must be provided at all times for required

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
pid		☑		Loyalty Program ID
version		☐		Select 'active' to retrieve list of all tiers w hich are live for clients. Select draft to retrieve list of all non deleted tiers. (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
balanceDefinitionId	guid		☐	Balance definition identifier
createdAt_2	datetime		☐	
createdAt	datetime	Creation Date	☐	
groupId	guid		☐	Associated group Id
imageRef	varchar2		☐	Tier image reference
loyaltyProgramId	guid		☐	Associated loyalty program Id
minimumValue	int64		☐	Minimum value required to access this tier
name	varchar2	Name	☐	Tier name
tierId	guid		☐	Tier id
updatedAt_2	datetime		☐	
updatedAt	datetime		☐	

LoyaltyProgramTier_ItemsTierRewards

List tiersReturns the list of tiers defined within the loyalty program.

Catalog: Brevo

Schema: Tier

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 LoyaltyProgramTier_ItemsTierRewards. 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
pid		☑		Loyalty Program ID
version		☐		Select 'active' to retrieve list of all tiers which are live for clients. Select draft to retrieve list of all non deleted tiers. (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt_2	datetime		☐	
createdAt	datetime	Creation Date	☐	
groupId	guid		☐	Associated group Id
imageRef	varchar2		☐	Tier image reference
loyaltyProgramId	guid		☐	Associated loyalty program Id
name	varchar2	Name	☐	Tier name
rewardId	guid		☐	Reward to be attributed on tier assignment
tierId	guid		☐	Tier id
updatedAt_2	datetime		☐	
updatedAt	datetime		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

LoyaltyProgramTiersByPid

List tiersReturns the list of tiers defined within the loyalty program.

Catalog: Brevo

Schema: Tier

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 LoyaltyProgramTiersByPid. 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
pid		☑		Loyalty Program ID
version		☐		Select 'active' to retrieve list of all tiers w hich are live for clients. Select draft to retrieve list of all non deleted tiers. (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	
groupId	guid		☐	Associated group Id
imageRef	varchar2		☐	Tier image reference
loyaltyProgramId	guid		☐	Associated loyalty program Id
name	varchar2	Name	☐	Tier name
tierId	guid		☐	Tier id
updatedAt	datetime		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

TierGroupByPidAndGid

Get tier groupReturns tier group information.

Catalog: Brevo

Schema: Tier

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 TierGroupByPidAndGid. 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
gid		☑		Tier group ID
pid		☑		Loyalty Program ID
version		☐		Select active to retrieve active version of tier group. Select draft to retrieve latest changes in tier group. (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Timestamp when the tier group was created
downgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
downgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
downgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
downgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
downgradeSchedule_strategy	varchar2		☐	Schedule strategy type.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
dow ngradeStrategy	varchar2		☐	Select real_time to dow ngrade tier on real time balance updates. Select membership_anniversary to dow ngrade tier on subscription anniversary. Select tier_anniversary to dow ngrade tier on tier anniversary.
id	guid	ID	☐	Tier group unique identifier
loyaltyProgramId	guid		☐	Associated loyalty program Id
name	varchar2	Name	☐	Tier group name
updatedAt	datetime		☐	Timestamp w hen the tier group w as last updated
upgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
upgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
upgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
upgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
upgradeSchedule_strategy	varchar2		☐	Schedule strategy type.
upgradeStrategy	varchar2		☐	Select real_time to upgrade tier on real time balance updates. Select membership_anniversary to upgrade tier on subscription anniversary. Select tier_anniversary to upgrade tier on tier anniversary.

TierGroupItemsByPid

List tier groupsReturns the list of tier groups defined within the loyalty program.

Catalog: Brevo

Schema: Tier

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 TierGroupItemsByPid. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Required	Default Value	Documentation
pid		☑		Loyalty Program ID
version		☐		Select 'active' to retrieve list of all tier groups which are live for clients. Select draft to retrieve list of all non deleted tier groups. (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Timestamp when the tier group was created
downgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
downgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
downgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
downgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
downgradeSchedule_strategy	varchar2		☐	Schedule strategy type.
downgradeStrategy	varchar2		☐	Select real_time to downgrade tier on real time balance updates. Select membership_anniversary to downgrade tier on subscription anniversary. Select tier_anniversary to downgrade tier on tier anniversary.
id	guid	ID	☐	Tier group unique identifier
loyaltyProgramId	guid		☐	Associated loyalty program Id
name	varchar2	Name	☐	Tier group name
TEXT	guid	Text	☐	
updatedAt	datetime		☐	Timestamp when the tier group was last updated
upgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
upgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
upgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
upgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
upgradeSchedule_strategy	varchar2		☐	Schedule strategy type.
upgradeStrategy	varchar2		☐	Select real_time to upgrade tier on real time balance updates. Select membership_anniversary to upgrade tier on subscription anniversary. Select tier_an-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				niversary to upgrade tier on tier anniversary.

TierGroupOrdersByPidAndGid

Get tier groupReturns tier group information.

Catalog: Brevo

Schema: Tier

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 TierGroupOrdersByPidAndGid. 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
gid		☑		Tier group ID
pid		☑		Loyalty Program ID
version		☐		Select active to retrieve active version of tier group. Select draft to retrieve latest changes in tier group. (Values: active, draft)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Timestamp w hen the tier group w as created
dow ngradeSchedule_duration-Modifier	varchar2		☐	Modifier for the duration.
dow ngradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
dow ngradeSchedule_dura-tionValue	int64		☐	Duration value for the schedule.
dow ngradeSchedule_sched-uledDate	varchar2		☐	Scheduled date in DD/MM format.
dow ngradeSchedule_strategy	varchar2		☐	Schedule strategy type.
dow ngradeStrategy	varchar2		☐	Select real_time to dow ngrade tier on real time balance updates. Select membership_anniversary to dow ngrade tier on subscrip-tion anniversary. Select tier_an-niversary to dow ngrade tier on tier anniversary.
id	guid	ID	☐	Tier group unique identifier
loyaltyProgramId	guid		☐	Associated loyalty program Id
name	varchar2	Name	☐	Tier group name
TEXT	guid	Text	☐	
updatedAt	datetime		☐	Timestamp w hen the tier group w as last updated
upgradeSchedule_durationModifier	varchar2		☐	Modifier for the duration.
upgradeSchedule_durationUnit	varchar2		☐	Duration unit for the schedule.
upgradeSchedule_durationValue	int64		☐	Duration value for the schedule.
upgradeSchedule_scheduledDate	varchar2		☐	Scheduled date in DD/MM format.
upgradeSchedule_strategy	varchar2		☐	Schedule strategy type.
upgradeStrategy	varchar2		☐	Select real_time to upgrade tier on real time balance updates. Select membership_anniversary to upgrade tier on subscription anniversary. Select tier_an-niversary to upgrade tier on tier anniversary.

3.1.28 Schema: TransactionalEmails

3.1.28.1 Tables

BlockedDomains: Sendinblue Blocked Domains

Get the list of blocked domainsGet the list of blocked domains

Catalog: Brevo

Schema: TransactionalEmails

Label: Blocked Domains

Retrieve: true

Insert: true

Update: false

Delete: false

Table Columns

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	Text	☐	name of blocked domain

EmailEvents

Get all your transactional email activity (unaggregated events)This endpoint will show the aggregated stats for past 30 days by default if `startDate` and `endDate` OR `days` is not passed. The date range can not exceed 90 days

Catalog: Brevo

Schema: TransactionalEmails

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 EmailEvents. 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
days		☐		Number of days in the past including today (positive integer, maximum 90). _Not compatible with 'startDate' and 'endDate'_. Defaults to 30 if neither dates nor days are provided.
email		☐		Filter the report for a specific email addresses
endDate		☐		**Mandatory if startDate is used.** Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
event		☐		Filter the report for a specific event type (Values: bounces, hardBounces, softBounces, delivered, spam, requests, opened, clicks, invalid, deferred, blocked,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				unsubscribed, error, loadedByProxy)
messageId		☐		Filter on a specific message id
startDate		☐		**Mandatory if endDate is used.** Starting date of the report (YYYY-MM-DD). Must be low er than equal to endDate
tags		☐		Filter the report for tags (serial- ized and urlencoded array)
templateId		☐		Filter on a specific template id

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
date	varchar2	Date	☐	UTC date-time on w hich the event has been generated
email	varchar2	Email Address	☐	Email address w hich generates the event
event	varchar2	Event	☐	Event w hich occurred
from	varchar2	From	☐	Sender email from w hich the emails are sent
ip	varchar2	IP	☐	IP from w hich the user has opened the email or clicked on the link (only available if the event is opened or clicks)
link	varchar2	Link	☐	The link w hich is sent to the user (only available if the event is re- quests or opened or clicks)
messageId	varchar2	Message ID	☐	Message ID w hich generated the event
reason	varchar2	Reason	☐	Reason of bounce (only avail- able if the event is hardbounce or softbounce)
subject	varchar2	Subject	☐	Subject of the event
tag	varchar2	Tag	☐	Tag of the email w hich gener- ated the event
templateId	int64	Template ID	☐	ID of the template (only available if the email is template based)

ScheduledEmailById

Fetch scheduled emails by batchId or messageIdFetch scheduled batch of emails by batchId or single scheduled email by messageId (Can retrieve data upto 30 days old)

Catalog: Brevo

Schema: TransactionalEmails

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 ScheduledEmailById. 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		☐		Mandatory if `startDate` is used. Ending date (YYYY-MM-DD) till w hich you w ant to fetch the list. Maximum time period that can be selected is one month.
identifier		☑		The `batchId` of scheduled emails batch (must be a valid UUIDv4) or the `messageId` of scheduled email (enclosed in angle brackets w ith @ sign, e.g. `<...@domain>`). When using `messageId`, the `limit`, `offset`, `sort`, and `status` query parameters are ignored.
startDate		☐		Mandatory if `endDate` is used. Starting date (YYYY-MM-DD) from w hich you w ant to fetch the list. Cannot be more than 30 days older than the current date.
status		☐		Filter the records by `status` of the scheduled email batch or message. Not valid w hen identifier is `messageId`. (Values: processed, inProgress, queued)

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
count	int64	Count	☐	Total number of batches

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
createdAt	datetime	Creation Date	☐	Datetime on w hich the email w as scheduled
scheduledAt	datetime	Schedule Date	☐	Datetime for w hich the email w as scheduled
status	varchar2	Status	☐	Current status of the scheduled email

SmtpReports: Sendinblue SMTP Reports

Get your transactional email activity aggregated over a period of timeThis endpoint will show the aggregated stats for past 90 days by default if `startDate` and `endDate` OR `days` is not passed. The date range can not exceed 90 days

Catalog: Brevo

Schema: TransactionalEmails

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		☐		Number of days in the past including today (positive integer, maximum 90). _Not compatible w ith 'startDate' and 'endDate'_. Defaults to 90 if neither dates nor days are provided.
endDate		☐		**Mandatory if startDate is used.** Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
startDate		☐		**Mandatory if endDate is used.** Starting date of the report

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				(YYYY-MM-DD). Must be low er than equal to endDate
tag		☐		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 time-frame
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 time-frame
range	varchar2	Range	☐	Time frame of the report
requests	int64	Requests	☐	Number of requests for the time-frame
softBounces	int64	Soft-bounces	☐	Number of softbounces for the timeframe
spamReports	int64	Spam Reports	☐	Number of complaint (spam re-port) 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

SmtpReports2: Sendinblue SMTP Reports

Get your transactional email activity aggregated per dayThis endpoint will show the aggreg-ated stats per day for the past 10 days by default if `startDate` and `endDate` OR `days` is not passed. The date range can not exceed 30 days.

Catalog: Brevo

Schema: TransactionalEmails

Label: SMTP Reports

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 SmtpReports2. 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		☐		Number of days in the past including today (positive integer, maximum 30). _Not compatible with 'startDate' and 'endDate'_
endDate		☐		**Mandatory if startDate is used.** Ending date of the report (YYYY-MM-DD)
startDate		☐		**Mandatory if endDate is used.** Starting date of the report (YYYY-MM-DD)
tag		☐		Tag of the emails

Columns of Table Function

The columns of the table function SmtpReports2 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
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

SmtpTemplateById: Sendinblue SMTP Template by ID

Returns the template informationRetrieve the full details of a specific transactional email template by its numeric ID or custom template identifier string. The response includes the template name, subject, sender information, HTML content, active status, creation and modification timestamps, reply-to address, tag, and a `doiTemplate` flag indicating whether the template is a double opt-in template (detected by the presence of optin-related tags or variables in the content).

Catalog: Brevo

Schema: TransactionalEmails

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		☑		ID of the template. Can be a numeric template ID or a custom template identifier string (alphanumeric, hyphens, and under-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				scores only, max 64 characters, must start w ith a letter).

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	varchar2	Creation Date	☐	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
customTemplateId	varchar2		☐	Custom template identifier, if one w as assigned during template creation. Only present w hen the template has a custom ID.
doiTemplate	char	DOI Template	☐	It is true if template is a valid Double opt-in (DOI) template, oth-erw ise it is false. This field w ill be available only in case of single template detail call.
htmlContent	varchar2	HTML Content	☐	HTML content of the template
id	int64	ID	☐	ID of the template
isActive	char	Is Active	☐	Status of template (true=active, false=inactive)
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the template
replyTo	varchar2	Reply-to	☐	Email defined as the "Reply to" for the template
sender_email	varchar2	Sender Email Address	☐	From email for the template
sender_id	varchar2	Sender ID	☐	Sender id of the template
sender_name	varchar2	Sender Name	☐	Sender name for the template
subject	varchar2	Subject	☐	Subject of the template
tag	varchar2	Tag	☐	Tag of the template
testSent	char	Test Sent	☐	Status of test sending for the template (true=test email has been sent, false=test email has not been sent)
toField	varchar2	To Field	☐	Customisation of the "to" field for the template

SmtpTemplates: Sendinblue SMTP Templates

Get the list of email templatesRetrieve a paginated list of all transactional email templates (in-cluding automation templates) with their details such as name, subject, sender, status, HTML content, and timestamps. Results default to 50 per page (max 1000) and are sorted in descending creation order unless overridden. You can filter by active/inactive status using

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

`templateStatus` and by editor type using `editorType` (currently only `richTextEditor` is supported).

Catalog: Brevo

Schema: TransactionalEmails

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
editorType		☐		Filter on the editor type used to create the template. Currently only `richTextEditor` is supported as a filter value. (Values: richTextEditor)
templateStatus		☐		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	varchar2	Creation Date	☐	Creation UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
customTemplateId	varchar2		☐	Custom template identifier, if one was assigned during template creation. Only present when the template has a custom ID.
doiTemplate	char	DOI Template	☐	It is true if template is a valid Double opt-in (DOI) template, oth-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				erw ise it is false. This field w ill be available only in case of single template detail call.
htmlContent	varchar2	HTML Content	☐	HTML content of the template
id	int64	ID	☐	ID of the template
isActive	char	Is Active	☐	Status of template (true=active, false=inactive)
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the template
replyTo	varchar2	Reply-to	☐	Email defined as the "Reply to" for the template
sender_email	varchar2	Sender Email Address	☐	From email for the template
sender_id	varchar2	Sender ID	☐	Sender id of the template
sender_name	varchar2	Sender Name	☐	Sender name for the template
subject	varchar2	Subject	☐	Subject of the template
tag	varchar2	Tag	☐	Tag of the template
testSent	char	Test Sent	☐	Status of test sending for the template (true=test email has been sent, false=test email has not been sent)
toField	varchar2	To Field	☐	Customisation of the "to" field for the template

TransacBlockedContacts: Sendinblue Transaction Blocked Contacts

Get the list of blocked or unsubscribed transactional contactsRetrieve a paginated list of transactional contacts that have been blocked or unsubscribed, along with the reason for blocking (e.g. hard bounce, admin blocked, spam complaint, or unsubscription via email/API/Marketing Automation). Both `startDate` and `endDate` must be provided together when filtering by date range, and neither date can be in the future. Results default to 50 per page (max 100) and are sorted in descending order of record creation unless overridden with the `sort` parameter.

Catalog: Brevo

Schema: TransactionalEmails

Primary Keys: email

Label: Transaction Blocked Contacts

Retrieve: true

Insert: false

Update: false

Delete: true

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) till which you want to fetch the blocked or unsubscribed contacts
startDate		☐		**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	varchar2	Blocked at	☐	Date when the contact was blocked or unsubscribed on
email	varchar2	Email Address	☐	Email address of the blocked or unsubscribed contact
reason_code	varchar2	Reason Code	☐	Reason code for blocking / unsubscribing (This code is safe for comparison)
reason_message	varchar2	Reason Message	☐	Reason for blocking / unsubscribing (This string is not safe for comparison)
senderEmail	varchar2	Sender Email	☐	Sender email address of the blocked or unsubscribed contact

TransacEmailContent_Events

Get the personalized content of a sent transactional emailYou can get the uuid using either of the following methods:Send a GET request to https://api.brevo.com/v3/smtp/emails and pass the message_id in the url. Use your api-key to authenticate the request and you will get your uuid as a response.The uuid can also be fetched from the transactional logs page in your Brevo account, from the address URL.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: TransactionalEmails

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		☑		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	varchar2	Body	☐	Actual content of the transactional email that has been sent
date	varchar2	Date	☐	Date on which transactional email was sent
email	varchar2	Email Address	☐	Email address to which transactional email has been sent
name	varchar2	Name	☐	Name of the event that occurred on the sent email
subject	varchar2	Subject	☐	Subject of the sent email
templateId	int64	Template ID	☐	Id of the template
time	varchar2	Time	☐	Time at which the event occurred

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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: Brevo

Schema: TransactionalEmails

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		☐		**Mandatory if templateId and messageId are not passed in query filters.** Email address to which transactional email has been sent.
endDate		☐		**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		☐		**Mandatory if templateId and email are not passed in query filters.** Message ID of the transactional email sent.
startDate		☐		**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		☐		**Mandatory if email and messageId are not passed in query filters.** Id of the template that was used to compose transactional email.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	Date	☐	Date on which transactional email was sent
email	varchar2	Email Address	☐	Email address to which transactional email has been sent
from	varchar2	From	☐	Email address of the sender from which the email was sent
messageId	varchar2	Message ID	☐	Message Id of the sent email
subject	varchar2	Subject	☐	Subject of the sent email
templateId	int64	Template ID	☐	Id of the template
TEXT	varchar2	Text	☐	
uuid	varchar2	UUID	☐	Unique id of the email sent to a particular contact

TransactionEmailContentByUuuid

Get the personalized content of a sent transactional emailYou can get the uuid using either of the following methods:Send a GET request to https://api.brevo.com/v3/smtp/emails and pass the message_id in the url. Use your api-key to authenticate the request and you will get your uuid as a response.The uuid can also be fetched from the transactional logs page in your Brevo account, from the address URL.

Catalog: Brevo

Schema: TransactionalEmails

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 TransactionEmailContentByUuuid. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Columns of Table Function

The columns of the table function TransactionEmailContentByUuuid 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	varchar2	Body	☐	Actual content of the transactional email that has been sent
date	varchar2	Date	☐	Date on which transactional email was sent
email	varchar2	Email Address	☐	Email address to which transactional email has been sent
subject	varchar2	Subject	☐	Subject of the sent email
templateId	int64	Template ID	☐	Id of the template

TransactionEmailsByEmail

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: Brevo

Schema: TransactionalEmails

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 TransactionEmailsByEmail. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
email		☐		**Mandatory if templateId and messageId are not passed in query filters.** Email address to which transactional email has been sent.
endDate		☐		**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		☐		**Mandatory if templateId and email are not passed in query filters.** Message ID of the transactional email sent.
startDate		☐		**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		☐		**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 TransactionEmailsByEmail are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
date	varchar2	Date	☐	Date on which transactional email was sent
email	varchar2	Email Address	☐	Email address to which transactional email has been sent
from	varchar2	From	☐	Email address of the sender from which the email was sent
messageId	varchar2	Message ID	☐	Message Id of the sent email
subject	varchar2	Subject	☐	Subject of the sent email
templateId	int64	Template ID	☐	Id of the template
uuid	varchar2	UUID	☐	Unique id of the email sent to a particular contact

3.1.29 Schema: TransactionalSMS

3.1.29.1 Tables

SmsEvents: Sendinblue SMS Events

Get all your SMS activity (unaggregated events)Retrieve a paginated list of individual SMS event records (unaggregated), including event type, phone number, message ID, timestamp, tag, and reason or reply content where applicable. Results default to 50 per page (max 100)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

and are sorted in descending order unless overridden. Filter by date range (`startDate`/`endDate`), past `days` (not compatible with date range), specific `event` type (e.g. delivered, bounces, replies), `phoneNumber`, or `tags`. Bounce events include the failure reason, and reply events include the reply content.

Catalog: Brevo

Schema: TransactionalSMS

Label: SMS Events

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

Parameters of Table Function

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

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

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

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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	varchar2	Date	☐	UTC date-time on w hich the event has been generated
event	varchar2	Event	☐	Event w hich occurred
messageId	varchar2	Message ID	☐	Message ID w hich generated the event
phoneNumber	varchar2	Phone Number	☐	Phone number w hich has generated the event
reason	varchar2	Reason	☐	Reason of bounce (only available if the event is hardbounce or softbounce)
reply	varchar2	Reply	☐	
tag	varchar2	Tag	☐	Tag of the SMS w hich generated the event

TransactionAggregatedSmsReports: Sendinblue SMS Aggregated Transactions Report

Get your SMS activity aggregated over a period of timeRetrieve an aggregated report of your transactional SMS activity over a specified time period, including counts for requests, delivered, hard bounces, soft bounces, blocked, unsubscribed, replied, accepted, rejected, and skipped messages. Filter by date range using `startDate` and `endDate` (both required together, YYYY-MM-DD format) or by a number of past `days` (not compatible with date range). You can further narrow results by `tag`. If no date filter is provided, the report covers all available data and returns the auto-detected date range.

Catalog: Brevo

Schema: TransactionalSMS

Label: SMS Aggregated Transactions Report

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

Parameters of Table Function

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

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

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		Number of days in the past including today (positive integer). **Not compatible with startDate and endDate**
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the report
startDate		☐		**Mandatory if endDate is used.** Starting date (YYYY-MM-DD) of the report
tag		☐		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 SMS 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	varchar2	Range	☐	Time frame of the report
rejected	int64	Rejected	☐	Number of rejected SMS for the timeframe
replied	int64	Replied	☐	Number of answered SMS for the timeframe
requests	int64	Requests	☐	Number of requests for the timeframe
skipped	int64		☐	Number of skipped SMS for the timeframe
softBounces	int64	Soft-bounces	☐	Number of softbounces for the timeframe
unsubscribed	int64	Unsubscribed	☐	Number of unsubscription for the timeframe

TransactionalSmsReports

Get your SMS activity aggregated per dayRetrieve a day-by-day breakdown of your transactional SMS activity, with each entry containing the date and counts for requests, delivered, hard bounces, soft bounces, blocked, unsubscribed, replied, accepted, rejected, and skipped messages. Filter by date range using `startDate` and `endDate` (both required together, YYYY-MM-DD format), by a number of past `days` (not compatible with date range), or by

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

`tag`. Results are sorted in descending order by default unless overridden with the `sort` parameter.

Catalog: Brevo

Schema: TransactionalSMS

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 TransactionalSmsReports. 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		☐		Number of days in the past including today (positive integer). **Not compatible with 'startDate' and 'endDate'**
endDate		☐		**Mandatory if startDate is used.** Ending date (YYYY-MM-DD) of the report
startDate		☐		**Mandatory if endDate is used.** Starting date (YYYY-MM-DD) of the report
tag		☐		Filter on a tag

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
accepted	int64	Accepted	☐	Number of accepted SMS for the date
blocked	int64	Blocked	☐	Number of blocked contact for the date
date	datetime	Date	☐	Date for which statistics are retrieved

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
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 SMS for the date
replied	int64	Replied	☐	Number of answered SMS for the date
requests	int64	Requests	☐	Number of requests for the date
skipped	int64		☐	Number of skipped SMS for the date
softBounces	int64	Soft-bounces	☐	Number of softbounces for the date
unsubscribed	int64	Unsubscribed	☐	Number of unsubscription for the date

3.1.30 Schema: TransactionalWhatsApp

3.1.30.1 Tables

WhatsappEventReports

Get all your WhatsApp activity (unaggregated events)This endpoint will show the unaggregated statistics for WhatsApp activity (30 days by default if `startDate` and `endDate` or `days` is not passed. The date range can not exceed 90 days)

Catalog: Brevo

Schema: TransactionalWhatsApp

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 WhatsappEventReports. 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
contactNumber		☐		Filter results for specific contact (WhatsApp Number with country

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				code. Example, 85264318721)
days		☐		Number of days in the past including today (positive integer). _Not compatible with 'startDate' and 'endDate'_
endDate		☐		**Mandatory if startDate is used.** Ending date of the report (YYYY-MM-DD). Must be greater than equal to startDate
event		☐		Filter the report for a specific event type (Values: sent, delivered, read, error, unsubscribe, reply, soft-bounce)
startDate		☐		**Mandatory if endDate is used.** Starting date of the report (YYYY-MM-DD). Must be lower than equal to endDate

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
body	varchar2	Body	☐	Text of the reply (will be there only in case of `reply` event with text)
contactNumber	varchar2		☐	WhatsApp Number with country code. Example, 85264318721
date	varchar2	Date	☐	UTC date-time on which the event has been generated
event	varchar2	Event	☐	Event which occurred
mediaUrl	varchar2		☐	Url of the media reply (will be there only in case of `reply` event with media)
messageId	varchar2	Message ID	☐	Message ID which generated the event
reason	varchar2	Reason	☐	Reason for the event (will be there in case of `error` and `soft-bounce` events)
senderNumber	varchar2		☐	WhatsApp Number with country code. Example, 85264318721

3.1.31 Schema: User

3.1.31.1 Tables

InvitedUsers

Get the list of all your usersRetrieves the list of all users associated with your organization, including both active and pending invited users. Each user entry includes their email address, owner status, current invitation status, and feature access levels for marketing, CRM, and conversations.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Catalog: Brevo

Schema: User

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 InvitedUsers are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
feature_access_conversations	varchar2		☐	Conversations features accessiblity.
feature_access_crm	varchar2		☐	CRM features accessiblity.
feature_access_marketing	varchar2		☐	Marketing features accessiblity.
is_owner	varchar2		☐	Flag for indicating is user owner of the organization.
status	varchar2	Status	☐	Status of the invited user.

UserPermission_Privileges

Check user permissionRetrieves the granular feature-level permissions assigned to a specific user in the organization, identified by their email address. The response includes the user''s current status (active or pending) and a detailed list of privileges specifying which features and permission levels are granted.

Catalog: Brevo

Schema: User

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 UserPermission_Privileges. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
email		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
feature	varchar2		☐	
status	varchar2	Status	☐	Status of the invited user.

UserPermission_PrivilegesPermissions

Check user permissionRetrieves the granular feature-level permissions assigned to a specific user in the organization, identified by their email address. The response includes the user''s current status (active or pending) and a detailed list of privileges specifying which features and permission levels are granted.

Catalog: Brevo

Schema: User

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 UserPermission_PrivilegesPermissions. 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		☑		Email of the invited user.

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.
feature	varchar2		☐	
status	varchar2	Status	☐	Status of the invited user.
TEXT	varchar2	Text	☐	

UserPermissionsByEmail

Check user permissionRetrieves the granular feature-level permissions assigned to a specific user in the organization, identified by their email address. The response includes the user''s current status (active or pending) and a detailed list of privileges specifying which features and permission levels are granted.

Catalog: Brevo

Schema: User

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 UserPermissionsByEmail. 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
email		☑		Email of the invited user.

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
email	varchar2	Email Address	☐	Email address of the user.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
status	varchar2	Status	☐	Status of the invited user.

3.1.32 Schema: Views
3.1.32.1 Views
AbTestCampaignResults: Sendinblue A/B Test Campaign Results

Catalog: Brevo

Schema: Views

Label: A/B Test Campaign Results

This is a read-only view. The Sendinblue API may not support changing the data or the 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	varchar2	Name	☐	Name of the campaign
CampaignSubject	varchar2	Subject	☐	Subject of the campaign. Only available if `abTesting` flag of the campaign is `false`
clickRate	varchar2	Clickrate	☐	Click rate for current w inning version
openRate	varchar2	Open Rate	☐	Open rate for current w inning version
statistics_clicks_Version_A	varchar2		☐	percentage of an event for version A
statistics_clicks_Version_B	varchar2		☐	percentage of an event for version B
statistics_complaints_Version_A	varchar2		☐	percentage of an event for version A
statistics_complaints_Version_B	varchar2		☐	percentage of an event for version B
statistics_hardBounces_Version_A	varchar2		☐	percentage of an event for version A
statistics_hardBounces_Version_B	varchar2		☐	percentage of an event for version B
statistics_openers_Version_A	varchar2		☐	percentage of an event for version A
statistics_openers_Version_B	varchar2		☐	percentage of an event for version B
statistics_softBounces_Version_A	varchar2		☐	percentage of an event for version A
statistics_softBounces_Version_B	varchar2		☐	percentage of an event for version B

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
statistics_unsubscribed_Version_A	varchar2		☐	percentage of an event for version A
statistics_unsubscribed_Version_B	varchar2		☐	percentage of an event for version B
winningCriteria	varchar2	Winning Criteria	☐	Criteria chosen for winning version (Open/Click)
winningSubjectLine	varchar2	Winning Subject Line	☐	Subject Line of current winning version
winningVersion	varchar2	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	varchar2	Winning Version Rate	☐	Open/Click rate for the winner version

FolderLists: Sendinblue Folder Lists

Catalog: Brevo

Schema: Views

Label: Folder Lists

This is a read-only view. The Sendinblue API may not support changing the data or the 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	varchar2	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	varchar2	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
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the list

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ListCampaignStatistics: Sendinblue List Campaign Statistics

Catalog: Brevo

Schema: Views

Label: List Campaign Statistics

This is a read-only view. The Sendinblue API may not support changing the data or the 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	varchar2	Creation Date	☐	Creation UTC date-time of the list (YYYY-MM-DDTHH:mm:ss.SSSZ)
dynamicList	char	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	varchar2	Name	☐	Name of the list
stats_appleMppOpens	int64		☐	Numbers of times your email has been opened automatically through Apple MPP.
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_estimatedViews	int64		☐	Rate of recipients without any privacy protection option enabled in their email client, applied to all delivered emails
stats_hardBounces	int64	#Hard-bounces	☐	Number of hard bounces 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_opensRate	double		☐	Percentage of recipients who open the email out of your total number of recipients. Depending on your Campaign settings, they may include Apple MPP opens.
stats_returnBounce	int64	#Return-bounces	☐	Total number of non-delivered campaigns for a particular campaign id.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
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 email client
stats_trackableViewsRate	double		☐	Rate of 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
uniqueSubscribers	int64	#Unique Subscribers	☐	Number of unique contacts in the list

3.1.33 Schema: Webhooks

3.1.33.1 Tables

Webhook_Headers

Get a webhook detailsRetrieves detailed information about a specific webhook configuration.Use this to:- Get complete webhook configuration and settings- Check webhook event subscriptions and triggers- Review authentication and security settings- Verify webhook URL and custom headers- Access webhook creation and modification historyKey information returned:- Complete webhook details and configuration- Event types and channel subscriptions- Authentication credentials and methods- Custom headers and request settings- Webhook status and activity information

Catalog: Brevo

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 Webhook_Headers. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		Id of the webhook

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
auth	varchar2		☐	Authentication header to be send with the webhook requests
batched	char		☐	Batching configuration of the webhook, we send batched webhooks if itstrue
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2	Description	☐	Description of the webhook
domain	varchar2	Domain	☐	Inbound domain of the webhook, only returned for inbound type webhooks
id	int64	ID	☐	ID of the webhook
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
type	varchar2	Type	☐	Type of webhook (marketing, transactional, or inbound)
url	varchar2	URL	☐	URL of the webhook

WebhookById: Sendinblue Webhook by ID

Get a webhook detailsRetrieves detailed information about a specific webhook configuration.Use this to:- Get complete webhook configuration and settings- Check webhook event subscriptions and triggers- Review authentication and security settings- Verify webhook URL and custom headers- Access webhook creation and modification historyKey information returned:- Complete webhook details and configuration- Event types and channel subscriptions- Authentication credentials and methods- Custom headers and request settings- Webhook status and activity information

Catalog: Brevo

Schema: Webhooks

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		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
auth	varchar2		☐	Authentication header to be send with the webhook requests
batched	char		☐	Batching configuration of the webhook, we send batched webhooks if itstrue
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2	Description	☐	Description of the webhook
domain	varchar2	Domain	☐	Inbound domain of the webhook, only returned for inbound type webhooks
id	int64	ID	☐	ID of the webhook
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
type	varchar2	Type	☐	Type of webhook (marketing, transactional, or inbound)
url	varchar2	URL	☐	URL of the webhook

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

WebhookEventsByWebhookId

Get a webhook detailsRetrieves detailed information about a specific webhook configuration.Use this to:- Get complete webhook configuration and settings- Check webhook event subscriptions and triggers- Review authentication and security settings- Verify webhook URL and custom headers- Access webhook creation and modification historyKey information returned:- Complete webhook details and configuration- Event types and channel subscriptions- Authentication credentials and methods- Custom headers and request settings- Webhook status and activity information

Catalog: Brevo

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		☑		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
auth	varchar2		☐	Authentication header to be send with the webhook requests
batched	char		☐	Batching configuration of the webhook, we send batched webhooks if itstrue
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2	Description	☐	Description of the webhook

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
domain	varchar2	Domain	☐	Inbound domain of the webhook, only returned for inbound type webhooks
id	int64	ID	☐	ID of the webhook
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
TEXT	varchar2	Text	☐	Events which will trigger the webhook when they occure
type	varchar2	Type	☐	Type of webhook (marketing, transactional, or inbound)
url	varchar2	URL	☐	URL of the webhook

Webhooks: Sendinblue Webhooks

Get all webhooksRetrieves all webhooks from your Brevo account with filtering and sorting options.Use this to:- Monitor webhook configurations and event handling- List webhooks by type (transactional, marketing, inbound)- Review webhook endpoints and authentication- Track webhook creation and modification history- Audit webhook event subscriptionsKey information returned:- Complete webhook details and configuration- Event types and channel subscriptions- Authentication and security settings- Webhook URLs and custom headers- Creation and modification timestamps

Catalog: Brevo

Schema: Webhooks

Primary Keys: id

Label: Webhooks

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

Parameters of Table Function

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

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select * from table(value1, value2, value3)` on a table with four 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		☐		Filter on webhook type (Values: marketing, transactional, inbound)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
auth	varchar2		☐	Authentication header to be send with the webhook requests
batched	char		☐	Batching configuration of the webhook, we send batched webhooks if itstrue
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
description	varchar2	Description	☐	Description of the webhook
domain	varchar2	Domain	☐	Inbound domain of the webhook, only returned for inbound type webhooks
id	int64	ID	☐	ID of the webhook
modifiedAt	varchar2	Modified Date	☐	Last modification UTC date-time of the webhook (YYYY-MM-DDTHH:mm:ss.SSSZ)
type	varchar2	Type	☐	Type of webhook (marketing, transactional, or inbound)
url	varchar2	URL	☐	URL of the webhook

3.1.34 Schema: WhatsAppCampaigns

3.1.34.1 Tables

WhatsAppCampaign_RecipientsExcludedLists

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppCampaign_RecipientsExcludedLists. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		Id of the campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
id	int64	ID	☐	ID of the WhatsApp Campaign
includedLists	varchar2		☐	List of included list IDs
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
segments	varchar2		☐	List of segment IDs
senderNumber	varchar2		☐	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	varchar2		☐	
template_category	varchar2		☐	description of the template
template_contains_button	char		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
template_display_header	char		☐	
template_header_type	varchar2		☐	type of header
template_hide_footer	char		☐	
template_language	varchar2		☐	language of the template
template_name	varchar2		☐	name of the template
TEXT	int64	Text	☐	
type	varchar2	Type	☐	Type of recipients (list or segment)

WhatsAppCampaign_RecipientsIncludedLists

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppCampaign_RecipientsIncludedLists. 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		☑		Id of the campaign

Columns of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

Name	Data Type	Label	Required	Documentation
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
excludedLists	varchar2		☐	List of excluded list IDs
id	int64	ID	☐	ID of the WhatsApp Campaign
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
segments	varchar2		☐	List of segment IDs
senderNumber	varchar2		☐	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	varchar2		☐	
template_category	varchar2		☐	description of the template
template_contains_button	char		☐	
template_display_header	char		☐	
template_header_type	varchar2		☐	type of header
template_hide_footer	char		☐	
template_language	varchar2		☐	language of the template
template_name	varchar2		☐	name of the template
TEXT	int64	Text	☐	
type	varchar2	Type	☐	Type of recipients (list or segment)

WhatsAppCampaign_RecipientsSegments

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppCampaign_RecipientsSegments. 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		☑		Id of the campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
excludedLists	varchar2		☐	List of excluded list IDs
id	int64	ID	☐	ID of the WhatsApp Campaign
includedLists	varchar2		☐	List of included list IDs
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
				Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
senderNumber	varchar2		☐	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	varchar2		☐	
template_category	varchar2		☐	description of the template
template_contains_button	char		☐	
template_display_header	char		☐	
template_header_type	varchar2		☐	type of header
template_hide_footer	char		☐	
template_language	varchar2		☐	language of the template
template_name	varchar2		☐	name of the template
TEXT	int64	Text	☐	
type	varchar2	Type	☐	Type of recipients (list or segment)

WhatsAppCampaign_TemplateBody_variables

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppCampaign_TemplateBody_variables. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☑		Id of the campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
button_type	varchar2		☐	
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
category	varchar2	Category	☐	description of the template
contains_button	char		☐	
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
datatype	varchar2		☐	
default	varchar2		☐	
display_header	char		☐	
header_type	varchar2		☐	type of header
hide_footer	char		☐	
id	int64	ID	☐	ID of the WhatsApp Campaign
language	varchar2		☐	language of the template
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name_2	varchar2		☐	name of the template
name	varchar2	Name	☐	
recipients_excludedLists	varchar2		☐	List of excluded list IDs
recipients_includedLists	varchar2		☐	List of included list IDs
recipients_segments	varchar2		☐	List of segment IDs
recipients_type	varchar2		☐	Type of recipients (list or segment)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
senderNumber	varchar2		☐	Sender of the WhatsApp Campaign
stats_delivered	int64	#Delivered	☐	
stats_notSent	int64		☐	
stats_read	int64		☐	
stats_sent	int64	#Sent	☐	
stats_unsubscribe	int64		☐	

WhatsAppCampaign_TemplateComponents

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppCampaign_TemplateComponents. 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		☑		Id of the campaign

Columns of Table Function

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
button_type	varchar2		☐	
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
category	varchar2	Category	☐	description of the template
contains_button	char		☐	
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
display_header	char		☐	
header_type	varchar2		☐	type of header
hide_footer	char		☐	
id	int64	ID	☐	ID of the WhatsApp Campaign
language	varchar2		☐	language of the template
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	name of the template
recipients_excludedLists	varchar2		☐	List of excluded list IDs
recipients_includedLists	varchar2		☐	List of included list IDs
recipients_segments	varchar2		☐	List of segment IDs
recipients_type	varchar2		☐	Type of recipients (list or segment)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
senderNumber	varchar2		☐	Sender of the WhatsApp Campaign
stats_delivered	int64	#Delivered	☐	
stats_notSent	int64		☐	
stats_read	int64		☐	
stats_sent	int64	#Sent	☐	
stats_unsubscribe	int64		☐	
text	varchar2	Text	☐	
type	varchar2	Type	☐	

WhatsAppCampaign_TemplateHeader_variables

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppCampaign_TemplateHeader_variables. 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		☑		Id of the campaign

Columns of Table Function

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

Name	Data Type	Label	Required	Documentation
button_type	varchar2		☐	
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
category	varchar2	Category	☐	description of the template
contains_button	char		☐	
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
datatype	varchar2		☐	
default	varchar2		☐	
display_header	char		☐	
header_type	varchar2		☐	type of header

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
hide_footer	char		☐	
id	int64	ID	☐	ID of the WhatsApp Campaign
language	varchar2		☐	language of the template
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
name_2	varchar2		☐	name of the template
name	varchar2	Name	☐	
recipients_excludedLists	varchar2		☐	List of excluded list IDs
recipients_includedLists	varchar2		☐	List of included list IDs
recipients_segments	varchar2		☐	List of segment IDs
recipients_type	varchar2		☐	Type of recipients (list or segment)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
senderNumber	varchar2		☐	Sender of the WhatsApp Campaign
stats_delivered	int64	#Delivered	☐	
stats_notSent	int64		☐	
stats_read	int64		☐	
stats_sent	int64	#Sent	☐	
stats_unsubscribe	int64		☐	

WhatsAppCampaignById: Sendinblue WhatsApp Campaign by ID

Get a WhatsApp campaignYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your accountThis API requires the List and Segment ids as recipients in Body params.You can use the below Contact endpoints to get the required information.[Get all the Lists](https://developers.brevo.com/reference/getlists-1)[Get all the Segments](https://developers.brevo.com/reference/getsegments)

Catalog: Brevo

Schema: WhatsAppCampaigns

Label: WhatsApp Campaign by ID

This is a read-only table function. The Sendinblue API may not support changing the data or the 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 op-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
campaignId		☑		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	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the WhatsApp Campaign
createdAt	varchar2	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	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
recipients_excludedLists	varchar2		☐	List of excluded list IDs
recipients_includedLists	varchar2		☐	List of included list IDs
recipients_segments	varchar2		☐	List of segment IDs
recipients_type	varchar2		☐	Type of recipients (list or segment)
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on w hich WhatsApp campaign is scheduled. Should be in YYYY-MM-DDTHH:mm:ss.SSSZ format
senderNumber	varchar2		☐	Sender of the WhatsApp Campaign
stats_delivered	int64	#Delivered	☐	
stats_notSent	int64		☐	
stats_read	int64		☐	
stats_sent	int64	#Sent	☐	
stats_unsubscribe	int64		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
template_button_type	varchar2		☐	
template_category	varchar2		☐	description of the template
template_contains_button	char		☐	
template_display_header	char		☐	
template_header_type	varchar2		☐	type of header
template_hide_footer	char		☐	
template_language	varchar2		☐	language of the template
template_name	varchar2		☐	name of the template

WhatsAppCampaigns: Sendinblue WhatsApp Campaigns

Return all your created WhatsApp campaignsRetrieve a paginated list of all your WhatsApp campaigns with their statistics and metadata. Results can be filtered by creation date range using startDate and endDate, with a default limit of 50 and maximum of 100 per page. The sort order defaults to descending by modification date.

Catalog: Brevo

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp 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 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		☐		**Mandatory if startDate is used**. Ending (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the WhatsApp campaigns created. **Prefer to pass your timezone in date-time format for accurate result**

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
startDate		☐		**Mandatory if endDate is used**. Starting (urlencoded) UTC date-time (YYYY-MM-DDTHH:mm:ss.SSSZ) to filter the WhatsApp 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. 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
campaignName	varchar2		☐	Name of the WhatsApp Campaign
campaignStatus	varchar2		☐	Status of the Whatsapp Campaign
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
errorReason	varchar2		☐	Error Reason associated with the WhatsApp campaign sending
id	int64	ID	☐	ID of the WhatsApp Campaign
invalidatedContacts	int64		☐	Count of invalidated contacts
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the WhatsApp campaign (YYYY-MM-DDTHH:mm:ss.SSSZ)
readPercentage	double		☐	Read percentage of the the whatsapp campaign created
scheduledAt	varchar2	Schedule Date	☐	UTC date-time on which 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	varchar2	Template ID	☐	Id of the WhatsApp template

WhatsAppConfigurations

Get your WhatsApp API account informationYou can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

follow the steps in the guide below.[Activating WhatsApp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your account

Catalog: Brevo

Schema: WhatsAppCampaigns

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 WhatsAppConfigurations are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
businessStatus	varchar2		☐	Verification status information of the Business account
phoneNumberNameStatus	varchar2		☐	Status of the name associated with WhatsApp Phone number
phoneNumberQuality	varchar2		☐	Quality status of phone number associated with WhatsApp account. There are three quality ratings. example - **High (GREEN) , Medium (YELLOW) and Low (RED)**
sendingLimit	varchar2		☐	Sending limit of the WhatsApp account
whatsappBusinessAccountId	varchar2		☐	Id of the WhatsApp business account
whatsappBusinessAccountStatus	varchar2		☐	Status information related to WhatsApp Api account

WhatsAppTemplates: Sendinblue WhatsApp Templates

Return all your created WhatsApp templatesRetrieve a paginated list of all your WhatsApp templates with their status, category, language, and metadata. Results can be filtered by creation date range and optionally by source (Automation or Conversations), with a default limit of 50 and maximum of 100 per page. The sort order defaults to descending by modification date.

Catalog: Brevo

Schema: WhatsAppCampaigns

Primary Keys: id

Label: WhatsApp Templates

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

Parameters of Table Function

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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		☐		**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**
source		☐		source of the template (Values: Automation, Conversations)
startDate		☐		**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	varchar2	Category	☐	category of the template
createdAt	varchar2	Creation Date	☐	Creation UTC date-time of the whatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
errorReason	varchar2		☐	Error reason in the template creation
id	int64	ID	☐	ID of the whatsApp template
language	varchar2		☐	Language in which template exists
modifiedAt	varchar2	Modified Date	☐	UTC date-time of last modification of the whatsApp template (YYYY-MM-DDTHH:mm:ss.SSSZ)
name	varchar2	Name	☐	Name of the whatsApp template

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
status	varchar2	Status	☐	Status of the w hatsApp template
type	varchar2	Type	☐	Type of the w hatsApp template

4 Package: dcr_metadata

4.1 Procedures

4.1.1 dcr_metadata.get_partitions: Sendinblue Data container metadata package

Get all partitions.

Documentation:

List all partitions.

5 Package: sib_balance

5.1 Procedures

5.1.1 sib_balance.begintransaction: Sendinblue Package for Balance operations that could not be mapped to a table.

Creates new transaction and returns information

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.begintransaction. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
amount	decimal	☑		Transaction amount. A positive value creates a credit transaction and a negative value creates a debit transaction (unless transactionType is explicitly provided).
transactionType	string	☐		Explicit transaction type. If not provided, the type is inferred from the sign of the amount (positive = credit, negative = debit). (Values: credit, debit)
balanceDefinitionId	guid	☑		Unique identifier (UUID) of the associated balance definition.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
contactId	int64	☐		Unique identifier of the contact involved in the transaction. Required unless `LoyaltySubscriptionId` is provided.
LoyaltySubscriptionId	string	☐		Unique identifier for the loyalty subscription. Required unless `contactId` is provided.
ttl	int64	☐		Time-to-live for the transaction in seconds. Must be at least 10 seconds if provided.
eventTime	datetime	☐		Timestamp specifying when the transaction event occurred (ISO 8601 / RFC 3339 format).
autoComplete	boolean	☐		Whether the transaction should be automatically completed.
balanceExpiryInMinutes	int64	☐		Expiry time for the balance in minutes. Must be greater than 0 if provided. Only applicable when autoComplete is true.

5.1.2 sib_balance.canceltransaction: Sendinblue Package for Balance operations that could not be mapped to a table.

Cancels transaction

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.canceltransaction. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_tid	guid	☑		Transaction Id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

5.1.3 sib_balance.completetransaction: Sendinblue Package for Balance operations that could not be mapped to a table.

Completes transaction

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.completetransaction. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_tid	guid	☑		Transaction Id

5.1.4 sib_balance.createbalancelimit: Sendinblue Package for Balance operations that could not be mapped to a table.

Creates balance limit and sends the created UUID along with the data

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.createbalancelimit. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_bdid	guid	☑		Balance Definition Id
constraintType	string	☑		Defines whether the limit applies to transaction count or amount. (Values: transaction, amount)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
durationUnit	string	☑		Unit of time for which the limit is applicable. (Values: day, week, month, year)
durationValue	int64	☑		Number of time units for the balance limit.
slidingSchedule	boolean	☐		Determines if the limit resets on a rolling schedule.
transactionType	string	☑		Specifies whether the limit applies to credit or debit transactions. (Values: credit, debit)
value	int64	☑		Maximum allowed value for the specified constraint type.

5.1.5 sib_balance.createbalanceorder: Sendinblue Package for Balance operations that could not be mapped to a table.

Returns created order

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.createbalanceorder. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
amount	decimal	☑		Order amount (must be non-zero).
balanceDefinitionId	string	☑		Unique identifier (UUID) of the associated balance definition.
contactId	int64	☑		Unique identifier of the contact placing the order (must be ≥ 1).
dueAt	string	☑		RFC3339 timestamp specifying when the order is due.
expiresAt	string	☐		Optional RFC3339 timestamp defining order expiration.
source	string	☑		Specifies the origin of the order. (Values: engine, user)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

5.1.6 sib_balance.deletebalancedefinition: Sendinblue Package for Balance operations that could not be mapped to a table.

Delete Balance definition

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.deletebalancedefinition. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_bdid	guid	☑		Balance Definition Id

5.1.7 sib_balance.deletebalancelimit: Sendinblue Package for Balance operations that could not be mapped to a table.

Delete balance limit

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.deletebalancelimit. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_bdid	guid	☑		Balance Definition Id
p_blid	guid	☑		Balance Limit Id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

5.1.8 sib_balance.updatebalancedefinition: Sendinblue Package for Balance operations that could not be mapped to a table.

Updates Balance definition

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.updatebalancedefinition. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_bdid	guid	☑		Balance Definition Id
balanceAvailabilityDurationModifier	string	☐		Defines when the balance expires within the selected duration. (Values: noModification, startOfPeriod, endOfPeriod)
balanceAvailabilityDurationUnit	string	☐		Unit of time for balance validity. (Values: day, week, month, year)
balanceAvailabilityDurationValue	int64	☐		Number of time units before the balance expires.
balanceExpirationDate	string	☐		Expiration date (`dd/mm` format) or empty if not applicable.
balanceOptionAmountOvertakingStrategy	string	☐		Defines whether partial credit is allowed when reaching max balance. (Values: strict, partial)
balanceOptionCreditRounding	string	☐		Rounding strategy for credit transactions. (Values: lower, upper, natural)
balanceOptionDebitRounding	string	☐		Rounding strategy for debit transactions. (Values: lower, upper, natural)
description	string	☐		Short description of the balance definition.
imageRef	string	☐		URL of an optional image reference.
maxAmount	decimal	☐		Maximum allowable balance amount.
maxCreditAmountLimit	decimal	☐		Maximum credit allowed per operation.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
maxDebitAmountLimit	decimal	☐		Maximum debit allow ed per operation.
isInternal	boolean	☐		Indicates w hether the balance definition is internal.
minAmount	decimal	☐		Minimum allow able balance amount.
name	string	☑		Name of the balance definition.
unit	string	☑		Unit of balance measurement. (Values: POINTS, EUR, USD, MXN, GBP, INR, CAD, SGD, RON, JPY, MYR, CLP, PEN, MAD, AUD, CHF, BRL)

5.1.9 sib_balance.updatebalancelimit: Sendinblue Package for Balance operations that could not be mapped to a table.

Updates balance limit

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_balance.updatebalancelimit. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program Id
p_bdid	guid	☑		Balance Definition Id
p_blid	guid	☑		Balance Limit Id
constraintType	string	☑		Defines w hether the limit applies to transaction count or amount. (Values: transaction, amount)
durationUnit	string	☑		Unit of time for w hich the limit is applicable. (Values: day, w eek, month, year)
durationValue	int64	☑		Number of time units for the balance limit.
slidingSchedule	boolean	☐		Determines if the limit resets on a rolling schedule.
transactionType	string	☑		Specifies w hether the limit applies to credit or debit transac-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				tions. (Values: credit, debit)
value	int64	☑		Maximum allow ed value for the specified constraint type.

6 Package: sib_contacts

6.1 Procedures

6.1.1 sib_contacts.addcontacttolist: Sendinblue Package for Contacts operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.addcontacttolist. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_listId	int64	☑		Id of the list

6.1.2 sib_contacts.createattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.

Create a new contact attribute under the specified category and name. The required body properties depend on the category: use "type" for normal, transactional, or category attributes; use "value" for calculated or global attributes; use "enumeration" for category attributes; and use "multiCategoryOptions" for normal multiple-choice attributes. None of the category or multicategory option values can exceed 200 characters.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.createattribute. 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 `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_attributeCategory	string	☑		Category of the attribute (Values: normal, transactional, category, calculated, global)
p_attributeName	string	☑		Name of the attribute
isRecurring	boolean	☐		Type of the attribute. **Use only if the attribute's category is 'calculated' or 'global'**
type	string	☐		Type of the attribute. **Use only if the attribute's category is 'normal', 'category' or 'transactional'** Type **user and multiple-choice** is only available if the category is **normal** attribute Type **id** is only available if the category is **transactional** attribute Type **category** is only available if the category is **category** attribute(Values: text, date, float, boolean, id, category, multiple-choice, user)
value	string	☐		Value of the attribute. **Use only if the attribute's category is 'calculated' or 'global'**

6.1.3 sib_contacts.createdoicontact: Sendinblue Package for Contacts operations that could not be mapped to a table.

<Note title="How to use attributes param?">attributes param in this endpoint is an object containing key-value pairs where values can be either a string, integer, array, or boolean. You can create key-value pairs with these four datatypes. When a value is an array, it should be an array of strings.</Note>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.createdoicontact. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
TEXT	decimal	☐		
email	string	☑		Email address where the confirmation email will be sent. This email address will be the identifier for all other contact attributes.
redirectionUrl	string	☑		URL of the web page that user will be redirected to after clicking on the double opt in URL. When editing your DOI template you can reference this URL by using the tag **{{ params.DOIurl }}**.
templateId	int64	☑		Id of the Double opt-in (DOI) template

6.1.4 sib_contacts.deleteattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.

Permanently delete an existing contact attribute by its category and name. The attribute must exist in the specified category (normal, transactional, category, calculated, or global), otherwise a 404 error is returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.deleteattribute. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_attributeCategory	string	☑		Category of the attribute (Values: normal, transactional, category, calculated, global)
p_attributeName	string	☑		Name of the existing attribute

6.1.5 sib_contacts.deletemultiattributeoptions: Sendinblue Package for Contacts operations that could not be mapped to a table.

Delete a specific option from an existing multiple-choice contact attribute. The attribute type must be "multiple-choice", and both the attribute name and the option to delete must already exist in your account.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.deletemultiattributeoptions. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_attributeType	string	☑		Type of the attribute (Values: multiple-choice)
p_multipleChoiceAttribute	string	☑		Name of the existing multiple-choice attribute
p_multipleChoiceAttributeOption	string	☑		Name of the existing multiple-choice attribute option that you want to delete

6.1.6 sib_contacts.importcontacts: Sendinblue Package for Contacts operations that could not be mapped to a table.

It returns the background process ID which on completion calls the notify URL that you have set in the input. **Note**: - Any contact attribute that doesn't exist in your account will be ignored at import end.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.importcontacts. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
disableNotification	boolean	☐	False	To disable email notification

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
emailBlacklist	boolean	☐	False	To blacklist all the contacts for email
emptyContactsAttributes	boolean	☐	False	To facilitate the choice to erase any attribute of the existing contacts w ith empty value. emptyContactsAttributes = true means the empty fields in your import w ill erase any attribute that currently contain data in Brevo, & emptyContactsAttributes = false means the empty fields w ill not affect your existing data (**only available if `updateExistingContacts` set to true **)
fileBody	string	☐		**Mandatory if fileUrl and jsonBody is not defined.** CSV content to be imported. Use semicolon to separate multiple attributes. **Maximum allow ed file body size is 10MB** . How ever w e recommend a safe limit of around 8 MB to avoid the issues caused due to increase of file body size w hile parsing. Please use fileUrl instead to import bigger files.
fileUrl	string	☐		**Mandatory if fileBody and jsonBody is not defined.** URL of the file to be imported (**no local file**). Possible file formats: #### .txt, .csv, .json
folderId	int64	☐		Id of the folder w here this new list shall be created. **Mandatory if listName is not empty**
listName	string	☐		List w ith listName w ill be created first and users w ill be imported in it. **Mandatory if listIds is empty**.
notifyUrl	string	☐		URL that w ill be called once the import process is finished. For reference, https://help.brevo.com/hc/en-us/articles/360007666479
smsBlacklist	boolean	☐	False	To blacklist all the contacts for sms
updateExistingContacts	boolean	☐	True	To facilitate the choice to update the existing contacts

6.1.7 sib_contacts.removecontactfromlist: Sendinblue Package for Contacts operations that could not be mapped to a table.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.removecontactfromlist. A value must be provided at all times for required parameters,

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_listId	int64	☑		Id of the list

6.1.8 sib_contacts.requestcontactexport: Sendinblue Package for Contacts operations that could not be mapped to a table.

It returns the background process ID which on completion calls the notify URL that you have set in the input. File will be available in csv.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.requestcontactexport. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
actionForContacts	string	☐		**Mandatory if neither actionForEmailCampaigns nor actionForSmsCampaigns is passed.** This w ill export the contacts on the basis of provided action applied on contacts as per the list id. * **allContacts** - Fetch the list of all contacts for a particular list. * **subscribed & unsubscribed** - Fetch the list of subscribed / unsubscribed (blacklisted via any means) contacts for a particular list. * **unsubscribedPerList** - Fetch the list of contacts that are

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				unsubscribed from a particular list only.
actionForEmailCampaigns	string	☐		**Mandatory if neither actionForContacts nor actionForSmsCampaigns is passed.** This will export the contacts on the basis of provided action applied on email campaigns. * **openers & nonOpeners** - emailCampaignId is mandatory. Fetch the list of readers / non-readers for a particular email campaign. * **clickers & nonClickers** - emailCampaignId is mandatory. Fetch the list of clickers / non-clickers for a particular email campaign. * **unsubscribed** - emailCampaignId is mandatory. Fetch the list of all unsubscribed (blacklisted via any means) contacts for a particular email campaign. * **hardBounces & softBounces** - emailCampaignId is optional. Fetch the list of hard bounces / soft bounces for a particular / all email campaign(s).
actionForSmsCampaigns	string	☐		**Mandatory if neither actionForContacts nor actionForEmailCampaigns is passed.** This will export the contacts on the basis of provided action applied on sms campaigns. * **unsubscribed** - Fetch the list of all unsubscribed (blacklisted via any means) contacts for all / particular sms campaigns. * **hardBounces & softBounces** - Fetch the list of hard bounces / soft bounces for all / particular sms campaigns.
emailCampaignId	int64	☐		Considered only if **actionForEmailCampaigns** is passed, ignored otherwise. **Mandatory if action is one of the following - openers, nonOpeners, clickers, nonClickers, unsubscribed.** The id of the email campaign for which the corresponding action shall be applied in the filter.
listId	int64	☐		ID of the list. This is mandatory if actionForContacts is specified and segmentId is not provided. Either segmentId or listId must be included.
segmentId	int64	☐		ID of the segment. This is mandatory if actionForContacts is specified and listId is not provided. Either segmentId or listId must be included.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
smsCampaignId	int64	☐		Considered only if **actionForSmsCampaigns** is passed, ignored otherwise. The id of sms campaign for which the corresponding action shall be applied in the filter.
disableNotification	boolean	☐	False	To avoid generating the email notification upon contact export, pass **true**
exportMandatoryAttributes	boolean	☐	True	To export mandatory attributes like EMAIL, ADDED_TIME, MODIFIED_TIME
exportDateInUTC	boolean	☐	False	Specifies whether the date fields createdAt, modifiedAt in the exported data should be returned in UTC format.
notifyUrl	string	☐		Webhook that will be called once the export process is finished. For reference, https://help.brevo.com/hc/en-us/articles/360007666479

6.1.9 sib_contacts.updateattribute: Sendinblue Package for Contacts operations that could not be mapped to a table.

Update an existing contact attribute identified by its category and name. For category-type attributes, you can update the enumeration values; for calculated or global attributes, update the computed value formula; and for normal multiple-choice attributes, update the multicategory options. None of the category or multicategory option values can exceed 200 characters.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_contacts.updateattribute. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_attributeCategory	string	☑		Category of the attribute (Values: category, calculated, global, normal)
p_attributeName	string	☑		Name of the existing attribute

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
value	string	☐		Value of the attribute to update. **Use only if the attribute's category is 'calculated' or 'global'**

6.1.10 sib_contacts.updatebatchcontacts: Sendinblue Package for Contacts operations that could not be mapped to a table.

Update multiple contacts in a single API call by passing an array of contact objects. Each contact in the array must be identified by one of: email, id, or sms (only one identifier per contact). You can update attributes, blacklist status, list memberships, ext_id, and transactional email forbidden senders for each contact in the batch.

7 Package: sib_coupons

7.1 Procedures

7.1.1 sib_coupons.createcoupons: Sendinblue Package for Coupons operations that could not be mapped to a table.

Add coupons to an existing coupon collection. The `coupons` array must contain between 1 and 10,000 unique coupon code strings, all associated with the specified `collectionId`. Coupon creation is processed asynchronously and a `204` status is returned immediately upon acceptance. Returns a `404` error if the specified coupon collection does not exist.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_coupons.createcoupons. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
collectionId	guid	☑		The id of the coupon collection for which the coupons will be created

7.1.2 sib_coupons.updatecouponcollection: Sendinblue Package for Coupons operations that could not be mapped to a table.

Update an existing coupon collection by its UUID. You can modify the default coupon value, set or remove the expiration date (pass `null` to remove), and configure or disable alert thresholds for remaining coupons or remaining days. Only the fields included in the request body are updated; omitted fields remain unchanged.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_coupons.updatecouponcollection. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_id	guid	☑		Id of the collection to update
defaultCoupon	string	☐		A default coupon to be used in case there are no coupons left
expirationDate	datetime	☐		Specify an expiration date for the coupon collection in RFC3339 format. Use null to remove the expiration date.
remainingCouponsAlert	int64	☐		Send a notification alert (email) when the remaining coupons count is equal or fall bellow this number. Use null to disable alerts.
remainingDaysAlert	int64	☐		Send a notification alert (email) when the remaining days until the expiration date are equal or fall bellow this number. Use null to disable alerts.

8 Package: sib_custom_objects

8.1 Procedures

8.1.1 sib_custom_objects.batchdeleteobjectrecords: Sendinblue Package for Custom Objects operations that could not be mapped to a table.

Use this endpoint to delete multiple object records of the same object-type in one request.The request is accepted and processed asynchronously. You can track the status of the deletion process using the returned **processId**.**Limitations:** - Each request can contain up to **1000** object record identifiers - Either `ids` or `ext_ids` must be provided, but **not both** in the same request - Deletion of Brevo standard object records is not supported via this endpoint - If more records must be deleted, send multiple batch requests

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_custom_objects.batchdeleteobjectrecords. 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-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_object_type	string	☑		Object type for the records to delete
identifiers	string	☐		Either `ids` or `ext_ids` must be provided, but not both in the same request.

8.1.2 sib_custom_objects.upsertrecords: Sendinblue Package for Custom Objects operations that could not be mapped to a table.

<Note title="Enterprise access only">Custom objects are only available to Enterprise plans.This feature is in beta. These are subject to change.</Note>Performs bulk create or update (upsert) operations for object records in a single asynchronous request. This endpoint is optimized for high-volume data imports and synchronization scenarios.**How Upsert Works:**- **Create**: Omit `identifiers`, or provide only `ext_id` (if it doesn't already exist). A new record is created with a Brevo-generated `id`.- **Update**: Provide `id` (Brevo internal ID) or an `ext_id` that already exists. The matching record is updated with the new attribute values.- **Important:** `id` is for **updates only**. Providing an `id` that does not belong to an existing record will fail during async processing (the HTTP response will still be 202, but the record will be rejected in the background). To create a new record with a stable external reference, use `ext_id` instead.**Request Structure:**Each object record in the `records` array can include:- `identifiers`: Either `id` (internal Brevo ID) or `ext_id` (your external system ID) — required for updates. **Note:** use `id` (singular), not `ids`.- `attributes`: Key-value pairs where each key is the attribute **key** (e.g., `company_name`), not the attribute label (e.g., "Company Name").- `associations`: Controls linking and unlinking of associated records (optional). Each entry specifies:- `object_type`: The type of the associated object- `action`: `link` (default) to create the association, or `unlink` to remove it- `records`: The associated records to link or unlink (each identified by `ext_id` or `id`)- **Unlink is idempotent** — unlinking a non-existing association is a no-op (no error returned)- `link` and `unlink` actions can be submitted for the same `object_type` in a single record entry- Both associated records must already exist before a link can be created> **Common mistake:** Passing the attribute **label** (the display name you see in the UI) instead of the attribute **key** will cause the attribute to be silently ignored and the record may not be created as expected.**Asynchronous Processing:**- Returns immediately with a `processId` (HTTP 202 Accepted)- Use the processId to track status via the Get process API**API and Schema Limitations:**- Max 1000 object records per request- Max request body size: 1 MB- Max 500 attributes per object record (matches the schema limit of 500 attributes per object)- Unknown attribute keys are silently ignored (no error, no attribute creation)- Max 10 association records per associated object-type in each record of the request. If you need more, send multiple requests.**Important Behaviors:**- The object schema must be created before upserting records- Unknown attribute keys are silently ignored (no error, no creation)- Both associated object records must already

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

exist before creating a link association- Unlink operations are idempotent: attempting to unlink a non-existing association returns success- `link` and `unlink` actions can be submitted for the same `object_type` in a single record entry- Contact objects cannot be created via this endpoint- For `category` and `multiple_category` attributes, pass the option **key** as the value (not the option label or option ID).- The `id` identifier (internal Brevo ID) can only be used for **updating** existing records. To create new records, either omit identifiers (Brevo auto-generates an ID) or provide an `ext_id`.**Errors:**- Make sure both object records exist before associating them, else the API will return an error.- This route does not create objects. The object where the object records are upserted by this API must be created already else the API will return an error "invalid object type".

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_custom_objects.upsertrecords. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_object_type	string	☑		Object type for the records to upsert. Must be a previously created custom object type. Only lowercase alphanumeric characters and underscores are allowed (max 32 characters).

9 Package: sib_domains

9.1 Procedures

9.1.1 sib_domains.authenticatedomain: Sendinblue Package for Domains operations that could not be mapped to a table.

Authenticates a specific domain.Use this to:- Authenticate a domain- Verify DNS record configuration- Complete domain setup for sending- Enable domain for email authenticationKey information returned:- Authentication success confirmation- Domain readiness status for email sending

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_domains.authenticatedomain. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_domainName	string	☑		Domain name

9.1.2 sib_domains.deletedomain: Sendinblue Package for Domains operations that could not be mapped to a table.

Deletes a domain from Brevo.Use this to:- Remove existing domains- Clean up unused domain configurations- Remove test domainsKey information returned:- Success confirmation message

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_domains.deletedomain. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_domainName	string	☑		Domain name

10 Package: sib_ecommerce

10.1 Procedures

10.1.1 sib_ecommerce.createbatchorder: Sendinblue Package for Ecommerce operations that could not be mapped to a table.

Create multiple orders at one time instead of one order at a time

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_ecommerce.createbatchorder. 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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
historical	boolean	☐	True	Defines whether you want your orders to be considered as live data or as historical data (import of past data, synchronising data). True: orders will not trigger any automation workflows. False: orders will trigger workflows as usual.
notifyUrl	string	☐		Webhook URL to receive the status of the batch request

10.1.2 sib_ecommerce.createorder: Sendinblue Package for Ecommerce operations that could not be mapped to a table.

Manages the transactional status of the order

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_ecommerce.createorder. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
amount	decimal	☑		Total amount of the order, including all shipping expenses, tax and the price of items.
address	string	☐		Full billing address.
city	string	☐		Exact city of the address.
country	string	☐		Billing country name.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
countryCode	string	☐		Billing country 2-letter ISO code.
paymentMethod	string	☐		How the visitor will pay for the item(s), e.g. paypal, check, etc.
phone	string	☐		Billing phone number.
postCode	string	☐		Postcode for delivery and billing.
region	string	☐		Exact region (state/province) for delivery and billing.
createdAt	string	☑		Event occurrence UTC date-time (YYYY-MM-DDTHH:mm:ssZ), when order is actually created.
id	string	☑		Unique ID of the order.
email_id	string	☐		Email of the contact associated with the order
ext_id	string	☐		ext_id associated with the order
loyalty_subscription_id	string	☐		loyalty_subscription_id associated with the order
phone_id	string	☐		Phone number of the contact associated with the order
value	string	☐		
status	string	☑		State of the order.
storeId	string	☐		ID of store where the order is placed
updatedAt	string	☑		Event updated UTC date-time (YYYY-MM-DDTHH:mm:ssZ), when the status of the order is actually changed/updated.

10.1.3 sib_ecommerce.createproductalert: Sendinblue Package for Ecommerce operations that could not be mapped to a table.

Register a contact to receive an alert for a specific product event, such as `back_in_stock`. At least one contact identifier (`ext_id`, `email`, or `sms`) must be provided; when multiple are given, priority is `ext_id` > `email` > `sms`. Returns a `404` error if the product ID does not exist, and a `403` error if product alerts are not enabled for the account.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_ecommerce.createproductalert. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same pro-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

cedure 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
p_id	string	☑		Product ID
p_type	string	☑		Alert type (Values: back-_in_stock)
email	string	☐		Email address of the contact
ext_id	string	☐		Contact ID in your system
sms	string	☐		Sms of the contact

10.1.4 sib_ecommerce.createupdatebatchcategory: Sendinblue Package for Ecommerce operations that could not be mapped to a table.

Create or update multiple ecommerce categories in a single request. The `categories` array accepts up to 100 category objects, each requiring a unique `id`. When `updateEnabled` is `false` (the default), all categories are inserted as new; if any ID already exists, a `400` error is returned. When `updateEnabled` is `true`, existing categories are updated and new ones are created via upsert. Duplicate IDs within the same request payload are rejected. The response returns the count of created and updated categories.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_ecommerce.createupdatebatchcategory. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
updateEnabled	boolean	☐		Facilitate to update the existing categories in the same request (updateEnabled = true)

10.1.5 sib_ecommerce.createupdatebatchproducts: Sendinblue Package for Ecommerce operations that could not be mapped to a table.

Create or update multiple ecommerce products in a single request. The `products` array accepts up to 100 product objects for creation (or up to 1000 when `updateEnabled` is `true` and the account has an increased limit). Each product requires a unique `id` and `name` (name is mandatory for creation only). When `updateEnabled` is `false`, all products are inserted as new; if any ID already exists, a `400` error is returned. When `updateEnabled` is `true`, existing products are updated and new ones are created via upsert. Duplicate IDs

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

within the same request payload are rejected. The response returns the count of created and updated products.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_ecommerce.createupdatebatchproducts. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
updateEnabled	boolean	☐		Facilitate to update the existing products in the same request (updateEnabled = true)

11 Package: sib_email_campaigns

11.1 Procedures

11.1.1 sib_email_campaigns.emailexportrecipients: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

Export the recipients of a sent email campaign as an asynchronous process, filtered by recipient type (e.g. openers, clickers, hardBounces). The recipientsType field is required and determines which subset of recipients to export. An optional notifyURL webhook will be called once the export is complete, and the response returns a processId to track the export status.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_email_campaigns.emailexportrecipients. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_campaignId	int64	☑		Id of the campaign
notifyURL	string	☐		Webhook called once the export process is finished. For reference, https://help.brevo.com/hc/en-us/articles/360007666479
recipientsType	string	☑		Type of recipients to export for a campaign (Values: all, nonClickers, nonOpeners, clickers, openers, softBounces, hardBounces, unsubscribed)

11.1.2 sib_email_campaigns.sendemailcampaignnow: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

Send an existing email campaign immediately by scheduling it for the current time. The campaign must have valid recipients and content configured before sending. The system verifies your account''s send limit and credit balance before dispatching; if credits are insufficient, a 402 error is returned.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_email_campaigns.sendemailcampaignnow. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		Id of the campaign

11.1.3 sib_email_campaigns.sendreport: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

A PDF will be sent to the specified email addresses

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_email_campaigns.sendreport. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		Id of the campaign
body	string	☐		Custom text message to be presented in the report email.
to	string	☐		Email addresses of the recipients
language	string	☐	fr	Language of email content for campaign report sending. (Values: fr, es, pt, it, de, en)

11.1.4 sib_email_campaigns.sendtestemail: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

Send a test version of an email campaign to specified email addresses or your entire test list. If the emailTo array is left empty, the test mail will be sent to all addresses in your test list. You can send a maximum of 50 test emails per day.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_email_campaigns.sendtestemail. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		Id of the campaign

11.1.5 sib_email_campaigns.updatecampaignstatus: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

Update the status of an email campaign, such as suspending, archiving, or replicating it. Available status values include suspended, archive, darchive, sent, queued, replicate, replic-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ateTemplate, cancel, and draft. Note that the replicateTemplate status is only available for template type campaigns.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_e-mail_campaigns.updatecampaignstatus. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		Id of the campaign
status	string	☐		Note:- **replicateTemplate** status w ill be available **only for template type campaigns.**(Values: suspended, archive, darchive, sent, queued, replicate, replicateTemplate, cancel, draft)

11.1.6 sib_email_campaigns.uploadimagetogallery: Sendinblue Package for Email Campaigns operations that could not be mapped to a table.

Upload an image to your account''s image gallery by providing an absolute URL to the image. The maximum allowed image size is 2MB and supported formats are jpeg, jpg, png, bmp, and gif; local file uploads are not supported.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_e-mail_campaigns.uploadimagetogallery. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
imageUrl	string	☑		The absolute url of the image (**no local file**). Maximum allow ed size for image is **2MB**. Allow ed extensions for images are: #### jpeg, jpg, png, bmp, gif.
name	string	☐		Name of the image.

12 Package: sib_event

12.1 Procedures

12.1.1 sib_event.createbatchevents: Sendinblue Package for Event operations that could not be mapped to a table.

Create multiple events to track contacts' interactions in a single request.

13 Package: sib_master_account

13.1 Procedures

13.1.1 sib_master_account.inviteadminuser: Sendinblue Package for Master account operations that could not be mapped to a table.

`This endpoint allows you to invite a member to manage the Admin accountFeatures and their respective permissions are as below: - `my_plan`:- "all"- `api`:- "none"- `user_management`:- "all"- `app_management` | Not available in ENTv2:- "all"- `sub_organization_groups`- "create"- "edit_delete"- `create_sub_organizations`- "all"- `manage_sub_organizations`- "all"- `analytics`- "download_data"- "create_alerts"- "my_looks"- "explore_create"- `security`- "all"**Note**: - If `all_features_access: false` then only privileges arerequired otherwise if `true` then it's assumed that all permissions will bethere for the invited admin user.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_master_account.inviteadminuser. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
all_features_access	boolean	☑		All access to the features (Values: true, false)
email	string	☑		Email address for the organization

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

14 Package: sib_payments

14.1 Procedures

14.1.1 sib_payments.createpaymentrequest: Sendinblue Package for Payments operations that could not be mapped to a table.

Create a new payment request for a Brevo contact. The request requires a reference (displayed on the payment page), a contact ID, and a cart with currency and amount in cents. You can optionally configure a custom success redirect URL and enable email notifications with reminders. Returns the payment request ID and its public payment URL. A `403` error is returned if Brevo Payments is not activated or the account is not validated.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_payments.createpaymentrequest. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
currency	string	☐		Currency code for the payment amount.
specificAmount	int64	☐		Payment amount, in cents. e.g. if you want to request €12.00, then the amount in cents is 1200.
configuration	string	☐		
contactId	int64	☑		Brevo ID of the contact requested to pay.
description	string	☐		Description of payment request.
channel	string	☐		Channel used to send the notifications.
text	string	☐		Use this field if you want to give more context to your contact about the payment request.
reference	string	☑		Reference of the payment request, it will appear on the payment page.

14.1.2 sib_payments.deletepaymentrequest: Sendinblue Package for Payments operations that could not be mapped to a table.

Delete a payment request by its UUID. Once deleted, the payment request can no longer be accessed or paid. Returns a `404` error if no payment request matches the provided ID, and a `403` error if Brevo Payments is not activated or the account is not validated.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_payments.deletepaymentrequest. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_id	guid	☑		ID of the payment request.

15 Package: sib_program

15.1 Procedures

15.1.1 sib_program.deletecontactmembers: Sendinblue Package for Program operations that could not be mapped to a table.

Deletes member from a subscription

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_program.deletecontactmembers. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID. A unique identifier for the loyalty program.
p_memberContactIds	string	☑		Comma-separated list of member contact IDs to delete from the subscription.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

15.1.2 sib_program.deletecontactsubscription: Sendinblue Package for Program operations that could not be mapped to a table.

Delete subscription for a contact

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_program.deletecontactsubscription. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID. A unique identifier for the loyalty program.
p_cid	int64	☑		Contact ID.

15.1.3 sib_program.partiallyupdateloyaltyprogram: Sendinblue Package for Program operations that could not be mapped to a table.

Partially updates loyalty program

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_program.partiallyupdateloyaltyprogram. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID. A unique identifier for the loyalty program.
description	string	☐		Loyalty Program description

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
name	string	☐		Loyalty Program name

15.1.4 sib_program.publishloyaltyprogram: Sendinblue Package for Program operations that could not be mapped to a table.

Publishes loyalty program

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_program.publishloyaltyprogram. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID. A unique identifier for the loyalty program.

15.1.5 sib_program.subscribemembertoasubscription: Sendinblue Package for Program operations that could not be mapped to a table.

Add member to a subscription

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_program.subscribemembertoasubscription. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_pid	guid	☑		Loyalty Program ID. A unique identifier for the loyalty program.
contactId	int64	☐		Required if LoyaltySubscriptionId is not provided, must be greater than 0
loyaltySubscriptionId	string	☐		Required if ContactId is not provided, max length 64

15.1.6 sib_program.subscribetoloyaltyprogram: Sendinblue Package for Program operations that could not be mapped to a table.

Subscribes to a loyalty program

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_program.subscribetoloyaltyprogram. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID. A unique identifier for the loyalty program.
contactId	int64	☑		Required contact ID; must be greater than 0.
loyaltySubscriptionId	string	☐		Optional subscription ID (max length 64).
creationDate	datetime	☐		Optional creation date in ISO 8601 format (YYYY-MM-DDThh:mm:ss.ffffff+HH:MM). Must be in the past.

16 Package: sib_reward

16.1 Procedures

16.1.1 sib_reward.completeredeemtransaction: Sendinblue Package for Reward operations that could not be mapped to a table.

Completes voucher redeem request.

Parameters of Procedure

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the procedure sib_reward.completeredeemtransaction. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_tid	string	☑		Redeem transaction ID

16.1.2 sib_reward.createreward: Sendinblue Package for Reward operations that could not be mapped to a table.

Creates a new reward in the loyalty program.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_reward.createreward. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
name	string	☑		Internal name of the reward
publicDescription	string	☐		Public facing description of the reward
publicImage	string	☐		URL of the public image for the reward
publicName	string	☐		Public facing name of the reward

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

16.1.3 sib_reward.createvoucher: Sendinblue Package for Reward operations that could not be mapped to a table.

Create a voucher and attribute it to a specific membership.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_reward.createvoucher. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
code	string	☐		Code generated to attribute reward to a contact
contactId	int64	☐		Contact to attribute the reward
expirationDate	string	☐		Reward expiration date
loyaltySubscriptionId	string	☐		One of contactId or loyaltySubscriptionId is required
rewardId	guid	☑		Reward id
value	double	☐		Value of the selected reward config
validFrom	string	☐		Date from which the voucher becomes valid. Accepts RFC 3339 or DD/MM/YYYY HH:MM AM/PM format. Converted to UTC using the organization's timezone.

16.1.4 sib_reward.redeemvoucher: Sendinblue Package for Reward operations that could not be mapped to a table.

Creates a request to redeem a voucher.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_reward.redeemvoucher. 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 `begin package.procedure(value1, value2, value3); end` on a proced-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
attributedRewardId	guid	☐		Unique identifier for the attributed reward
code	string	☐		Redemption code for the reward
contactId	int64	☐		Unique identifier for the contact
loyaltySubscriptionId	string	☐		Identifier for the loyalty subscription
order	string	☐		Order details for the redemption
rewardId	guid	☐		Unique identifier for the reward
ttl	int64	☐		Time to live in seconds for the redemption request
autoComplete	boolean	☐		Whether the redemption should be automatically completed

16.1.5 sib_reward.revokevouchers: Sendinblue Package for Reward operations that could not be mapped to a table.

Revoke attributed vouchers.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_reward.revokevouchers. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_attributedRewardIds	string	☐		Reward Attribution IDs (comma seperated)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

16.1.6 sib_reward.validatereward: Sendinblue Package for Reward operations that could not be mapped to a table.

Validates a reward.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_reward.validatereward. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
attributedRewardId	guid	☐		Unique identifier for the attributed reward
code	string	☐		Validation code for the reward
contactId	int64	☐		Unique identifier for the contact
loyaltySubscriptionId	string	☐		Identifier for the loyalty subscription
pointOfSellId	string	☐		Identifier for the point of sale
rewardId	guid	☐		Unique identifier for the reward

17 Package: sib_sales

17.1 Procedures

17.1.1 sib_sales.insert_note: Sendinblue Brevo sales

Insert a note and link it to contacts, deals or companies.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_sales.insert_note. 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 `begin package.procedure(value1, value2, value3); end` on a procedure with four parameters will use the default value for the fourth parameter and the specified values for the first three.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by name is done by listing all values that require a value. For example with `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
text	string	☑		Text
contactIds	object	☐		Contact IDs
dealIds	object	☐		Deal IDs
companyIds	object	☐		Company IDs

18 Package: sib_senders

18.1 Procedures

18.1.1 sib_senders.validatesenderbyotp: Sendinblue Package for Senders operations that could not be mapped to a table.

Validates a sender using the OTP (One-Time Password) received via email.Use this to:- Complete sender verification process- Activate a newly created sender- Verify ownership of the sender email address- Enable the sender for use in email campaignsKey information returned:- Success confirmation of sender verification- Sender activation status

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_senders.validatesenderbyotp. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_senderId	int64	☑		Id of the sender
otp	int64	☑		6 digit OTP received on email

19 Package: sib_sms_campaigns

19.1 Procedures

19.1.1 sib_sms_campaigns.requestsmsrecipientexport: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.

It returns the background process ID which on completion calls the notify URL that you have set in the input.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_sms_campaigns.requestsmsrecipientexport. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		id of the campaign
notifyURL	string	☐		URL that will be called once the export process is finished. For reference, https://help.brevo.com/hc/en-us/articles/360007666479
recipientsType	string	☑		Filter the recipients based on how they interacted with the campaign (Values: all, delivered, answered, softBounces, hardBounces, unsubscribed)

19.1.2 sib_sms_campaigns.sendsmscampaignnow: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.

Send an existing SMS campaign immediately by scheduling it for the current time. The system verifies your account''s SMS credit balance before dispatching; if credits are insufficient or the remaining credit is less than the number of recipients, a 402 error is returned. The campaign must have valid recipients and content already configured.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_sms_campaigns.sendsmscampaignnow. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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

19.1.3 sib_sms_campaigns.sendsmsreport: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.

Send report of Sent and Archived campaign, to the specified email addresses, with respective data and a pdf attachment in detail.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_sms_campaigns.sendsmsreport. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		id of the campaign
body	string	☐		Custom text message to be presented in the report email.
to	string	☐		Email addresses of the recipients
language	string	☐	fr	Language of email content for campaign report sending. (Values: fr, es, pt, it, de, en)

19.1.4 sib_sms_campaigns.sendtestsms: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.

Send a test SMS to a specified phone number to preview the campaign before sending it to all recipients. The phone number must belong to one of your existing contacts in your Brevo account and must not be blacklisted. The number should include the country code (e.g. 33689965433).

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_sms_campaigns.sendtestsms. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		Id of the SMS campaign
phoneNumber	string	☐		Mobile number of the recipient with the country code. This number **must belong to one of your contacts in Brevo account and must not be blacklisted**

19.1.5 sib_sms_campaigns.updatesmscampaignstatus: Sendinblue Package for SMS Campaigns operations that could not be mapped to a table.

Update the status of an SMS campaign, such as suspending, archiving, or replicating it. Available status values include suspended, archive, darchive, sent, queued, replicate, replicateTemplate, cancel, and draft. Note that the replicateTemplate status is only available for template type campaigns.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_sms_campaigns.updatesmscampaignstatus. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_campaignId	int64	☑		id of the campaign
status	string	☐		Note:- **replicateTemplate** status will be available **only for template type campaigns.**(Values: suspended, archive, darchive, sent, queued, replicate, replicateTemplate, cancel, draft)

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

20 Package: sib_tier

20.1 Procedures

20.1.1 sib_tier.addsubscriptiontotier: Sendinblue Package for Tier operations that could not be mapped to a table.

Manually assigns a tier to a specific membership.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tier.addsubscriptiontotier. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_cid	guid	☑		Contact ID
p_tid	guid	☑		Tier ID

20.1.2 sib_tier.createtierfortiergroup: Sendinblue Package for Tier operations that could not be mapped to a table.

Creates a new tier in a loyalty program tier group. *(The changes will take effect with the next publication of the loyalty program)*

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tier.createtierfortiergroup. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_pid	guid	☑		Loyalty Program ID
p_gid	guid	☑		Tier group ID
imageRef	string	☐		Image of the tier
name	string	☑		Name of the tier to be created

20.1.3 sib_tier.createtiergroup: Sendinblue Package for Tier operations that could not be mapped to a table.

Creates a new tier group in a loyalty program. *(The changes will take effect with the next publication of the loyalty program)*

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tier.createtiergroup. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
dow ngradeStrategy	string	☐	real_time	Select real_time to dow ngrade tier on real time balance updates. Select membership_anniversary to dow ngrade tier on subscription anniversary. Select tier_anniversary to dow ngrade tier on tier anniversary. (Values: real_time, membership_anniversary, tier_anniversary)
name	string	☑		Name of the tier group
isInternal	boolean	☐		Indicates w hether the tier group is internal.
upgradeStrategy	string	☐	real_time	Select real_time to upgrade tier on real time balance updates. Select membership_anniversary to upgrade tier on subscription anniversary. Select tier_anniversary to upgrade tier on tier anniversary. (Values: real_time, membership_anniversary, tier_anniversary)
durationValue	int64	☐		Duration value for the schedule.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
durationUnit	string	☐		Duration unit for the schedule.
durationModifier	string	☐		Modifier for the duration.
scheduledDate	string	☐		Scheduled date in DD/MM format.
durationValue_2	int64	☐		Duration value for the schedule.
durationUnit_2	string	☐		Duration unit for the schedule.
durationModifier_2	string	☐		Modifier for the duration.
scheduledDate_2	string	☐		Scheduled date in DD/MM format.

20.1.4 sib_tier.deletetier: Sendinblue Package for Tier operations that could not be mapped to a table.

Deletes a tier from a loyalty program tier group. *(The changes will take effect with the next publication of the loyalty program)*

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tier.deletetier. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_tid	guid	☑		Tier ID

20.1.5 sib_tier.deletetiergroup: Sendinblue Package for Tier operations that could not be mapped to a table.

Deletes a tier group from a loyalty program. *(The changes will take effect with the next publication of the loyalty program)*

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tier.deletetiergroup. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_gid	guid	☑		Tier group ID

20.1.6 sib_tier.updatetier: Sendinblue Package for Tier operations that could not be mapped to a table.

Modifies an existing tier for the specified tier group *(The changes will take effect with the next publication of the loyalty program)*

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tier.updatetier. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_tid	guid	☑		Tier ID
imageRef	string	☐		Image of the tier
name	string	☑		Name of the tier to be created

20.1.7 sib_tier.updatetiergroup: Sendinblue Package for Tier operations that could not be mapped to a table.

Updates a tier group from a loyalty program. *(The changes will take effect with the next publication of the loyalty program)*

Parameters of Procedure

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The following parameters can be used to control the behaviour of the procedure sib_tier.updatetiergroup. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_pid	guid	☑		Loyalty Program ID
p_gid	guid	☑		Tier group ID
dow ngradeStrategy	string	☑	real_time	Select real_time to dow ngrade tier on real time balance updates. Select membership_anniversary to dow ngrade tier on subscription anniversary. Select tier_anniversary to dow ngrade tier on tier anniversary. (Values: real_time, membership_anniversary, tier_anniversary)
name	string	☑		Name of the tier group
isInternal	boolean	☐		Indicates w hether the tier group is internal.
upgradeStrategy	string	☑	real_time	Select real_time to upgrade tier on real time balance updates. Select membership_anniversary to upgrade tier on subscription anniversary. Select tier_anniversary to upgrade tier on tier anniversary. (Values: real_time, membership_anniversary, tier_anniversary)

21 Package: sib_tracker

21.1 Procedures

21.1.1 sib_tracker.identify: Sendinblue Brevo tracker

The Identify endpoint lets you create or update users on the Marketing Automation platform. You can also define extra attributes for such users.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tracker.identify. 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.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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 of an existing contact or contact to be created.
attributes	string	☐		JSON object specifying key/value pairs for contact fields.

21.1.2 sib_tracker.track_event: Sendinblue Brevo tracker

The Track Event endpoint lets you capture events on your website, along with optional extra information about the event.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tracker.track_event. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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 of an existing contact or contact to be created.
event	string	☑		Short name of the event that occurred.
properties	string	☐		JSON object specifying key/value pairs for contact fields.
eventData	string	☐		JSON object specifying additional event data as key/value pairs.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

21.1.3 sib_tracker.track_link: Sendinblue Brevo tracker

The Track Link endpoint lets you capture clicks on your website, along with optional extra information about the click.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tracker.track_link. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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 of an existing contact or contact to be created.
link	string	☑		URL for the link clicked upon by the user.
properties	string	☐		JSON object specifying key/value pairs for contact fields.

21.1.4 sib_tracker.track_page: Sendinblue Brevo tracker

The Track Page endpoint lets you capture page views on your website, along with optional extra information about the viewed page.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_tracker.track_page. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Email address of an existing contact or contact to be created.
page	string	☑		URL for the w ebpage visited by the user.
properties	string	☐		JSON object specifying key/value pairs for contact fields.

22 Package: sib_transactional_emails

22.1 Procedures

22.1.1 sib_transactional_emails.deleteblockeddomain: Sendinblue Package for Transactional emails operations that could not be mapped to a table.

Unblocks an existing domain from the list of blocked domains

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_transactional_emails.deleteblockeddomain. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_domain	string	☑		The name of the domain to be deleted

22.1.2 sib_transactional_emails.deletehardbounces: Sendinblue Package for Transactional emails operations that could not be mapped to a table.

Delete hardbounces. To use carefully (e.g. in case of temporary ISP failures)

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_transactional_emails.deletehardbounces. 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 `begin package.procedure(value1, value2, value3); end` on a proced-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
contactEmail	string	☐		Target a specific email address
endDate	string	☐		Ending date (YYYY-MM-DD) of the time period for deletion. The hardbounces until this date w ill be deleted. Must be greater than or equal to the startDate
startDate	string	☐		Starting date (YYYY-MM-DD) of the time period for deletion. The hardbounces occurred after this date w ill be deleted. Must be less than or equal to the endDate

22.1.3 sib_transactional_emails.deletescheduledemailbyid: Sendinblue Package for Transactional emails operations that could not be mapped to a table.

Delete scheduled batch of emails by batchId or single scheduled email by messageId

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_transactional_emails.deletescheduledemailbyid. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_identifier	string	☑		The `batchId` of scheduled emails batch (must be a valid UUIDv4) or the `messageId` of scheduled email (enclosed in angle brackets w ith @ sign, e.g. `<...@domain>`).

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

22.1.4 sib_transactional_emails.postpreviewsmtpemailtemplates: Sendinblue Package for Transactional emails operations that could not be mapped to a table.

Generate a fully rendered preview of a transactional email template by resolving dynamic variables. Provide either an `email` address (to populate variables from the contact''s attributes) or a `params` object with key-value pairs for manual substitution; at least one of these is required alongside the mandatory `templateId`. The response includes the rendered HTML, subject, sender details, preview text, and any feed names used in the template.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_transactional_emails.postpreviewsmtpemailtemplates. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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 of the contact.(Required if params not provided)
Firstname	string	☐		firstname of the contact. **Dynamic parameter**
Lastname	string	☐		Lastname of the contact. **Dynamic parameter**
templateId	int64	☑		Id of the template.

22.1.5 sib_transactional_emails.sendtesttemplate: Sendinblue Package for Transactional emails operations that could not be mapped to a table.

Send a test email of the specified transactional template to one or more recipients. Provide an array of email addresses in the `emailTo` field; if left empty, the test mail is sent to your entire test list. You can send a maximum of 50 test emails per day, and all provided email addresses must be valid.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_transactional_emails.sendtesttemplate. 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 `begin package.procedure(value1, value2, value3); end` on a proced-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_templateId	int64	☑		Id of the template

22.1.6 sib_transactional_emails.sendtransacemail: Sendinblue Package for Transactional emails operations that could not be mapped to a table.

Send a transactional email to one or more recipients, either using inline HTML content or a pre-built template via `templateId`. You can schedule emails for future delivery using `scheduledAt` (UTC, up to 5-minute delay), send multiple personalized versions with `messageVersions` (max 2000 total recipients, 99 per version), and attach files via URL or base64-encoded content. A `sender` and `subject` are required when no `templateId` is provided; when a `templateId` is used, the template''s sender and subject are applied unless overridden.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_transactional_emails.sendtransacemail. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
batchId	string	☐		UUIDv4 identifier for the scheduled batch of transactional emails. If omitted, a valid UUIDv4 batch identifier is automatically generated.
htmlContent	string	☐		HTML body content of the email. Required when `templateId` is not provided. Ignored when `templateId` is provided.
email	string	☐		Email address in reply to
name	string	☐		Display name for the reply-to address. Maximum length is 70 characters.
scheduledAt	datetime	☐		UTC date-time when the email should be sent (format: YYYY-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				MM-DDTHH:mm:ss.SSSZ). Include timezone information in the date-time value. Scheduled emails may be delayed by up to 5 minutes.
email_2	string	☐		Sender email address. Required when `id` is not provided.
id	int64	☐		Sender identifier. Required when `email` is not provided. Use this field to select a sender with a specific IP pool (dedicated IP users only).
name_2	string	☐		Display name of the sender. Maximum length is 70 characters. Only applicable when `email` is provided.
subject	string	☐		Email subject line. Required when `templateId` is not provided.
templateId	int64	☐		Template identifier
textContent	string	☐		Plain text body content of the email. Ignored when `templateId` is provided.

23 Package: sib_transactional_sms

23.1 Procedures

23.1.1 sib_transactional_sms.sendasynctransactionalsms: Sendinblue Package for Transactional SMS operations that could not be mapped to a table.

<Note>If the user includes stop code in the Transactional SMS, then it will be switched to Marketing SMS automatically and it will be interpreted as a Marketing SMS. To send Transactional SMS as Transactional, it is important not to use stop code.Note: For adding a stop code, client has to add reply STOP to [STOP_CODE] and the [STOP_CODE] will be replaced with the number.</Note><Note title="For end users in France">Transactional SMS can be sent at any time without time restrictions. However, if a message is categorized as Marketing, it must adhere to specific time restrictions. Messages sent outside of these restricted hours will experience delays and will be processed during allowable times. Specifically, Marketing SMS cannot be processed between 10pm and 8am, on Sundays, and on French public holidays.</Note>

23.1.2 sib_transactional_sms.sendtransacsms: Sendinblue Package for Transactional SMS operations that could not be mapped to a table.

Send a transactional SMS message to a single mobile number. The `sender`, `recipient`, and either `content` or `templateId` fields are required. The sender name is limited to 11 alphanumeric characters or 15 numeric characters, and the recipient must be a valid international phone number (6-15 digits, optional leading +). Tags can be a string or an array of up to 10 strings. The SMS type defaults to `transactional` but can be set to `marketing`; if the content includes a stop code, it is automatically treated as marketing. Returns the message ID, SMS count, credits used, and remaining credits.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

24 Package: sib_transactional_whatsapp

24.1 Procedures

24.1.1 sib_transactional_whatsapp.sendwhatsappmessage: Sendinblue Package for Transactional WhatsApp operations that could not be mapped to a table.

<Note>You can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating Whatsapp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your account</Note>This endpoint is used to send a WhatsApp message.
(**The first message you send using the API must contain a Template ID. You must create a template on WhatsApp on the Brevo platform to fetch the Template ID.**)

25 Package: sib_user

25.1 Procedures

25.1.1 sib_user.edituserpermission: Sendinblue Package for User operations that could not be mapped to a table.

`Feature` - A Feature represents a specific functionality like Emailcampaign, Deals, Calls, Automations, etc. on Brevo. While inviting a user,determine which feature you want to manage access to. You must specify thefeature accurately to avoid errors. `Permission` - A Permission defines thelevel of access or control a user has over a specific feature. Whileinviting user, decide on the permission level required for the selectedfeature. Make sure the chosen permission is related to the selected feature.Features and their respective permissions are as below: - `email_campaigns`:- "create_edit_delete"- "send_schedule_suspend"- `sms_campaigns`:- "create_edit_delete"- "send_schedule_suspend"- `contacts`:- "view"- "create_edit_delete"- "import"- "export"- "list_and_attributes"- "forms"- `templates`:- "create_edit_delete"- "activate_deactivate"- `workflows`:- "create_edit_delete"- "activate_deactivate_pause"- "settings"- `landing_pages`:- "all"- `transactional_emails`:- "settings"- "logs"- `smtp_api`:- "smtp"- "api_keys"- "authorized_ips"- `user_management`:- "all"- `sales_platform`:- "create_edit_deals"- "delete_deals"- "manage_others_deals_tasks"- "reports"- "settings"- `phone`:- "all"- `conversations`:- "access"- "assign"- "configure"- `senders_domains_dedicated_ips`:- "senders_management"- "domains_management"- "dedicated_ips_management"- `push_notifications`:- "view"- "create_edit_delete"- "send"- "settings"- `companies`:- "manage_owned_companies"- "manage_other_companies"- "settings"**Note**: - The privileges array remains the same as in the send invitation;the user simply needs to provide the permissions that need to be updated. -The availability of feature and its permission depends on your current plan.Please select the features and permissions accordingly.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_user.edituserpermission. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
all_features_access	boolean	☑		All access to the features (Values: true, false)
email	string	☑		Email address for the organization

25.1.2 sib_user.inviteuser: Sendinblue Package for User operations that could not be mapped to a table.

`Feature` - A Feature represents a specific functionality like Emailcampaign, Deals, Calls, Automations, etc. on Brevo. While inviting a user,determine which feature you want to manage access to. You must specify thefeature accurately to avoid errors. `Permission` - A Permission defines thelevel of access or control a user has over a specific feature. Whileinviting user, decide on the permission level required for the selectedfeature. Make sure the chosen permission is related to the selected feature.Features and their respective permissions are as below: - `email_campaigns`:- "create_edit_delete"- "send_schedule_suspend"- `sms_campaigns`:- "create_edit_delete"- "send_schedule_suspend"- `contacts`:- "view"- "create_edit_delete"- "import"- "export"- "list_and_attributes"- "forms"- `templates`:- "create_edit_delete"- "activate_deactivate"- `workflows`:- "create_edit_delete"- "activate_deactivate_pause"- "settings"- `landing_pages`:- "all"- `transactional_emails`:- "settings"- "logs"- `smtp_api`:- "smtp"- "api_keys"- "authorized_ips"- `user_management`:- "all"- `sales_platform`:- "create_edit_deals"- "delete_deals"- "manage_others_deals_tasks"- "reports"- "settings"- `phone`:- "all"- `conversations`:- "access"- "assign"- "configure"- `senders_domains_dedicated_ips`:- "senders_management"- "domains_management"- "dedicated_ips_management"- `push_notifications`:- "view"- "create_edit_delete"- "send"- "settings"- `companies`:- "manage_owned_companies"- "manage_other_companies"- "settings"**Note**: - If `all_features_access: false` then only privileges arerequired otherwise if `true` then it's assumed that all permissions will bethere for the invited user. - The availability of feature and its permissiondepends on your current plan. Please select the features and permissionsaccordingly.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_user.inviteuser. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
all_features_access	boolean	☑		All access to the features (Values: true, false)
email	string	☑		Email address for the organization

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

25.1.3 sib_user.putresendcancelinvitation: Sendinblue Package for User operations that could not be mapped to a table.

Resends or cancels a pending invitation for a user in the organization, depending on the action path parameter. Use `resend` to send a new invitation email to the user, or `cancel` to revoke the pending invitation entirely and remove the user''s pending access.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_user.putresendcancelinvitation. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
p_action	string	☑		action (Values: resend, cancel)
p_email	string	☑		Email of the invited user.

25.1.4 sib_user.putrevokeuserpermission: Sendinblue Package for User operations that could not be mapped to a table.

Revokes all permissions for an invited user in the organization, effectively removing their access to the platform. If the user''s plan change generated credit notes, they are returned in the response for billing reconciliation.

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_user.putrevokeuserpermission. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
p_email	string	☑		Email of the invited user.

26 Package: sib_webhooks

26.1 Procedures

26.1.1 sib_webhooks.exportwebhookshistory: Sendinblue Package for Webhooks operations that could not be mapped to a table.

<Note>This is an enterprise feature.To have it activated please send us a request and we will activate it for your account.</Note>Exports webhook event history to CSV format for analysis and reporting.Use this to:- Generate comprehensive webhook event reports- Analyze webhook delivery patterns and success rates- Export event data for external analysis tools- Create historical reports for compliance and auditing- Track webhook performance and reliability metricsKey information returned:- Process ID for tracking export completion- CSV file will be delivered to specified webhook URL

27 Package: sib_whatsapp_campaigns

27.1 Procedures

27.1.1 sib_whatsapp_campaigns.createwhatsapptemplate: Sendinblue Package for WhatsApp Campaigns operations that could not be mapped to a table.

<Note>You can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activating WhatsApp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your account</Note>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_whatsapp_campaigns.createwhatsapptemplate. 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 `begin package.procedure(value1, value2, value3); end` on a procedure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same procedure 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
bodyText	string	☑		Body of the template. **Maximum allow ed characters are 1024**
category	string	☑		Category of the template (Values: MARKETING, UTILITY)
headerText	string	☐		Text content of the header in the template. **Maximum allow ed characters are 45** **Use this field to add text content in tem-

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
				plate header and if mediaUrl is empty**
language	string	☑		Language of the template. For Example :**en** for English
mediaUrl	string	☐		Absolute url of the media file **(no local file)** for the header. **Use this field in you w ant to add media in Template header and headerText is empty** Al-low ed extensions for media files are: #### jpeg | png | mp4 | pdf
name	string	☑		Name of the template
source	string	☐		source of the template (Values: Automation, Conversations)

27.1.2 sib_whatsapp_campaigns.sendwhatsapptemplateapproval: Sendinblue Package for WhatsApp Campaigns operations that could not be mapped to a table.

<Note>You can use this API for WhatsApp only if you have setup your WhatsApp account on Brevo platform. To setup your WhatsApp account, follow the steps in the guide below.[Activ-ating WhatsApp](https://developers.brevo.com/docs/whatsapp-campaigns-1) in your ac-count</Note>

Parameters of Procedure

The following parameters can be used to control the behaviour of the procedure sib_whats-app_campaigns.sendwhatsapptemplateapproval. 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 `begin package.procedure(value1, value2, value3); end` on a proced-ure 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 `begin package.procedure(name1 => value1, name3 => value3); end` on the same pro-cedure will use the default values for the second and fourth parameters and the specified val-ues for the first and third.

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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Index

- # -

#Clickers 89, 301
#Complaints 89, 301
#Deferred 89, 301
#Delivered 89, 301, 307, 309, 310, 312, 314, 315, 317, 319
#Hard-bounces 89, 301
#Return-bounces 89, 301
#Sent 89, 301, 307, 309, 310, 312, 314, 315, 317, 319
#Soft-bounces 89, 301
#Trackable Views 89, 301
#Unique Clicks 89, 277, 278, 301
#Unique Opens 277, 278
#Unique Subscribers 86, 87, 89, 91, 92, 300, 301
#Unique Views 89, 301
#Unsubscriptions 89, 301
#Viewed 89, 301

- A -

A/B Test Campaign Result by ID 151
A/B Test Campaign Results 299
A/B-Testing 152, 156
ABONNEMENTEN_ACTIEF 70, 76, 79
ABONNEMENTEN_ACTIEF_TOT 70, 76, 79
AbTestCampaignResultById 151
AbTestCampaignResults 299
abTesting 152, 156
Accepted 291, 292
AccountActivities 17
action 17
Active 191, 193, 241, 242, 244
add-odata-mandatory-filters 2
Address City 19
Address Country 19
Address Street 19
Address Zip Code 19
address_city 19
address_country 19
address_street 19
address_zipCode 19
agentId 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
agentName 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
agentUserpic 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
alternativePrice 145, 146, 148
alternativePrice[eq] 148
alternativePrice[gt] 148
alternativePrice[gte] 148
alternativePrice[lt] 148
alternativePrice[lte] 148
alternativePrice[ne] 148
amount 26, 28
analysis-enforce-row-uniqueness 2
api-access-token 2
api-client-id 2
api-client-secret 2
api-pre-expiry-refresh-sec 2
api-redirect-url 2
api-refresh-token 2
api-scope 2
api-token-url 2
api-url 2
assignToId 253, 255, 256, 257, 259, 260, 261, 262
association 120, 121
Attachment Count 284, 287
attachmentCount 284, 287
attachmentFile 152, 156
attachmentUrl 152, 156
Attribute Values 42
AttributeMultiCategoryOptions 41
Attributes 33, 34, 36, 37, 38, 39, 41, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 120, 121, 123, 124, 126, 128, 129
attributeTypeName 40, 125, 126
AttributeValues 42
attributionPerConsumer 228, 231, 234, 237
auth 302, 303, 305, 306
authenticated 134, 136
authorId 168, 170, 204, 205, 206, 207, 209, 210, 211, 212
authType 162, 163, 165, 166
automation-key 2
averageBasket 139, 141

- B -

Background Process by ID 214
Background Processes 216
BackgroundProcessById 214
BackgroundProcesses 216
balance_contactId 219, 220, 221, 223
balance_loyaltyProgramId 219, 220, 221, 223
balanceAvailabilityDurationModifier 20, 22

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

balanceAvailabilityDurationUnit 20, 22
balanceAvailabilityDurationValue 20, 22
BalanceDefinitionByPidAndBdid 20
balanceDefinitionId 23, 24, 25, 26, 28, 29, 30, 32, 228, 231, 234, 237, 265
BalanceDefinitionsByPid 22
balanceExpirationDate 20, 22, 28
BalanceLimitsByPidAndBdid 23
balanceOptionAmountOvertakingStrategy 20, 22
balanceOptionCreditRounding 20, 22
balanceOptionDebitRounding 20, 22
batched 302, 303, 305, 306
bdid 20, 23
billingInfo_address_countryCode 188, 189
billingInfo_address_locality 188, 189
billingInfo_address_postalCode 188, 189
billingInfo_address_stateCode 188, 189
billingInfo_address_streetAddress 188, 189
billingInfo_companyName 188, 189
billingInfo_email 188, 189
billingInfo_name_familyName 188, 189
billingInfo_name_givenName 188, 189
birthdayAttribute 217, 218
BLACKLIST 70, 76, 79
blid 23
BLOB Preferred 202
BLOB_PREFERRED 202
Blocked 277, 278, 291, 292
Blocked at 283
Blocked Domains 273
blockedAt 283
BlockedDomains 273
Body 284, 287, 294
BOL_RESPONSE_CACHE_MAX_AGE_SEC 202
brand 145, 146, 148
BUBS_GBR_ID 70, 76, 79
BUBS_GBR_ID_MASTER 70, 76, 79
bulk-delete-page-size-rows 2
bulk-insert-page-size-bytes 2
bulk-insert-page-size-rows 2
businessStatus 320
button_type 312, 314, 315

- C -

cache 162, 163, 165, 166
Calculated Value 41, 42
calculatedValue 41, 42
Campaign ID 44, 45, 46, 49, 51, 53, 55, 57, 59, 61, 63, 68, 72, 74, 75, 81, 85, 89, 301
campaignId 44, 45, 46, 49, 51, 53, 55, 57, 59, 61, 63, 68, 72, 74, 75, 81, 85, 89, 151, 152, 160, 245, 246, 248, 249, 299, 301, 307, 309, 310, 312, 314, 315, 317
campaignName 299, 307, 309, 310, 312, 314, 315, 317, 319
campaignStatus 307, 309, 310, 312, 314, 315, 317, 319
CampaignSubject 299
cancelledAt 28
cart_currency 213
cart_specificAmount 213
Categories 136
Category 41, 42, 312, 314, 315, 321
CategoryById 138
categoryName 93
cid 31, 32
CLICKERS 70, 76, 79
Clickrate 151, 299
Clicks 277, 278
codeCount 217, 218, 228, 231, 234, 237
CodeCountsByPidAndCpid 224
codeGeneratorId 228, 231, 234, 237
codePoolId 228, 231, 234, 237
Companies 33
Companies_ItemsLinkedContactsIds 34
Companies_ItemsLinkedDealsIds 36
Companies_LinkedContactsIdsById 37
Companies_LinkedDealsIdsById 38
CompaniesById 39
Company Name 19, 70, 76, 79, 176, 188, 189, 190, 191, 192, 193
COMPANY_NAME 70, 76, 79
companyId 168, 170
companyName 19, 176, 188, 189, 190, 191, 192, 193
completedAt 28
compliance_compliance_toggle 252
compliance_organisation_prefix 252
compliance_stop_keyword 252
config 228, 231, 234, 237
configuration_customSuccessUrl 213
constraintType 23
consumedAt 225
Contact Complaint Statistics by Email 45
Contact Detail Complaint Statistics by Email 51
Contact Detail Hard Bounce Statistics by Email 55
Contact Detail Link Click Statistics by Email 57
Contact Detail Message Sent Statistics by Email 59
Contact Detail Open Statistics by Email 61
Contact Detail Soft Bounce Statistics by Email 63

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Contact Detail Transaction Attributes by Email 65
Contact Detail Unsubscriptions by Email (Administrative Mode) 47
Contact Detail User Unsubscriptions by Email 67
Contact Hard Bounce Statistics by Email 68
Contact Information by Email 70
Contact Link Clicks by Email 72
Contact Message Sent Statistics by Email 74
Contact Open Statistics by Email 75
Contact Soft Bounce Statistics by Email 81
Contact Transaction Attributes by Email 83
Contact Unsubscriptions by Email (Administrative Mode) 43
Contact Unsubscriptions by Email (User Mode) 85
contact_id 161
contact_properties 161
CONTACT_TIMEZONE 70, 76, 79
ContactAdminUnsubscriptionsByEmail 43
ContactBalances 24
ContactBalances_Balances 25
ContactClickStatisticsByEmail 44
ContactComplaintStatisticsByEmail 45
ContactDeliveryStatisticsByEmail 46
ContactDetailAdminUnsubscriptionsByEmail 47
ContactDetailClickStatisticsByEmail 49
ContactDetailComplaintStatisticsByEmail 51
ContactDetailDeliveryStatisticsByEmail 53
ContactDetailHardBounceStatisticsByEmail 55
ContactDetailLinkClickStatisticsByEmail 57
ContactDetailMessageSentStatisticsByEmail 59
ContactDetailOpenStatisticsByEmail 61
ContactDetailSoftBounceStatisticsByEmail 63
ContactDetailTransactionAttributesByEmail 65
ContactDetailUserUnsubscriptionsByEmail 67
ContactHardBounceStatisticsByEmail 68
contactId 25, 26, 28, 29, 30, 31, 32, 168, 170, 213, 219, 220, 221, 223, 225, 240
contactIds 204, 206, 207, 209, 211, 212
ContactInfoByEmail 70
ContactLinkClicksByEmail 72
ContactMessageSentStatisticsByEmail 74
contactNumber 294
ContactOpenStatisticsByEmail 75
Contacts 76
Contacts by List ID 79
ContactsByListId 79
contactsIds 253, 256, 257, 259, 261, 262
ContactSoftBounceStatisticsByEmail 81
ContactStats 82
ContactTransactionAttributesByEmail 83
ContactUserUnsubscriptionsByEmail 85
contains_button 312, 314, 315
Content 245, 246, 248, 249, 250
Content Type 202
CONTENT_TYPE 202
contentId 174
contentLength 174
contentType 174
Contract Code 70, 76, 79
CONTRACT_CODE 70, 76, 79
ConversationMessageBccsByConversationId 93
ConversationMessageCcsByConversationId 96
ConversationMessageTosByConversationId 98
ConversationPushedMessageBccsByConversationId 100
ConversationPushedMessageCcsByConversationId 103
ConversationPushedMessageTosByConversationId 105
ConversationsMessages_AttachmentsById 107
ConversationsMessagesById 110
ConversationsPushedMessages_AttachmentsById 112
ConversationsPushedMessagesById 115
conversionSource 139, 141, 142
conversionSourceId 141, 142
CorporateGroup_Sub-accountsById 176
CorporateGroup_UsersById 177
CorporateGroupById 178
CorporateInvitedUserAppsManagement 179
CorporateInvitedUserMyPlans 180
CorporateInvitedUsers 181
CorporateInvitedUsersList_UsersFeature_accessAnalytics 182
CorporateInvitedUsersList_UsersFeature_accessApi_keys 183
CorporateInvitedUsersList_UsersFeature_accessCreate_sub_organizations 184
CorporateInvitedUsersList_UsersFeature_accessManage_sub_organizations 185
CorporateInvitedUsersList_UsersFeature_accessSub_organization_groups 186
CorporateInvitedUserUserManagement 187
CorporateIpAddresses 188
CorporateMasterAccounts 188
CorporatePlanMasterAccounts 189
CorporateSubAccountById 190
CorporateSubAccountGroups 191
CorporateSubAccountGroupsBySubAccountId 192
CorporateSubAccounts 193
CorporateUserPermissionAnalyticsByEmail 193
CorporateUserPermissionApiKeysByEmail 194
CorporateUserPermissionAppsManagementByEmail 195

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

CorporateUserPermissionCreateSubOrganizationsByEmail 196
CorporateUserPermissionGroupsByEmail 197
CorporateUserPermissionManageSubOrganizationsByEmail 198
CorporateUserPermissionMyPlansByEmail 198
CorporateUserPermissionsByEmail 199
CorporateUserPermissionSubOrganizationGroupsByEmail 200
CorporateUserPermissionUserManagementByEmail 201
Count 24, 25, 26, 28, 29, 30, 61, 72, 75, 135, 136, 224, 225, 240, 275
COUNTRY 70, 76, 79
CouponCollections 117
CouponsById 119
cpid 224
createdAt 20, 22, 23, 26, 28, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79, 87, 89, 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 117, 119, 120, 121, 136, 138, 145, 146, 148, 152, 156, 162, 163, 165, 166, 168, 170, 176, 191, 193, 204, 205, 206, 207, 209, 210, 211, 212, 217, 218, 220, 221, 223, 225, 226, 228, 231, 234, 237, 245, 246, 248, 249, 250, 252, 253, 255, 256, 257, 259, 260, 261, 262, 264, 265, 266, 268, 269, 270, 272, 275, 280, 281, 301, 302, 303, 305, 306, 307, 309, 310, 312, 314, 315, 317, 319, 321
createdAt_2 228, 231, 265, 266
createdSince 33, 34, 36, 76, 123, 124, 126, 136, 144, 148
Creation Date 20, 22, 23, 26, 28, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79, 87, 89, 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 117, 119, 120, 121, 136, 138, 145, 146, 148, 152, 156, 162, 163, 165, 166, 168, 170, 176, 191, 193, 204, 205, 206, 207, 209, 210, 211, 212, 217, 218, 220, 221, 223, 225, 226, 228, 231, 234, 237, 245, 246, 248, 249, 250, 252, 253, 255, 256, 257, 259, 260, 261, 262, 264, 265, 266, 268, 269, 270, 272, 275, 280, 281, 301, 302, 303, 305, 306, 307, 309, 310, 312, 314, 315, 317, 319, 321
creator_creationDate 136
creator_email 136
creator_id 136
Crm_CompanyIdsNotes 204
Crm_ContactIdsNotes 205
Crm_DealIdsNotes 206
Crm_ItemsCompaniesIdsTasks 253
Crm_ItemsContactsIdsTasks 255
Crm_ItemsDealsIdsTasks 256
Crm_ItemsLinkedCompaniesIdsDeals 123
Crm_ItemsLinkedContactsIdsDeals 124
CrmAttributes_AttributeOptionsCompanies 40
CrmAttributes_AttributeOptionsDeals 125
CrmAttributesCompanies 40
CrmAttributesDeals 126
CrmDeals 126
CrmDeals_LinkedCompaniesIdsById 128
CrmDeals_LinkedContactsIdsById 129
CrmDealsById 129
CrmFiles 168
CrmFilesById 169
CrmFilesByIdData 170
CrmNotes 207
CrmNotes_CompanyIdsById 209
CrmNotes_ContactIdsById 210
CrmNotes_DealIdsById 211
CrmNotesById 212
CrmPipeline_StagesDetails 130
CrmPipelineDetails 131
CrmPipelineDetails_StagesByPipelineID 131
CrmPipelineDetailsByPipelineID 132
CrmTasks 257
CrmTasks_CompaniesIdsById 259
CrmTasks_ContactsIdsById 260
CrmTasks_DealsIdsById 261
CrmTasksById 262
CrmTasktypes 263
currencyCode 188, 189
currencyCode_2 189
current_page 135, 136
customTemplateId 280, 281

- D -

D31APPLICATIONS 70, 76, 79
D31BILLINGIDS 70, 76, 79
D31BUBSGBRIDS 70, 76, 79
D31DATACONTAINERCNT 70, 76, 79
D31MBIO 70, 76, 79
D31PARTITIONCNT 70, 76, 79
D31PROVIDERCNT 70, 76, 79
D31ROWCNT 70, 76, 79
D31TABLECNT 70, 76, 79
Database Driver 1
datatype 312, 315
Date 17, 172, 175, 253, 255, 256, 257, 259, 260, 261, 262, 274, 278, 284, 286, 287, 288, 289, 292, 294
DATE_ENDED 202
DATE_STARTED 202
dateFrom 168, 204, 205, 206, 207, 253, 255, 256, 257
dateTo 168, 204, 205, 206, 207, 253, 255, 256, 257
days 274, 277, 278, 289, 291, 292, 294
dealId 168, 170
default 312, 315

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

defaultCoupon 117, 119
deletedAt 20, 22
Delivered 277, 278, 291, 292
deliveredAt 173, 174, 175
disabledAt 228, 231, 234, 237
display_header 312, 314, 315
dns_records_brevo_code_host_name 134
dns_records_brevo_code_status 134
dns_records_brevo_code_type 134
dns_records_brevo_code_value 134
dns_records_dkim_record_host_name 134
dns_records_dkim_record_status 134
dns_records_dkim_record_type 134
dns_records_dkim_record_value 134
dns_records_dmarc_record_host_name 134
dns_records_dmarc_record_status 134
dns_records_dmarc_record_type 134
dns_records_dmarc_record_value 134
DOCUMENT 171
documentId 217, 218
DOI Template 280, 281
doiTemplate 280, 281
Domain 134, 188, 241, 242, 302, 303, 305, 306
domain_name 136
DomainByName 134
domainName 134
Domains 135
Domains_Domains 136
done 253, 255, 256, 257, 259, 260, 261, 262
DOUBLE_OPT-IN 70, 76, 79
downgradeSchedule_durationModifier 264, 269, 270, 272
downgradeSchedule_durationUnit 264, 269, 270, 272
downgradeSchedule_durationValue 264, 269, 270, 272
downgradeSchedule_scheduledDate 264, 269, 270, 272
downgradeSchedule_strategy 264, 269, 270, 272
downgradeStrategy 264, 269, 270, 272
download-error-400-bad-request-max-tries 2
download-error-400-bad-request-sleep-initial-ms 2
download-error-400-bad-request-sleep-max-ms 2
download-error-400-bad-request-sleep-multiplicator 2
download-error-403-forbidden-max-tries 2
download-error-403-forbidden-sleep-initial-ms 2
download-error-403-forbidden-sleep-max-ms 2
download-error-403-forbidden-sleep-multiplicator 2
download-error-408-request-timeout-max-tries 2
download-error-408-request-timeout-sleep-initial-ms 2
download-error-408-request-timeout-sleep-max-ms 2
download-error-408-request-timeout-sleep-multiplicator 2
download-error-422-bad-request-max-tries 2
download-error-422-bad-request-sleep-initial-ms 2
download-error-422-bad-request-sleep-max-ms 2
download-error-422-bad-request-sleep-multiplicator 2
download-error-429-too-many-requests-max-tries 2
download-error-429-too-many-requests-sleep-initial-ms 2
download-error-429-too-many-requests-sleep-max-ms 2
download-error-429-too-many-requests-sleep-multiplicator 2
download-error-500-internal-server-error-max-tries 2
download-error-500-internal-server-error-sleep-initial-ms 2
download-error-500-internal-server-error-sleep-max-ms 2
download-error-500-internal-server-error-sleep-multiplicator 2
download-error-502-server-unavailable-max-tries 2
download-error-502-server-unavailable-sleep-initial-ms 2
download-error-502-server-unavailable-sleep-max-ms 2
download-error-502-server-unavailable-sleep-multiplicator 2
download-error-503-server-unavailable-max-tries 2
download-error-503-server-unavailable-sleep-initial-ms 2
download-error-503-server-unavailable-sleep-max-ms 2
download-error-503-server-unavailable-sleep-multiplicator 2
download-error-504-gateway-timeout-max-tries 2
download-error-504-gateway-timeout-sleep-initial-ms 2
download-error-504-gateway-timeout-sleep-max-ms 2
download-error-504-gateway-timeout-sleep-multiplicator 2
download-error-590-network-connect-timeout-max-tries 2
download-error-590-network-connect-timeout-sleep-initial-ms 2
download-error-590-network-connect-timeout-sleep-max-ms 2
download-error-590-network-connect-timeout-sleep-multiplicator 2
download-error-599-network-connect-timeout-max-tries 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

download-error-599-network-connect-timeout-sleep-initial-ms 2
download-error-599-network-connect-timeout-sleep-max-ms 2
download-error-599-network-connect-timeout-sleep-multiplicator 2
download-error-argument-exception-max-tries 2
download-error-argument-exception-sleep-initial-ms 2
download-error-argument-exception-sleep-max-ms 2
download-error-argument-exception-sleep-multiplicator 2
download-error-internet-down-max-tries 2
download-error-internet-down-sleep-initial-ms 2
download-error-internet-down-sleep-max-ms 2
download-error-internet-down-sleep-multiplicator 2
download-error-io-exception-max-tries 2
download-error-io-exception-sleep-initial-ms 2
download-error-io-exception-sleep-max-ms 2
download-error-io-exception-sleep-multiplicator 2
download-error-json-exception-max-tries 2
download-error-json-exception-sleep-initial-ms 2
download-error-json-exception-sleep-max-ms 2
download-error-json-exception-sleep-multiplicator 2
download-error-name-resolution-failure-max-tries 2
download-error-name-resolution-failure-sleep-initial-ms 2
download-error-name-resolution-failure-sleep-max-ms 2
download-error-name-resolution-failure-sleep-multiplicator 2
download-error-other-exception-max-tries 2
download-error-other-exception-sleep-initial-ms 2
download-error-other-exception-sleep-max-ms 2
download-error-other-exception-sleep-multiplicator 2
download-error-socket-exception-max-tries 2
download-error-socket-exception-sleep-initial-ms 2
download-error-socket-exception-sleep-max-ms 2
download-error-socket-exception-sleep-multiplicator 2
download-error-web-exception-max-tries 2
download-error-web-exception-sleep-initial-ms 2
download-error-web-exception-sleep-max-ms 2
download-error-web-exception-sleep-multiplicator 2
download-error-web-not-found-max-tries 2
download-error-web-not-found-sleep-initial-ms 2
download-error-web-not-found-sleep-max-ms 2
download-error-web-not-found-sleep-multiplicator 2
download-error-web-not-implemented-max-tries 2
download-error-web-not-implemented-sleep-initial-ms 2
download-error-web-not-implemented-sleep-max-ms 2
download-error-web-not-implemented-sleep-multiplicator 2
download-error-web-timeout-max-tries 2
download-error-web-timeout-sleep-initial-ms 2
download-error-web-timeout-sleep-max-ms 2
download-error-web-timeout-sleep-multiplicator 2
download-error-web-unauthorized-max-tries 2
download-error-web-unauthorized-sleep-initial-ms 2
download-error-web-unauthorized-sleep-max-ms 2
download-error-web-unauthorized-sleep-multiplicator 2
download-error-web-unknown-max-tries 2
download-error-web-unknown-sleep-initial-ms 2
download-error-web-unknown-sleep-max-ms 2
download-error-web-unknown-sleep-multiplicator 2
downloadToken 171
DRY_RUN 202
DUMMY 82, 144
duplicate-row-mode 2
Duration (ms) 202
DURATION_MS 202
durationUnit 23, 228
durationValue 23, 228
Dynamic List 87, 89, 301
dynamicList 87, 89, 301

- E -

EcommerceAttribution_ResultsMetrics 139
EcommerceAttributionMetrics 140
EcommerceAttributionMetricsByConversionSourceConversionSourceId 141
EcommerceAttributionProductsByConversionSourceConversionSourceId 142
EcommerceDisplayCurrencies 143
editorType 281
email 17, 19, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79, 93, 96, 98, 100, 103, 105, 177, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 242, 244, 274, 283, 284, 286, 287, 288, 295, 296, 297, 298
Email Address 19, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79, 93, 96, 98, 100, 103, 105, 177, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 242, 244, 274, 283, 284, 286, 287, 288, 295, 296, 297, 298
Email Campaign by ID 152
Email Campaigns 156
Email is Blacklisted 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
emailBlacklisted 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

EmailCampaignById 152
EmailCampaigns 156
EmailEvents 274
emailExpirationDate_duration 152, 156
emailExpirationDate_unit 152, 156
encryption 252
End Date 202, 226, 228, 231, 234, 237
endDate 17, 43, 44, 45, 46, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 68, 70, 72, 74, 75, 81, 82, 83, 85, 87, 89, 156, 161, 165, 166, 172, 226, 228, 231, 234, 237, 250, 274, 275, 277, 278, 283, 286, 288, 289, 291, 292, 294, 319, 321
enterprise 19
entity 168, 204, 205, 206, 207
entityIds 168, 204, 205, 206, 207
Error Message Code 202
Error Message Text 202
ERROR_MESSAGE_CODE 202
ERROR_MESSAGE_TEXT 202
errorReason 319, 321
Estate Person ID 70, 76, 79
Event 274, 289, 294
Event Time 43, 45, 46, 51, 53, 55, 59, 61, 63, 68, 72, 74, 75, 81, 85
event_date 161
event_filter_id 161
event_name 161
event_properties 161
Events 161
eventTime 43, 45, 46, 51, 53, 55, 59, 61, 63, 68, 72, 74, 75, 81, 85
excludedLists 309, 310
excludeHtmlContent 152, 156
excludePdfAttachment 156
expirationDate 117, 119, 221, 225, 228, 231, 234, 237
expirationModifier 228, 231, 234, 237
expirationUnit 228, 231, 234, 237
expirationValue 228, 231, 234, 237
expiresAt 26
Export URL 214, 216
export_url 214, 216
EXT_ID 70, 76, 79
ExternalFeedByUUID 162
ExternalFeedByUUID_Headers 163
ExternalFeeds 165
ExternalFeeds_FeedsHeaders 166

- F -

Fail on Error 202
FAIL_ON_ERROR 202
feature 296, 297
feature_access_conversations 295
feature_access_crm 295
feature_access_marketing 295
file_imageInfo_height 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_imageInfo_previewLink 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_imageInfo_width 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_inlineId 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_isAllowedFileType 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_isImage 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_isInline 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_isSticker 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_link 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_mimeType 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_name 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
file_size 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
fileUrl 169
filter 76
filter[assignTo] 253, 255, 256, 257
filter[companies] 253, 255, 256, 257
filter[contacts] 253, 255, 256, 257
filter[date] 253, 255, 256, 257
filter[deals] 253, 255, 256, 257
filter[status] 253, 255, 256, 257
filter[type] 253, 255, 256, 257
filters[attributes.deal_name] 123, 124, 126
filters[attributes.name] 33, 34, 36
filters[linkedCompaniesIds] 123, 124, 126
filters[linkedContactsIds] 123, 124, 126
First Name 19, 70, 76, 79, 177
firstName 19, 70, 76, 79, 177
Folder by ID 86
Folder ID 87, 89, 91, 301
Folder Lists 300
FolderById 86
folderId 86, 87, 89, 91, 92, 300, 301
FolderLists 300
Foldername 300
Folders 87
FoldertotalBlacklisted 300
FoldertotalSubscribers 300

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

FolderuniqueSubscribers 300
Footer 152, 156
force-case-sensitive-identifiers 2
forced-casing-identifiers 2
Forums Login 70, 76, 79
FORUMS_LOGIN 70, 76, 79
forwardedToSourceStatus_error 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
forwardedToSourceStatus_isSuccess 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
From 274, 286, 288
from_email 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
from_name 93, 96, 98, 100, 103, 105, 107, 110, 112, 115

- G -

generator_createdAt 228, 231, 234, 237
generator_description 228, 231, 234, 237
generator_id 228, 231, 234, 237
generator_name 228, 231, 234, 237
generator_pattern 228, 231, 234, 237
generator_updatedAt 228, 231, 234, 237
gid 269, 272
group_createdAt 176, 177, 178
group_groupName 176, 177, 178
group_id 176, 177, 178
groupId 223, 265, 266, 268
groupName 202
groups_id 179, 180, 181, 182, 183, 184, 185, 186, 187
groups_name 179, 180, 181, 182, 183, 184, 185, 186, 187

- H -

hardBounces 277, 278, 291, 292
Hard-bounces 277, 278, 291, 292
Header 152, 156
header_type 312, 314, 315
hide_footer 312, 314, 315
html 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
HTML Content 152, 156, 280, 281
htmlContent 152, 156, 280, 281
HTTP Disk Cache Maximum Age (sec) 202
HTTP Memory Cache Maximum Age (sec) 202
HTTP Method 202
HTTP Status Code 202
HTTP_DISK_CACHE_MAX_AGE_SEC 202
HTTP_DISK_CACHE_SAVE 202
HTTP_DISK_CACHE_USE 202
HTTP_MEMORY_CACHE_MAX_AGE_SEC 202
HTTP_MEMORY_CACHE_SAVE 202
HTTP_MEMORY_CACHE_USE 202
HTTP_METHOD 202
HTTP_STATUS_CODE 202
http-disk-cache-compression-level 2
http-disk-cache-directory 2
http-disk-cache-ignore-write-errors 2
http-disk-cache-max-age-sec 2
http-get-timeout-max-ms 2
http-get-timeout-ms 2
http-memory-cache-compression-level 2
http-memory-cache-max-age-sec 2
http-post-timeout-max-ms 2
http-post-timeout-ms 2
http-public-disk-cache-directory 2

- I -

id 37, 38, 39, 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 119, 128, 129, 138, 145, 146, 169, 170, 176, 177, 178, 190, 192, 209, 210, 211, 212, 213, 259, 260, 261, 262
id_2 191, 245, 246
identifier 43, 44, 45, 46, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 68, 70, 72, 74, 75, 81, 82, 83, 85, 275
identifiers_ext_id 120, 121
identifiers_id 120, 121
identifierType 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70
ignore-http-400-errors 2
ignore-http-401-errors 2
ignore-http-402-errors 2
ignore-http-403-errors 2
ignore-http-404-errors 2
ignore-http-422-errors 2
ignore-http-429-errors 2
ignore-http-500-errors 2
ignore-http-502-errors 2
ignore-http-503-errors 2
ignore-unknown-path-type 2
ignore-values-unknown-path 2
imageInfo_height 107, 112
imageInfo_previewLink 107, 112
imageInfo_width 107, 112
imageRef 20, 22, 231, 265, 266, 268
imageUrl 142, 145, 146, 148
InboundEmailAttachmentByToken 171
InboundEmailEvents 172
InboundEmailEventsByUuid 173

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

InboundEmailEventsByUuid_Attachments 174
InboundEmailEventsByUuid_Logs 175
includedLists 307, 310
includeInternal 24, 25, 26, 29, 31, 32, 219, 220, 221, 223
info_import_duplicate_contact_id 214
info_import_duplicate_email_id 214
info_import_duplicate_ext_id 214
info_import_duplicate_landline_number_id 214
info_import_duplicate_phone_id 214
info_import_duplicate_whatsapp_id 214
info_import_invalid_emails 214
Inline Image Activation 152, 156
inlineId 107, 112
inlineImageActivation 152, 156
integrations 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
internalName 40, 125, 126
Invalid 277, 278
invalidatedContacts 319
invalid-json-on-get-max-tries 2
invalid-json-on-get-sleep-initial-ms 2
invalid-json-on-get-sleep-max-ms 2
invalid-json-on-get-sleep-multiplicator 2
invalid-json-on-post-max-tries 2
invalid-json-on-post-sleep-initial-ms 2
invalid-json-on-post-sleep-max-ms 2
invalid-json-on-post-sleep-multiplicator 2
invantive-sql-compress-sparse-arrays 2
invantive-sql-correct-invalid-date 2
invantive-sql-execution-profile-disk-path 2
invantive-sql-execution-profile-to-disk 2
invantive-sql-forward-filters-to-data-containers 2
invantive-sql-share-byte-arrays 2
invantive-sql-share-strings 2
invantive-sql-shuffle-fetch-results-data-containers 2
invantive-use-cache 2
InvitedUsers 295
IP 43, 61, 72, 75, 85, 136, 188, 241, 242, 244, 274
IP Addresses 241
IP Addresses by Send ID 241
IpAddresses 241
IpAddressesBySenderId 241
Is Active 280, 281
is_owner 179, 180, 181, 182, 183, 184, 185, 186, 187, 295
isActive 280, 281
isAllowedFileType 107, 112
isBot 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isDeleted 136, 138, 145, 146, 148
isForward 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isImage 107, 112
isInline 107, 112
isMissed 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isMissedByVisitor 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isPushed 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isRequired 40, 125, 126
isSentViaJsApi 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isSticker 107, 112
isTrigger 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
isValueReadonly 40, 125, 126

- J -

Job Title 70, 76, 79
JOB_TITLE 70, 76, 79
JOBTITLE 70, 76, 79
join-set-points-per-request 2

- K -

Key 40, 125

- L -

Label 40, 42, 125, 126
LANDLINE_NUMBER 70, 76, 79
language 312, 314, 315, 321
Language Code 70, 76, 79
LANGUAGE_CODE 70, 76, 79
Last Name 19, 70, 76, 79, 177
lastName 19, 70, 76, 79, 177
limit-partition-calls-left 2
limitValue 228
Link 107, 112, 274
linkedContactsIds 33, 34, 36, 38, 39, 123, 126, 128, 129
linkedDealsIds 33, 34, 36
LINKEDIN 70, 76, 79
List by ID 87
List Campaign Statistics 301
List Campaign Statistics by List ID 89
List ID 89, 301
List IDs 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
ListById 87
ListCampaignStatistics 301

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ListCampaignStatisticsByListId 89
listId 79, 87, 89
listIds 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
ListOfTierGroups 264
Lists 91
Lists by Folder ID 92
ListsByFolderId 92
listUnsubscribed 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
log-native-calls-to-disk-max-events 2
log-native-calls-to-disk-max-seconds 2
log-native-calls-to-disk-on-error 2
log-native-calls-to-disk-on-success 2
log-native-calls-to-trace 2
LoyaltyBalanceProgramsByPid_ActiveBalancesActiveBalance 26
LoyaltyBalanceProgramsByPid_TransactionHistoryTransactionHistory 28
LoyaltyBalanceProgramsByPidActiveBalance 29
LoyaltyBalanceProgramsByPidTransactionHistory 30
LoyaltyOfferProgramsByPid_ContactRewardsVouchers 225
LoyaltyOfferProgramsByPid_ItemsOffers 226
LoyaltyOfferProgramsByPidOffers 227
LoyaltyOfferProgramsByPidRewards_LimitsRid 228
LoyaltyOfferProgramsByPidRewards_ProductsRid 231
LoyaltyOfferProgramsByPidRewards_RedeemRulesRid 234
LoyaltyOfferProgramsByPidRewardsRid 237
LoyaltyOfferProgramsByPidVouchers 240
loyaltyProgramId 24, 25, 26, 28, 29, 30, 31, 32, 221, 223, 225, 226, 228, 231, 234, 237, 240, 264, 265, 266, 268, 269, 270, 272
LoyaltyPrograms 217
LoyaltyProgramsByPid 218
LoyaltyProgramTier_ItemsAccessConditions 265
LoyaltyProgramTier_ItemsTierRewards 266
LoyaltyProgramTiersByPid 268
loyaltySubscriptionId 25, 28, 30, 219, 220, 221, 223, 225, 240

- M -

Marketing Automation Enabled 19
Marketing Automation Key 19
marketingAutomation_enabled 19
marketingAutomation_key 19
maxAmount 20, 22
maxCreditAmountLimit 20, 22
maxDebitAmountLimit 20, 22
maximum-discovered-column-count 2
maximum-length-identifiers 2
max-odata-filters 2
max-odata-rewrite-in-count 2
maxRetries 162, 163, 165, 166
max-url-length-accepted 2
max-url-length-desired 2
media_file 252
media_file_size 252
media_url 252
mediaUrl 294
memberContactId 220
message 252
Message ID 173, 174, 175, 274, 286, 288, 289, 294
messageId 173, 174, 175, 274, 286, 288, 289, 294
messageType 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
meta 20, 22, 28, 217, 218, 221, 223, 225, 228, 231, 234, 237
metadata_key_value 225, 240
metadata-cache-max-age-sec 2
metaInfo 145, 146, 148
mimeType 107, 112
minAmount 20, 22
minimumValue 265
Mirror Active 152, 156
mirrorActive 152, 156
Mobile Number 70, 76, 79
MOBILE_NUMBER 70, 76, 79
Modified Date 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79, 136, 138, 145, 146, 148, 152, 156, 162, 163, 165, 166, 245, 246, 248, 249, 250, 280, 281, 302, 303, 305, 306, 307, 309, 310, 312, 314, 315, 317, 319, 321
modifiedAt 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79, 136, 138, 145, 146, 148, 152, 156, 162, 163, 165, 166, 245, 246, 248, 249, 250, 280, 281, 302, 303, 305, 306, 307, 309, 310, 312, 314, 315, 317, 319, 321
modifiedSince 33, 34, 36, 76, 79, 123, 124, 126, 136, 144, 148

- N -

Name 20, 22, 41, 42, 86, 87, 89, 91, 92, 93, 96, 98, 100, 103, 105, 107, 112, 117, 119, 130, 131, 133, 136, 138, 142, 145, 146, 148, 152, 156, 162, 163, 165, 166, 168, 170, 174, 189, 190, 191, 192, 197, 214, 216, 217, 218, 226, 228, 231, 234, 237, 242, 244, 245, 246, 248, 249, 250, 252, 253, 255, 256, 257, 259, 260, 261, 262, 264, 265, 266, 268, 269, 270, 272, 280, 281, 284, 299, 300, 301, 312, 314, 315, 321

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

name_2 163, 166, 192, 245, 246, 312, 315
Native Platform Scalar Requests 202
NATIVEPLATFORMSCALARREQUESTS 202
newCustomersCount 141
nextBillingAt 189
notes 253, 255, 256, 257, 259, 260, 261, 262
notification_channel 213
notification_text 213
npt 202
numberOfRemindersSent 213

- O -

oauth-unauthorized-max-tries 2
oauth-unauthorized-sleep-initial-ms 2
oauth-unauthorized-sleep-max-ms 2
oauth-unauthorized-sleep-multiplicator 2
object_type 120, 121, 161
Open Rate 151, 299
openRate 151, 299
Opens 277, 278
OPT_IN 70, 76, 79
Order Date 83
Order ID 83
Order Price 83
orderDate 83
orderId 83
orderPrice 83
Orders 144
ordersCount 139, 141, 142
organisationPrefix 245, 246, 248, 249, 250
organization_id 19
ORIG_SYSTEM_GROUP 202
ORIG_SYSTEM_REFERENCE 202
Original System Group 202
Original System Reference 202

- P -

page 33, 34, 36
page_num 120, 121
ParameterSubscriptionById 219
ParameterSubscriptionInfo_Members 220
ParameterSubscriptionRewardsByPid 221
ParameterSubscriptionTiersByPid 223
params 219, 220, 221, 223
parentId 145, 146, 148
partition-slot-based-rate-limit-length-ms 2
partition-slot-based-rate-limit-slots 2
Password 162, 163, 165, 166
pattern 217, 218
Payload 202
PAYLOAD_TEXT 202
PaymentRequestsById 213
periodFrom 139, 140
periodTo 139, 140
Phone Number 70, 76, 79, 289
PHONE_NUMBER 70, 76, 79
phoneNumber 289
phoneNumberNameStatus 320
phoneNumberQuality 320
pid 20, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 218, 219, 220, 221, 223, 224, 225, 226, 227, 228, 231, 234, 237, 240, 264, 265, 266, 268, 269, 270, 272
pipeline 130, 131, 132, 133
pipeline_name 130, 131, 132, 133
pipelineID 131, 132
Pipelines 133
PipelineStages 133
planInfo_currencyCode 188
planInfo_nextBillingAt 188
planInfo_planPeriod 188
planInfo_planType 190, 192
planInfo_price 188
planInfo_subAccounts 188
planPeriod 189
pre-request-delay-ms 2
previewText 152, 156
price 142, 145, 146, 148, 189
price[eq] 148
price[gt] 148
price[gte] 148
price[lt] 148
price[lte] 148
price[ne] 148
Primary Audience 70, 76, 79
PRIMARY_AUDIENCE 70, 76, 79
processId 214
ProductById 145
productId 231
ProductInfo_Categories 146
Products 148
provider 136
publicDescription 228, 231, 234, 237
publicImage 226, 228, 231, 234, 237
publicName 228, 231, 234, 237

- Q -

quantity 189
quantityWithOverages 189

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

- R -

Range 277, 291
rawUnsafeHtml 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
READERS 70, 76, 79
readPercentage 319
Reason 274, 289, 294
Reason Code 283
Reason Message 283
reason_code 283
reason_message 283
receivedAt 173, 174, 175
receivedFrom 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
recipient 172, 173, 174, 175
Recipients 152, 156, 250
recipients_excludedLists 312, 314, 315, 317
recipients_includedLists 312, 314, 315, 317
recipients_segments 312, 314, 315, 317
recipients_type 312, 314, 315, 317
records 120
records_RecordsAssociations 121
Recurring 152, 156
redeemPerConsumer 228, 231, 234, 237
Reference 213
Rejected 291, 292
rejectedAt 28
rejectReason 28
Relay Data Port 19
Relay Data Relay 19
Relay Data Username 19
Relay Enabled 19
relay_data_port 19
relay_data_relay 19
relay_data_userName 19
relay_enabled 19
Remaining 189
remainingCoupons 117, 119
remainingCouponsAlert 117, 119
remainingDaysAlert 117, 119
Replied 291, 292
Reply 289
replyTo 152, 156, 280, 281
Reply-to 152, 156, 280, 281
replyTo_email 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
replyTo_name 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
requested-maximum-usable-partitions 2
requested-page-size 2
Requests 277, 278, 291, 292
requests-parallel-max 2
Response Cache Maximum Age (sec) 202
Result BLOB 202
Result Date Time 202
Result Number 202
Result Text 202
RESULT_BLOB 202
RESULT_DATE_TIME_UTC 202
RESULT_NUMBER 202
RESULT_TEXT 202
Return Bounce 152, 156
returnBounce 152, 156
revenue 139, 141, 142
rewardConfigs_attribution 228, 231, 234, 237
rewardConfigs_code 228, 231, 234, 237
rewardConfigs_value 228, 231, 234, 237
rewardId 221, 225, 240, 266
rewardLimitId 228
rid 228, 231, 234, 237
rule_condition_op 228, 231, 234, 237
rule_createdAt 228, 231, 234, 237
rule_description 228, 231, 234, 237
rule_event_name 228, 231, 234, 237
rule_event_source 228, 231, 234, 237
rule_isInternal 228, 231, 234, 237
rule_loyaltyProgramId 228, 231, 234, 237
rule_loyaltyVersionId 228, 231, 234, 237
rule_meta 228, 231, 234, 237
rule_name 228, 231, 234, 237
rule_ruleId 228, 231, 234, 237
rule_ruleType 228, 231, 234, 237
rule_updatedAt 228, 231, 234, 237
Run without Actions 202

- S -

s3Original 145, 146, 148
s3ThumbAnalytics 145, 146, 148
s3ThumbEditor 145, 146, 148
Save HTTP Disk Cache 202
Save HTTP Memory Cache 202
Schedule Date 152, 156, 245, 246, 248, 249, 250, 275, 307, 309, 310, 312, 314, 315, 317, 319
scheduledAt 152, 156, 245, 246, 248, 249, 250, 275, 307, 309, 310, 312, 314, 315, 317, 319
ScheduledEmailById 275
search 148, 165, 166
segmentId 76
segmentName 93

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

segments 93, 307, 309
Send at Best Time 152, 156
sendAtBestTime 152, 156
Sender 172, 173, 174, 175, 245, 246, 248, 249, 250
Sender Email 283
Sender Email Address 152, 156, 280, 281
Sender ID 152, 156, 280, 281
Sender IP Addresses 242
Sender Name 152, 156, 280, 281
sender_email 152, 156, 280, 281
sender_id 152, 156, 280, 281
sender_name 152, 156, 280, 281
senderEmail 283
senderId 241
SenderIpAddresses 242
senderNumber 294, 307, 309, 310, 312, 314, 315, 317
Senders 244
Sendinblue 1, 17, 19, 20, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 68, 70, 72, 74, 75, 76, 79, 81, 82, 83, 85, 86, 87, 89, 91, 92, 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 117, 119, 120, 121, 123, 124, 125, 126, 128, 129, 130, 131, 132, 133, 134, 135, 136, 138, 139, 140, 141, 142, 143, 144, 145, 146, 148, 151, 152, 156, 160, 161, 162, 163, 165, 166, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 204, 205, 206, 207, 209, 210, 211, 212, 213, 214, 216, 217, 218, 219, 220, 221, 223, 224, 225, 226, 227, 228, 231, 234, 237, 240, 241, 242, 244, 245, 246, 248, 249, 250, 252, 253, 255, 256, 257, 259, 260, 261, 262, 263, 264, 265, 266, 268, 269, 270, 272, 273, 274, 275, 277, 278, 280, 281, 283, 284, 286, 287, 288, 289, 291, 292, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 305, 306, 307, 309, 310, 312, 314, 315, 317, 319, 320, 321
sendingLimit 320
Sent Date 152, 156, 245, 246, 248, 249, 250
sentDate 152, 156, 245, 246, 248, 249, 250
Share Link 152, 156
Shared URL 160
Shared URL Template by Campaign ID 160
SharedTemplateUrlByCampaignId 160
sharedUrl 160
shareLink 152, 156
short_url_enabled 252
simulate-http-400-errors 2
simulate-http-400-errors-percentage 2
simulate-http-401-errors 2
simulate-http-401-errors-percentage 2
simulate-http-403-errors 2
simulate-http-403-errors-percentage 2
simulate-http-408-errors 2
simulate-http-408-errors-percentage 2
simulate-http-429-errors 2
simulate-http-429-errors-percentage 2
simulate-http-500-errors 2
simulate-http-500-errors-percentage 2
simulate-http-502-errors 2
simulate-http-502-errors-percentage 2
simulate-http-503-errors 2
simulate-http-503-errors-percentage 2
simulate-http-504-errors 2
simulate-http-504-errors-percentage 2
simulate-http-522-errors 2
simulate-http-522-errors-percentage 2
simulate-http-524-errors 2
simulate-http-524-errors-percentage 2
simulate-http-protocol-errors 2
simulate-http-protocol-errors-percentage 2
simulate-http-timeout-errors 2
simulate-http-timeout-errors-percentage 2
size 107, 112, 168, 170
skipped 291, 292
sku 142, 145, 146, 148
slidingSchedule 23, 228
slot-based-rate-limit-length-ms 2
slot-based-rate-limit-slots 2
SMS 70, 76, 79
SMS Aggregated Transactions Report 291
SMS Campaign by ID 249
SMS Campaigns 250
SMS Events 289
SMS is Blacklisted 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
smsBlacklisted 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
SmsCampaign_RecipientsExclusionLists 245
SmsCampaign_RecipientsLists 246
SmsCampaign_Tags 248
SmsCampaignById 249
SmsCampaigns 250
SmsEvents 289
SMSTemplates 252
SMTP Reports 277, 278
SMTP Template by ID 280
SMTP Templates 281
SmtpReports 277
SmtpReports2 278
SmtpTemplateById 280
SmtpTemplates 281
softBounces 277, 278, 291, 292

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Soft-bounces 277, 278, 291, 292
sort_field 217
sortBy 33, 34, 36, 117, 123, 124, 126, 253, 255, 256, 257
sortByField 148
sortField 22, 24, 25, 26, 28, 29, 30, 225, 240
source 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 321
sourceMessageId 93, 96, 98, 100, 103, 105, 107, 110, 112, 115
Spam Reports 277, 278
spamReports 277, 278
Split Rule 152, 156
splitRule 152, 156
standardize-identifiers 2
standardize-identifiers-casing 2
Start Date 202, 226, 228, 231, 234, 237
startDate 17, 43, 44, 45, 46, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 68, 70, 72, 74, 75, 81, 82, 83, 85, 87, 89, 156, 161, 165, 166, 172, 226, 228, 231, 234, 237, 250, 274, 275, 277, 278, 283, 286, 288, 289, 291, 292, 294, 319, 321
state 217, 218, 226, 227
Statistics 152, 156, 245, 246, 248, 249, 250
statistics_clicks_Version_A 151, 299
statistics_clicks_Version_B 151, 299
statistics_complaints_Version_A 151, 299
statistics_complaints_Version_B 151, 299
statistics_hardBounces_Version_A 151, 299
statistics_hardBounces_Version_B 151, 299
statistics_openers_Version_A 151, 299
statistics_openers_Version_B 151, 299
statistics_softBounces_Version_A 151, 299
statistics_softBounces_Version_B 151, 299
statistics_unsubscribed_Version_A 151, 299
statistics_unsubscribed_Version_B 151, 299
stats_appleMppOpens 89, 301
stats_clickers 89, 301
stats_complaints 89, 301
stats_deferred 89, 301
stats_delivered 89, 301, 307, 309, 310, 312, 314, 315, 317, 319
stats_estimatedViews 89, 301
stats_hardBounces 89, 301
stats_listId 89, 301
stats_notSent 307, 309, 310, 312, 314, 315, 317, 319
stats_opensRate 89, 301
stats_read 307, 309, 310, 312, 314, 315, 317, 319
stats_returnBounce 89, 301
stats_sent 89, 301, 307, 309, 310, 312, 314, 315, 317, 319
stats_softBounces 89, 301
stats_trackableViews 89, 301
stats_trackableViewsRate 89, 301
stats_uniqueClicks 89, 301
stats_uniqueViews 89, 301
stats_unsubscribe 307, 309, 310, 312, 314, 315, 317, 319
stats_unsubscriptions 89, 301
stats_viewed 89, 301
Status 28, 30, 152, 156, 179, 180, 181, 182, 183, 184, 185, 186, 187, 193, 194, 195, 196, 197, 198, 199, 200, 201, 213, 214, 216, 245, 246, 248, 249, 250, 275, 295, 296, 297, 298, 321
stock 145, 146, 148
SubAccountGroups 202
subAccounts 189
Subject 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 152, 156, 173, 174, 175, 274, 280, 281, 284, 286, 287, 288, 299
Subject A 152, 156
Subject B 152, 156
subjectA 152, 156
subjectB 152, 156
Subscription Account 19
SubscriptionAccount 19
SubscriptionBalances 31
SubscriptionBalances_Balance 32
subscriptionGeneratorId 217, 218
subscriptionPoolId 217, 218
subtractBalanceDefinitionId 228, 231, 234, 237
subtractBalanceStrategy 228, 231, 234, 237
subtractBalanceValue 228, 231, 234, 237
subtractTotalBalance 228, 231, 234, 237
Succesful 202
SUCCESSFUL 202
swagger-specification-allow-fallback 2
swagger-specification-download-tries 2
swagger-specification-file 2
swagger-specification-http-disk-cache-max-age-sec 2
swagger-specification-url 2

- T -

tag 152, 156, 274, 277, 278, 280, 281, 289, 291, 292
tags 274, 289
taskTypeId 253, 255, 256, 257, 259, 260, 261, 262
Template ID 274, 284, 286, 287, 288, 319
template_button_type 307, 309, 310, 317
template_category 307, 309, 310, 317
template_contains_button 307, 309, 310, 317

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

template_display_header 307, 309, 310, 317
template_header_type 307, 309, 310, 317
template_hide_footer 307, 309, 310, 317
template_language 307, 309, 310, 317
template_name 307, 309, 310, 317
templateId 274, 280, 284, 286, 287, 288, 319
templateStatus 281
Test Sent 152, 156, 280, 281
testSent 152, 156, 280, 281
Text 34, 36, 37, 38, 41, 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 123, 124, 128, 129, 146, 193, 194, 195, 196, 198, 200, 201, 204, 205, 206, 207, 209, 210, 211, 212, 234, 253, 255, 256, 259, 260, 261, 270, 272, 273, 286, 297, 305, 307, 309, 310, 314
text_2 204, 205, 206, 209, 210, 211
TierGroupByPidAndGid 269
TierGroupItemsByPid 270
TierGroupOrdersByPidAndGid 272
tierId 223, 265, 266, 268
Time 284
Timeout (sec) 202
TIMEOUT_SEC 202
timezone 188, 189
title 263
To Field 152, 156, 280, 281
toField 152, 156, 280, 281
Token 162, 163, 165, 166
Total Blacklisted 86, 87, 89, 91, 92, 300, 301
Total Subscribers 86, 87, 89, 91, 92, 300, 301
total_pages 135, 136
totalAttribution 228, 231, 234, 237
totalBlacklisted 86, 87, 89, 91, 92, 300, 301
totalCount 226, 227
totalCoupons 117, 119
totalRedeem 228, 231, 234, 237
totals_averageBasket 139, 140
totals_ordersCount 139, 140
totals_revenue 139, 140
totalSubscribers 86, 87, 89, 91, 92, 300, 301
totp-secret 2
TransacBlockedContacts 283
TransacEmailContent_Events 284
TransacEmailsList_TransactionalEmailsTags 286
Transaction Blocked Contacts 283
Transaction ID 202
TRANSACTION_ID 202
TransactionAggregatedSmsReports 291
transactional 188
TransactionalSmsReports 292
TransactionEmailContentByUuuid 287
TransactionEmailsByEmail 288
transactionType 23, 28, 30
triggerId 228, 231, 234, 237
Type 41, 42, 93, 96, 98, 100, 103, 105, 107, 110, 112, 115, 152, 156, 175, 179, 180, 181, 182, 183, 184, 185, 186, 187, 228, 302, 303, 305, 306, 307, 309, 310, 314, 321

- U -

unicode_sms 252
uniqueClicks 277, 278
uniqueOpens 277, 278
uniqueSubscribers 86, 87, 89, 91, 92, 300, 301
unit 20, 22, 225, 228, 231, 234, 237
unitValue 189
Unsubscribed 277, 278, 291, 292
Unsubscribed List IDs 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
unsubscribeInstruction 245, 246, 248, 249, 250
updatedAt 20, 22, 23, 25, 93, 120, 121, 204, 205, 206, 207, 209, 210, 211, 212, 217, 218, 220, 221, 223, 225, 226, 228, 231, 234, 237, 252, 253, 255, 256, 257, 259, 260, 261, 262, 264, 265, 266, 268, 269, 270, 272
updatedAt_2 228, 265, 266
upgradeSchedule_durationModifier 264, 269, 270, 272
upgradeSchedule_durationUnit 264, 269, 270, 272
upgradeSchedule_durationValue 264, 269, 270, 272
upgradeSchedule_scheduledDate 264, 269, 270, 272
upgradeSchedule_strategy 264, 269, 270, 272
upgradeStrategy 264, 269, 270, 272
URL 72, 136, 138, 142, 145, 146, 148, 162, 163, 165, 166, 202, 302, 303, 305, 306
Use HTTP Disk Cache 202
Use HTTP Memory Cache 202
use-batch-insert 2
used 189
usedOverages 189
use-http-disk-cache 2
use-http-disk-cache-read 2
use-http-disk-cache-write 2
use-http-memory-cache 2
use-http-memory-cache-read 2
use-http-memory-cache-write 2
user_agent 17
user_email 17
user_id 19
user_ip 17
username 162, 163, 165, 166
UserPermission_Privileges 296
UserPermission_PrivilegesPermissions 297
UserPermissionsByEmail 298

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

utmCampaignValue 152, 156
utmID 152, 156
utmMedium 152, 156
utmSource 152, 156
UUID 162, 163, 172, 173, 174, 175, 284, 286, 287, 288

- V -

validFrom 221, 225
Value 23, 25, 32, 40, 42, 125, 163, 166, 225, 228, 231, 234, 237, 248
value_2 231
valueStr 42
valueType 228, 231, 234, 237
verified 134, 136
version 20, 22, 23, 226, 227, 228, 231, 234, 237, 264, 265, 266, 268, 269, 270, 272
visitorId 93, 96, 98, 100, 103, 105, 107, 110, 112, 115

- W -

Webhook by ID 303
Webhook_Headers 302
WebhookById 303
WebhookEventsByWebhookId 305
webhookId 302, 303, 305
Webhooks 306
Weight 241, 242
WHATSAPP 70, 76, 79
WhatsApp Campaign by ID 317
WhatsApp Campaigns 319
WhatsApp Templates 321
whatsappBlacklisted 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 70, 76, 79
whatsappBusinessAccountId 320
whatsappBusinessAccountStatus 320
WhatsAppCampaign_RecipientsExcludedLists 307
WhatsAppCampaign_RecipientsIncludedLists 309
WhatsAppCampaign_RecipientsSegments 310
WhatsAppCampaign_TemplateBody_variables 312
WhatsAppCampaign_TemplateComponents 314
WhatsAppCampaign_TemplateHeader_variables 315
WhatsAppCampaignById 317
WhatsAppCampaigns 319
WhatsAppConfigurations 320
WhatsappEventReports 294
WhatsAppTemplates 321
Winner Criteria 152, 156
Winner Delay 152, 156
winnerCriteria 152, 156
winnerDelay 152, 156
Winning Criteria 151, 299
Winning Subject Line 151, 299
Winning Version 151, 299
Winning Version Rate 151, 299
winningCriteria 151, 299
winningSubjectLine 151, 299
winningVersion 151, 299
winningVersionRate 151, 299

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.



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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.
24.0.92
29/06/2026, 19:49
ISBN 978-94-6302-000-8: