Zoom 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 Zoom API 1

2 SQL Driver Attributes for Zoom API 2

3 Schema: Accounts 15

3.1 Tables 15
3.1.1 Accounts 15
3.1.2 AccountsByAccountId 16
3.1.3 AccountsByAccountId_DomainsManagedDomains 17
3.1.4 AccountsByAccountIdManagedDomains 18
3.1.5 AccountsByAccountIdSettings 19
3.1.6 deleteAccountsByAccountId 24
3.1.7 patchAccountsByAccountIdOptions 25
3.1.8 patchAccountsByAccountIdSettings 26
3.1.9 postAccounts 27

4 Schema: Billing 28

4.1 Tables 28
4.1.1 AccountsByAccountId_Plan_large_meetingPlans 28
4.1.2 AccountsByAccountId_Plan_webinarPlans 30
4.1.3 AccountsByAccountIdBilling 32
4.1.4 AccountsByAccountIdPlans 33
4.1.5 patchAccountsByAccountIdBilling 34
4.1.6 postAccountsByAccountId_Plan_large_meetingPlans 35
4.1.7 postAccountsByAccountId_Plan_webinarPlans 37
4.1.8 postAccountsByAccountIdPlans 39
4.1.9 postAccountsByAccountIdPlansAddons 40
4.1.10 putAccountsByAccountIdPlansAddons 41
4.1.11 putAccountsByAccountIdPlansBase 42

5 Schema: CloudRecording 44

5.1 Tables 44
5.1.1 deleteMeetingsByMeetingIdRecordings 44
5.1.2 deleteMeetingsByMeetingIdRecordingsRecordingId 45
5.1.3 MeetingsByMeetingIdRecordings 46
5.1.4 MeetingsByMeetingIdRecordingsSettings 47
5.1.5 patchMeetingsByMeetingIdRecordingsSettings 48
5.1.6 putMeetingsByMeetingIdRecordingsRecordingIdStatus 49
5.1.7 putMeetingsByMeetingIdRecordingsStatus 50
5.1.8 UsersByUserId_MeetingsRecordings 51
5.1.9 UsersByUserIdRecordings 53

6 Schema: Dashboards 55

6.1 Tables 55
6.1.1 Metrics_Crc_ports_usageCrc_ports_hour_usageCrc 55
6.1.2 Metrics_MeetingsMeetings 56
6.1.3 Metrics_UsersIm 58
6.1.4 Metrics_WebinarsWebinars 60
6.1.5 MetricsCrc 62
6.1.6 MetricsIm 63

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

6.1.7 MetricsMeetings ... 64
6.1.8 MetricsMeetingsByMeetingId ... 65
6.1.9 MetricsMeetingsByMeetingIdParticipants ... 67
6.1.10 MetricsMeetingsByMeetingIdParticipants_ParticipantsDetailsSharing ... 69
6.1.11 MetricsMeetingsByMeetingIdParticipantsParticipantIdQos ... 70
6.1.12 MetricsMeetingsByMeetingIdParticipantsQos ... 72
6.1.13 MetricsMeetingsByMeetingIdParticipantsSharing ... 74
6.1.14 MetricsWebinars ... 75
6.1.15 MetricsWebinarsByWebinarId ... 77
6.1.16 MetricsWebinarsByWebinarIdParticipants ... 78
6.1.17 MetricsWebinarsByWebinarIdParticipants_ParticipantsDetailsSharing ... 80
6.1.18 MetricsWebinarsByWebinarIdParticipantsParticipantIdQos ... 81
6.1.19 MetricsWebinarsByWebinarIdParticipantsQos ... 83
6.1.20 MetricsWebinarsByWebinarIdParticipantsSharing ... 85
6.1.21 MetricsZoomrooms ... 86
6.1.22 MetricsZoomrooms_Past_meetingsMeetingsByZoomroomId ... 88
6.1.23 MetricsZoomroomsByZoomroomId ... 90

7 Schema: Devices 92

7.1 Tables ... 92
7.1.1 deleteH323DevicesByDeviceId ... 92
7.1.2 H323Devices ... 93
7.1.3 patchH323DevicesByDeviceId ... 93
7.1.4 postH323Devices ... 94

8 Schema: Groups 95

8.1 Tables ... 95
8.1.1 deleteGroupsByGroupId ... 95
8.1.2 deleteGroupsByGroupIdMembersMemberId ... 96
8.1.3 Groups ... 97
8.1.4 Groups_Groups ... 98
8.1.5 GroupsByGroupId ... 98
8.1.6 GroupsByGroupIdMembers ... 99
8.1.7 patchGroupsByGroupId ... 101
8.1.8 postGroups ... 102
8.1.9 postGroupsByGroupIdMembers ... 103

9 Schema: IMChat 104

9.1 Tables ... 104
9.1.1 ImChat_SessionsSessions ... 104
9.1.2 ImChatSessions ... 105
9.1.3 ImChatSessions_MessagesBySessionId ... 107
9.1.4 ImChatSessionsBySessionId ... 108

10 Schema: IMGroups 110

10.1 Tables ... 110
10.1.1 deleteImGroupsByGroupId ... 110
10.1.2 deleteImGroupsByGroupIdMembersMemberId ... 111
10.1.3 ImGroups ... 112
10.1.4 ImGroupsByGroupId ... 112
10.1.5 ImGroupsByGroupIdMembers ... 113
10.1.6 patchImGroupsByGroupId ... 115
10.1.7 postImGroups ... 116
10.1.8 postImGroupsByGroupIdMembers ... 117

11 Schema: Meetings 118

11.1 Tables ... 118

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

11.1.1 deleteMeetingsByMeetingId ... 118
11.1.2 deleteMeetingsByMeetingIdPollsPollId ... 119
11.1.3 MeetingsByMeetingId ... 120
11.1.4 MeetingsByMeetingIdInvitation ... 122
11.1.5 MeetingsByMeetingIdPolls ... 123
11.1.6 MeetingsByMeetingIdPollsPollId ... 124
11.1.7 MeetingsByMeetingIdRegistrants ... 125
11.1.8 PastMeetingsByMeetingIdInstances ... 126
11.1.9 PastMeetingsByMeetingUUID ... 127
11.1.10 PastMeetingsByMeetingUUIDParticipants ... 128
11.1.11 patchMeetingsByMeetingId ... 130
11.1.12 patchMeetingsByMeetingIdLivestream ... 131
11.1.13 patchMeetingsByMeetingIdLivestreamStatus ... 132
11.1.14 postMeetingsByMeetingIdPolls ... 133
11.1.15 postMeetingsByMeetingIdRegistrants ... 134
11.1.16 postUsersByUserIdMeetings ... 135
11.1.17 putMeetingsByMeetingIdPollsPollId ... 137
11.1.18 putMeetingsByMeetingIdRegistrantsStatus ... 138
11.1.19 putMeetingsByMeetingIdStatus ... 139
11.1.20 UsersByUserIdMeetings ... 140

12 Schema: Native 141

12.1 Tables ... 141
12.1.1 NATIVEPLATFORMSCALARREQUESTS: Zoom Native Platform Scalar Requests ... 141

13 Schema: PAC 143

13.1 Tables ... 143
13.1.1 UsersByUserId_Tsp_accountsDedicated_dial_in_numberPac ... 143
13.1.2 UsersByUserId_Tsp_accountsGlobal_dial_in_numbersPac ... 144
13.1.3 UsersByUserIdPac ... 145

14 Schema: Reports 146

14.1 Tables ... 146
14.1.1 Report_DatesDaily ... 146
14.1.2 Report_Telephony_usageTelephone ... 148
14.1.3 Report_UsersUsers ... 149
14.1.4 ReportCloudRecording ... 151
14.1.5 ReportDaily ... 152
14.1.6 ReportMeetings_Tracking_fieldsByMeetingId ... 153
14.1.7 ReportMeetingsByMeetingId ... 154
14.1.8 ReportMeetingsByMeetingId_QuestionsPolls ... 156
14.1.9 ReportMeetingsByMeetingId_QuestionsQuestion_detailsPolls ... 157
14.1.10 ReportMeetingsByMeetingIdParticipants ... 158
14.1.11 ReportMeetingsByMeetingIdPolls ... 159
14.1.12 ReportTelephone ... 160
14.1.13 ReportUsers ... 161
14.1.14 ReportUsersByUserId_MeetingsMeetings ... 163
14.1.15 ReportUsersByUserIdMeetings ... 165
14.1.16 ReportWebinars_Tracking_fieldsByWebinarId ... 166
14.1.17 ReportWebinarsByWebinarId ... 167
14.1.18 ReportWebinarsByWebinarId_QuestionsPolls ... 169
14.1.19 ReportWebinarsByWebinarId_QuestionsQa ... 170
14.1.20 ReportWebinarsByWebinarId_QuestionsQuestion_detailsPolls ... 171
14.1.21 ReportWebinarsByWebinarId_QuestionsQuestion_detailsQa ... 172
14.1.22 ReportWebinarsByWebinarIdParticipants ... 173
14.1.23 ReportWebinarsByWebinarIdPolls ... 175
14.1.24 ReportWebinarsByWebinarIdQa ... 176

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

15 Schema: TrackingField 177

15.1 Tables ... 177
15.1.1 deleteV2TrackingFieldsByFieldId ... 177
15.1.2 patchV2TrackingFieldsByFieldId ... 178
15.1.3 postV2TrackingFields ... 179
15.1.4 V2TrackingFields ... 180
15.1.5 V2TrackingFieldsByFieldId ... 180

16 Schema: TSP 181

16.1 Tables ... 181
16.1.1 deleteUsersByUserIdTspTspId ... 181
16.1.2 patchTsp ... 182
16.1.3 patchUsersByUserIdTspTspId ... 183
16.1.4 postUsersByUserId_Dial_in_numbersTsp ... 184
16.1.5 postUsersByUserIdTsp ... 186
16.1.6 Tsp ... 187
16.1.7 Tsp_Dial_in_numbers ... 187
16.1.8 UsersByUserId_Tsp_accountsDial_in_numbersTsp ... 188
16.1.9 UsersByUserIdTsp ... 189
16.1.10 UsersByUserIdTsp_Dial_in_numbersTspId ... 190
16.1.11 UsersByUserIdTspTspId ... 191

17 Schema: Users 192

17.1 Tables ... 192
17.1.1 deleteUsersByUserId ... 192
17.1.2 deleteUsersByUserIdAssistants ... 194
17.1.3 deleteUsersByUserIdAssistantsAssistantId ... 195
17.1.4 deleteUsersByUserIdSchedulers ... 196
17.1.5 deleteUsersByUserIdSchedulersSchedulerId ... 197
17.1.6 deleteUsersByUserIdToken ... 198
17.1.7 patchUsersByUserId ... 199
17.1.8 patchUsersByUserIdSettings ... 200
17.1.9 postUsers ... 201
17.1.10 postUsersByUserIdAssistants ... 201
17.1.11 postUsersByUserIdPicture ... 202
17.1.12 putUsersByUserIdEmail ... 203
17.1.13 putUsersByUserIdPassw ord ... 204
17.1.14 putUsersByUserIdStatus ... 205
17.1.15 Users ... 206
17.1.16 Users_Group_idsByUserId ... 207
17.1.17 Users_Im_group_idsByUserId ... 209
17.1.18 Users_UsersGroup_ids ... 210
17.1.19 Users_UsersIm_group_ids ... 212
17.1.20 UsersByUserId ... 213
17.1.21 UsersByUserIdAssistants ... 215
17.1.22 UsersByUserIdPermissions ... 216
17.1.23 UsersByUserIdSchedulers ... 217
17.1.24 UsersByUserIdSettings ... 218
17.1.25 UsersByUserIdToken ... 220
17.1.26 UsersEmail ... 221
17.1.27 UsersVanityName ... 222
17.1.28 UsersZpk ... 223

18 Schema: Webhooks 224

18.1 Tables ... 224
18.1.1 deleteWebhooksByWebhookId ... 224

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

18.1.2 patchWebhooksByWebhookId ... 225
18.1.3 patchWebhooksOptions ... 226
18.1.4 postWebhooks ... 227
18.1.5 Webhooks ... 229
18.1.6 WebhooksByWebhookId ... 229

19 Schema: Webinars 230

19.1 Tables ... 230
19.1.1 deleteWebinarsByWebinarId ... 230
19.1.2 deleteWebinarsByWebinarIdPanelists ... 231
19.1.3 deleteWebinarsByWebinarIdPanelistsPanelistId ... 232
19.1.4 deleteWebinarsByWebinarIdPollsPollId ... 233
19.1.5 PastWebinarsByWebinarIdInstances ... 234
19.1.6 patchWebinarsByWebinarId ... 235
19.1.7 postUsersByUserIdWebinars ... 236
19.1.8 postWebinarsByWebinarIdPanelists ... 238
19.1.9 postWebinarsByWebinarIdPolls ... 239
19.1.10 postWebinarsByWebinarIdRegistrants ... 240
19.1.11 putWebinarsByWebinarIdPollsPollId ... 241
19.1.12 putWebinarsByWebinarIdRegistrantsStatus ... 242
19.1.13 putWebinarsByWebinarIdStatus ... 243
19.1.14 UsersByUserIdWebinars ... 244
19.1.15 WebinarsByWebinarId ... 246
19.1.16 WebinarsByWebinarIdPanelists ... 247
19.1.17 WebinarsByWebinarIdPolls ... 248
19.1.18 WebinarsByWebinarIdPollsPollId ... 249
19.1.19 WebinarsByWebinarIdRegistrants ... 250

Index 253

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

1 SQL Driver for Zoom API

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

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

Zoom.

The Zoom driver covers 194 tables and 1433 columns.

Zoom API Clients

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

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

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

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

Monitor API Calls

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

Specifications

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

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

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

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

The catalog name is used to compose the full qualified name of an object like a table or view. The schema name is used to compose the full qualified name of an object like a table or view. On Zoom the comparison of two texts is case sensitive by default.

Changes and bug fixes on the Zoom SQL driver can be found in the release notes. There is currently no specific section on the Invantive forums for Zoom. Please reach out to other users of Zoom by leaving a question or contact request.

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

Alias: `zoom`

Recommended alias: `zom`

More technical documentation as provided by the supplier of the Zoom API on the native API-connection used can be found at https://marketplace.zoom.us/docs/api-reference/zoom-api.

General documentation on Zoom is available at https://zoom.com/

Updated: 16-06-2022 17:21 using Invantive SQL version 22.1.46-BETA+3385.

2 SQL Driver Attributes for Zoom API

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

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

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

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

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

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

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 Zoom driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Drivers File	Set from Log On
add-odata-mandatory-filters	Whether to automatically add OData filters deemed necessary by the platform.	OData	False	✓	✓	✓	
analysis-enforce-row-uniqueness	Use for analysis only! Enforce rows to be unique.	Shared	False	✓	✓	✓	
api-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		✓		✓	✓
api-scope	The scope to request an OAuth token for.	OData		✓		✓	
api-token-url	The token URI is the OAuth2 endpoint to exchange tokens.	OData		✓		✓	
api-url	URL to access the API.	OData		✓		✓	
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting	Shared	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
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting	Shared	250	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when OData server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries OData server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-408-request-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 408.		10	✓	✓	✓	
download-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 408.		10000	✓	✓	✓	
download-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 408.		300000	✓	✓	✓	
download-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 408.		2	✓	✓	✓	
download-error-422-bad-request-max-tries	Maximum number of tries when OData server reports unprocessable entity during retrieval of data.		30	✓	✓	✓	
download-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when OData server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
download-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when OData server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
download-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries OData server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
download-error-429-too-many-requests-max-tries	Maximum number of tries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	

(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-429-too-many-requests-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10000	✓	✓	✓	
download-error-429-too-many-requests-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		300000	✓	✓	✓	
download-error-429-too-many-requests-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
download-error-502-server-unavailable-max-tries	Maximum number of tries when OData server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
download-error-502-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when OData server reports a bad gateway during retrieval of data.		10000	✓	✓	✓	
download-error-502-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when OData server reports that a bad gateway during retrieval of data.		300000	✓	✓	✓	
download-error-502-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries OData server reports a bad gateway during retrieval of data.		2	✓	✓	✓	
download-error-503-server-unavailable-max-tries	Maximum number of tries when OData server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
download-error-503-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
download-error-503-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when OData server reports that the API server is unavailable during retrieval of data.		300000	✓	✓	✓	
download-error-503-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries OData server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-504-gateway-timeout-max-tries	Maximum number of tries when the website reports a gateway timeout.		10	✓	✓	✓	
download-error-504-gateway-	Initial sleep in milliseconds between retries when the website reports a gateway timeout.		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
timeout-sleep-initial-ms							
dow nload-error-504-gatew ay-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a gatew ay timeout.		300000	✓	✓	✓	
dow nload-error-504-gatew ay-timeout-sleep-multi-plicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a gatew ay timeout.		2	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-max-tries	Maximum number of tries w hen the w ebsite reports a HTTP status 590.		10	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 590.		10000	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the w ebsite reports a HTTP status 590.		300000	✓	✓	✓	
dow nload-error-590-netw ork-con-nect-timeout-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the w ebsite reports a HTTP status 590.		2	✓	✓	✓	
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-argument-exception-max-tries	Maximum number of tries w hen an argument exception is returned w hen dow nloading a blob.		10	✓	✓	✓	
dow nload-error-argument-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-argument-exception-	Maximum sleep in milliseconds betw een retries w hen an argument		300000	✓	✓	✓	

(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
sleep-max-ms	exception is returned w hen dow n-loading a blob.						
dow nload-error-ar-gument-exception-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen an argument exception is returned w hen dow n-loading a blob.		2	✓	✓	✓	
dow nload-error-in-ternet-dow n-max-tries	Maximum number of tries w hen the Internet connection seems dow n during retrieval of data.		10	✓	✓	✓	
dow nload-error-in-ternet-dow n-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen the Internet connection seems dow n during retrieval of data.		10000	✓	✓	✓	
dow nload-error-in-ternet-dow n-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen the Internet connection seems dow n during re-trieval of data.		300000	✓	✓	✓	
dow nload-error-in-ternet-dow n-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen the Internet connection seems dow n during re-trieval of data.		2	✓	✓	✓	
dow nload-error-io-exception-max-tries	Maximum number of tries w hen a netw ork I/O connection failure oc-curs during retrieval of data.		10	✓	✓	✓	
dow nload-error-io-exception-sleep-ini-tial-ms	Initial sleep in milliseconds betw een retries w hen a netw ork I/O connec-tion failure occurs during retrieval of data.		10000	✓	✓	✓	
dow nload-error-io-exception-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen a netw ork I/O connection failure occurs during re-trieval of data.		300000	✓	✓	✓	
dow nload-error-io-exception-sleep-mul-tiplicator	Multiplication factor for sleep betw een retries w hen a netw ork I/O connection failure occurs during re-trieval of data.		2	✓	✓	✓	
dow nload-error-json-exception-max-tries	Maximum number of tries w hen an invalid JSON body is returned.		3	✓	✓	✓	
dow nload-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds betw een retries w hen an invalid JSON body is returned.		1000	✓	✓	✓	
dow nload-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds betw een retries w hen an invalid JSON body is returned.		10000	✓	✓	✓	
dow nload-error-json-exception-sleep-multiplicator	Multiplication factor for sleep betw een retries w hen an invalid JSON body is returned.		2	✓	✓	✓	
dow nload-error-other-exception-max-tries	Maximum number of tries w hen an unqualified error occurs during re-trieval of data.		3	✓	✓	✓	

(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-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-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	✓	✓	✓	

(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-web-timeout-max-tries	Maximum number of tries when the connection reports a timeout.		10	✓	✓	✓	
download-error-web-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports a timeout.		1000	✓	✓	✓	
download-error-web-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports a timeout.		30000	✓	✓	✓	
download-error-web-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports a timeout.		2	✓	✓	✓	
download-error-web-unauthorized-max-tries	Maximum number of tries when the connection reports an unauthorized error.		1	✓	✓	✓	
download-error-web-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports an unauthorized error.		10000	✓	✓	✓	
download-error-web-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports an unauthorized error.		300000	✓	✓	✓	
download-error-web-unauthorized-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports an unauthorized error.		2	✓	✓	✓	
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.	Shared	5	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.	Shared	C:\Users\gle3.WS212\Invantive\Cache\http\gle3\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.	Shared	False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.	Shared	2592000	✓	✓	✓	
http-get-timeout-max-ms	HTTP GET maximum timeout on retry (ms).		300000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		60000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.	OData	5	✓	✓	✓	

(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
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.	OData	14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		300000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		300000	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
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.		10	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial 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-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		300000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		300000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
invantive-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-correct-invalid-date	Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are removed. Otherwise they are replaced by 01-01-1753.	SQL Engine V1	False	✓	✓	✓	
invantive-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invantive-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invantive-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register from last activation.	Shared		✓	✓	✓	

(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 an error occurred.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when successful.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
maximum-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	The maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.	OData	2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	10000	✓	✓	✓	
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.	OData	2	✓	✓	✓	
page-size-rows	Number of rows to retrieve per page.	Zoom	100	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in ms across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online	Shared		✓	✓	✓	

(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
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-429-errors	Simulate HTTP 429 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors when exchanging results with the OData endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors when exchanging results with the OData endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors when exchanging results		0	✓	✓	✓	

(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
	w ith the OData endpoint.						
simulate-http-pro-tocol-errors	Simulate HTTP protocol errors w hen exchanging results w ith the OData endpoint.		False	✓	✓	✓	
simulate-http-pro-tocol-errors-per-centage	Percentage of simulated HTTP pro-tocol errors w hen exchanging res-ults w ith the OData endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors w hen exchanging results w ith the OData endpoint.		False	✓	✓	✓	
simulate-http-timeout-errors-per-centage	Percentage of simulated HTTP timeout errors w hen exchanging results w ith the OData endpoint.		0	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in ms 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-identifi-ers	Rew rite all identifiers to the pre-ferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
standardize-identifi-ers-casing	Rew rite all identifiers to the recom-mended standard platform-specific casing w hen changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	
sw agger-specifica-tion-file	The Sw agger file path, such as C: \temp\sw agger.json.			✓	✓	✓	✓
sw agger-specifica-tion-http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of Sw agger specification data in the HTTP disk cache.		86400	✓	✓	✓	
sw agger-specifica-tion-url	The Sw agger URL such as ht-tps://example.org/rest/sw agger.json.			✓	✓	✓	✓
use-batch-insert	Whether to use batch insert.	OData	True	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answ er the current query.	Shared	True	✓	✓	✓	
use-http-disk-cache-w rite	Whether to memorize HTTP re-sponses on disk.	Shared	True	✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answ er the current query.	OData	True	✓	✓	✓	
use-http-memory-cache-w rite	Whether to memorize HTTP re-sponses from previous queries for use by future queries.	OData	True	✓	✓	✓	

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

3 Schema: Accounts

3.1 Tables

3.1.1 Accounts

List sub accountsList all the sub accounts under the master account

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts`

Insert Zoom API URL: `/accounts`

Update Zoom API URL: `/accounts`

Delete Zoom API URL: `/accounts`

Field Selection Method: NotRequired

Base Path: `accounts[*]`

Select Zoom API Operation: `get /accounts`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call

Table Function Columns

The columns of the table function `Accounts` 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
account_name	string		☐	Account name
account_type	string		☐	Account type
created_at	datetime		☐	Account creation date/time
id	guid		☐	Account ID
ow ner_email	string		☐	Account ow ner email
seats	int64		☐	Account seats
subscription_end_time	datetime		☐	Account subscription end date/time
subscription_start_time	datetime		☐	Account subscription start date/time

3.1.2 AccountsByAccountId

Retrieve a sub accountRetrieve a sub account under the master account. <aside>Your account must be a master account and have this privilege to read sub accounts. Zoom only assigns this privilege to trusted partners</aside>.

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}`

Insert Zoom API URL: `/accounts/{accountId}`

Update Zoom API URL: `/accounts/{accountId}`

Delete Zoom API URL: `/accounts/{accountId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /accounts/{accountId}`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
created_at	datetime		☐	Account creation date/time
id	string		☐	Account ID
options_meeting_connectors	string		☐	Meeting Connector, multiple values separated by comma
options_pay_mode	string		☐	Payee
options_room_connectors	string		☐	Virtual Room Connector, multiple value separated by comma
options_share_mc	boolean		☐	Enable Share Meeting Connector
options_share_rc	boolean		☐	Enable Share Virtual Room Connector
owner_email	string		☐	Account Owner email
owner_id	string		☐	Account Owner ID
vanity_url	string		☐	Account Vanity URL

3.1.3 AccountsByAccountId_DomainsManagedDomains

Retrieve a sub account's managed domainsRetrieve a sub account's managed domains under the master account

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}/managed_domains`

Insert Zoom API URL: `/accounts/{accountId}/managed_domains`

Update Zoom API URL: `/accounts/{accountId}/managed_domains`

Delete Zoom API URL: `/accounts/{accountId}/managed_domains`

Field Selection Method: NotRequired

Base Path: `domains[*]`

Select Zoom API Operation: `get /accounts/{accountId}/managed_domains`

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
domain	string		☐	Domain Name
status	string		☐	Domain Status
total_records	int64		☐	Total records

3.1.4 AccountsByAccountIdManagedDomains

Retrieve a sub account's managed domainsRetrieve a sub account's managed domains under the master account

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}/managed_domains`

Insert Zoom API URL: `/accounts/{accountId}/managed_domains`

Update Zoom API URL: `/accounts/{accountId}/managed_domains`

Delete Zoom API URL: `/accounts/{accountId}/managed_domains`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /accounts/{accountId}/managed_domains`

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
total_records	int64		□	Total records

3.1.5 AccountsByAccountIdSettings

Retrieve a sub account's settingsRetrieve a sub account's settings under the master account

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}/settings`

Insert Zoom API URL: `/accounts/{accountId}/settings`

Update Zoom API URL: `/accounts/{accountId}/settings`

Delete Zoom API URL: `/accounts/{accountId}/settings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /accounts/{accountId}/settings`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountsByAccountIdSettings. 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 treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
email_notification_alternat-ive_host_reminder	boolean		☐	Notify w hen an alternative host is set or removed from a meeting
email_notification_cancel_meet-ing_reminder	boolean		☐	Notify host and participants w hen the meeting is cancelled
email_notification_cloud_record-ing_avaliable_reminder	boolean		☐	Notify host w hen cloud record-ing is available
email_notification_jbh_reminder	boolean		☐	Notify host w hen participants join the meeting before them
email_notification_low _host_coun-t_reminder	boolean		☐	Notify w hen host licenses are running low
feature_meeting_capacity	int64		☐	Set the maximum number of par-ticipants this user can have in a single meeting
in_meeting_alert_guest_join	boolean		☐	Identify guest participants in the meeting/w ebinar
in_meeting_allow _live_streaming	boolean		☐	Allow live streaming
in_meeting_allow _show _-zoom_w indow s	boolean		☐	Show Zoom Desktop application w hen sharing screen
in_meeting_annotation	boolean		☐	Allow participants to use annota-tion tools to add information to shared screens
in_meeting_anonymous_ques-tion_answ er	boolean		☐	Allow Anonymous Q&A in We-binar
in_meeting_attendee_on_hold	boolean		☐	Allow hosts to temporarily re-move an attendee from the meet-ing
in_meeting_attention_tracking	boolean		☐	Lets the host see an indicator in the participant panel if a meet-ing/w ebinar attendee does not

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

Name	Data Type	Label	Required	Documentation
				have Zoom in focus during screen sharing
in_meeting_auto_answ er	boolean		☐	Enable users to see and add contacts to 'auto-answ er group' in the contact list on chat. Any call from members of this group w ill be automatically answ ered.
in_meeting_auto_saving_chat	boolean		☐	Automatically save all in-meeting chats so that hosts do not need to manually save the text of the chat after the meeting starts
in_meeting_breakout_room	boolean		☐	Allow host to split meeting participants into separate, smaller rooms
in_meeting_chat	boolean		☐	Allow meeting participants to send a message visible to all participants
in_meeting_closed_caption	boolean		☐	Allow host to type closed captions or assign a participant/third party device to add closed captions
in_meeting_co_host	boolean		☐	Allow the host to add co-hosts
in_meeting_custom_live_streaming	boolean		☐	Custom live streaming
in_meeting_custom_service_instructions	string		☐	Custom service instructions
in_meeting_dscp_audio	int64		☐	DSCP Audio
in_meeting_dscp_marking	boolean		☐	DSCP marking
in_meeting_dscp_video	int64		☐	DSCP Video
in_meeting_e2e_encryption	boolean		☐	Require that all meetings are encrypted using AES
in_meeting_far_end_camera_control	boolean		☐	Allow another user to take control of your camera during a meeting
in_meeting_feedback	boolean		☐	Add a Feedback tab to the Window s Settings or Mac Preferences dialog, and also enable users to provide feedback to Zoom at the end of the meeting
in_meeting_file_transfer	boolean		☐	Hosts and participants can send files through the in-meeting chat
in_meeting_group_hd	boolean		☐	Activate higher quality video for host and participants. (This w ill use more bandw idth.)
in_meeting_original_audio	boolean		☐	Allow users to select original sound in their client settings
in_meeting_p2p_connetion	boolean		☐	Peer to Peer connection w hile only 2 people are in a meeting
in_meeting_p2p_ports	boolean		☐	P2P listening ports range
in_meeting_polling	boolean		☐	Add 'Polls' to the meeting controls.
in_meeting_ports_range	string		☐	Listening ports range, separated by comma (ex 55,56). The ports

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

Name	Data Type	Label	Required	Documentation
				range must be betw een 1 to 65535.
in_meeting_post_meeting_feed-back	boolean		☐	Display a thumbs up/dow n sur-vey at the end of each meeting
in_meeting_private_chat	boolean		☐	Allow meeting participants to send a private 1:1 message to another participants
in_meeting_remote_control	boolean		☐	Allow users to request remote control
in_meeting_screen_sharing	boolean		☐	Allow screen sharing
in_meeting_sending_default_e-mail_invites	boolean		☐	Only show default email w hen sending email invites
in_meeting_show _meeting_con-trol_toolbar	boolean		☐	Alw ays show meeting control toolbar
in_meeting_stereo_audio	boolean		☐	Allow users to select stereo au-dio in their client settings
in_meeting_use_html_format_email	boolean		☐	Use HTML format email for Outlook plugin
in_meeting_virtual_background	boolean		☐	Allow users to replace their background w ith any selected image. Choose or upload an im-age in the Zoom Desktop applica-tion settings.
in_meeting_w atermark	boolean		☐	Add w atermark w hen view ing shared screen
in_meeting_w ebinar_ques-tion_answ er	boolean		☐	Q&A in w ebinar
in_meeting_w hiteboard	boolean		☐	Allow participants to share a w hiteboard that includes annota-tion tools
in_meeting_w orkplace_by_face-book	boolean		☐	Workplace by facebook
integration_box	boolean		☐	Enables users w ho join a meet-ing from their mobile device to share content from their Box ac-count
integration_dropbox	boolean		☐	Enables users w ho join a meet-ing from their mobile device to share content from their Dropbox account
integration_google_calendar	boolean		☐	Enables meetings to be sched-uled using Google Calendars
integration_google_drive	boolean		☐	Enables users w ho join a meet-ing from their mobile device to share content from their Google Drive
integration_kubi	boolean		☐	Enables users to control a con-nected Kubi device from w ithin a Zoom meeting
integration_microsoft_one_drive	boolean		☐	Enables users w ho join a meet-ing from their mobile device to share content from their Mi-crosoft OneDrive account

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

Name	Data Type	Label	Required	Documentation
recording_account_user_access_recording	boolean		☐	Cloud recordings are only accessible to account members. People outside of your organization cannot open links that provide access to cloud recordings.
recording_auto_delete_cmr_days	int64		☐	When `auto_delete_cmr` is 'true' this value will set the number of days before auto deletion of cloud recordings
recording_auto_delete_cmr	boolean		☐	Allow Zoom to automatically delete recordings permanently after a specified number of days
recording_auto_recording	string		☐	Record meetings automatically as they start
recording_cloud_recording_download_host	boolean		☐	Only the host can download cloud recordings
recording_cloud_recording_download	boolean		☐	Cloud Recording Downloads
recording_cloud_recording	boolean		☐	Allow hosts to record and save the meeting in the cloud
recording_local_recording	boolean		☐	Allow hosts and participants to record the meeting to a local file
recording_record_audio_file	boolean		☐	Record an audio only file
recording_record_gallery_view	boolean		☐	Record gallery view with shared screen
recording_record_speaker_view	boolean		☐	Record active speaker with shared screen
recording_recording_audio_transcript	boolean		☐	Automatically transcribe the audio of the meeting or webinar to the cloud
recording_save_chat_text	boolean		☐	Save chat text from the meeting
recording_show_timestamp	boolean		☐	Add a timestamp to the recording
schedule_meting_audio_type	string		☐	Determine how participants can join the audio portion of the meeting
schedule_meting_enforce_login_domains	string		☐	Only signed-in users with a specified domains
schedule_meting_enforce_login_with_domains	boolean		☐	Only signed-in users with a specific domain can join meetings
schedule_meting_enforce_login	boolean		☐	Only signed-in (Zoom users) users can join meetings
schedule_meting_force_pmi_jbh_password	boolean		☐	Require a password for Personal Meetings if attendees can join before host
schedule_meting_host_video	boolean		☐	Start meetings with host video on
schedule_meting_join_before_host	boolean		☐	Allow participants to join the meeting before the host arrives
schedule_meting_not_store_meeting_topic	boolean		☐	Always display "Zoom Meeting" as the meeting topic

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

Name	Data Type	Label	Required	Documentation
schedule_meting_participant_video	boolean		☐	Start meetings w ith participant video on. Participants can change this during the meeting.
security_admin_change_name_pic	boolean		☐	Only account administrators can change user's username and picture
security_hide_billing_info	boolean		☐	Hide billing information
security_import_photos_-from_devices	boolean		☐	Allow importing of photos from photo library on the user's device
telephony_audio_conference_info	string		☐	3rd party audio conference info
telephony_third_party_audio	boolean		☐	Users can join the meeting using the existing 3rd party audio con-figuration
zoom_room-s_auto_start_stop_sched-uled_meetings	boolean		☐	Automatic start/stop for sched-uled meetings
zoom_rooms_cmr_for_in-stant_meeting	boolean		☐	Cloud recording for instant meet-ings
zoom_rooms_force_private_meet-ing	boolean		☐	Transform all meetings to private
zoom_rooms_hide_host_informa-tion	boolean		☐	Hide host and meeting ID from private meetings
zoom_rooms_list_meeting-s_w ith_calendar	boolean		☐	Display meeting list w ith calendar integration
zoom_rooms_start_airplay_manu-ally	boolean		☐	Start AirPlay service manually
zoom_rooms_ultrasonic	boolean		☐	Automatic direct sharing using ultrasonic proximity signal
zoom_rooms_upcoming_meet-ing_alert	boolean		☐	Upcoming meeting alert
zoom_rooms_w eekly_system_re-start	boolean		☐	Weekly system restart
zoom_rooms_zr_post_meet-ing_feedback	boolean		☐	Zoom Room post meeting feed-back

3.1.6 deleteAccountsByAccountId

Disassociate an accountDisassociate a sub account from the master account. This will leave the account intact but the sub account will not longer be associated with the master account.

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}`

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

Insert Zoom API URL: `/accounts/{accountId}`

Update Zoom API URL: `/accounts/{accountId}`

Delete Zoom API URL: `/accounts/{accountId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /accounts/{accountId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

3.1.7 patchAccountsByAccountIdOptions

Update a sub account's optionsUpdate a sub account's options under the master account

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}/options`

Insert Zoom API URL: `/accounts/{accountId}/options`

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

Update Zoom API URL: `/accounts/{accountId}/options`

Delete Zoom API URL: `/accounts/{accountId}/options`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /accounts/{accountId}/options`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

3.1.8 patchAccountsByAccountIdSettings

Update a sub account's settingsUpdate a sub account's settings under the master account

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts/{accountId}/settings`

Insert Zoom API URL: `/accounts/{accountId}/settings`

Update Zoom API URL: `/accounts/{accountId}/settings`

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

Delete Zoom API URL: `/accounts/{accountId}/settings`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /accounts/{accountId}/settings`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

3.1.9 postAccounts

Create a sub accountCreate a sub account under the master account. <aside>Your account must be a master account and have this privilege to create sub account. Zoom only assigns this privilege to trusted partners. The created user will not receive a confirmation email.</aside>.

Catalog: Zoom

Schema: Accounts

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

Select Zoom API URL: `/accounts`

Insert Zoom API URL: `/accounts`

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

Update Zoom API URL: `/accounts`

Delete Zoom API URL: `/accounts`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /accounts`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Account

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
created_at	string		☐	Account created date time
id	string		☐	Account ID
owner_email	string		☐	Account owner email
owner_id	string		☐	Account Owner ID
RESULT	string		☐	Outcome of operation as single plain text column.

4 Schema: Billing

4.1 Tables

4.1.1 AccountsByAccountId_Plan_large_meetingPlans

Retrieve plan information for a sub accountRetrieve plan information for a sub account under the master account <aside>Only for the sub account which is paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

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

Select Zoom API URL: `/accounts/{accountId}/plans`

Insert Zoom API URL: `/accounts/{accountId}/plans`

Update Zoom API URL: `/accounts/{accountId}/plans`

Delete Zoom API URL: `/accounts/{accountId}/plans`

Field Selection Method: NotRequired

Base Path: `plan_large_meeting[*]`

Select Zoom API Operation: `get /accounts/{accountId}/plans`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
hosts	int64		☐	Account plan number of hosts
plan_audio_callout_countries	string		☐	Call-out countries, multiple value separated by comma
plan_audio_ddi_numbers	int64		☐	Dedicated Dial-In Numbers
plan_audio_premium_countries	string		☐	Premium countries, multiple value separated by comma
plan_audio_tollfree_countries	string		☐	Toll-free countries, multiple value separated by comma
plan_audio_type	string		☐	Additional Audio Conferencing plan type

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

Name	Data Type	Label	Required	Documentation
plan_base_hosts	int64		☑	Account base plan number of hosts. For a Pro Plan, please select a value betw een 1 and 9. For a Business Plan, please select a value betw een 10 and 49. For a Education Plan, please select a value betw een 20 and 149. For a Free Trial Plan, please select a value betw een 1 and 9999.
plan_base_type	string		☑	Account base plan type
plan_recording	string		☐	Additional Cloud Recording Plan
plan_room_connector_hosts	int64		☐	Account plan number of hosts
plan_room_connector_type	string		☐	Account plan type
plan_zoom_rooms_hosts	int64		☐	Account plan number of hosts
plan_zoom_rooms_type	string		☐	Account plan type
type	string		☐	Account plan type

4.1.2 AccountsByAccountId_Plan_webinarPlans

Retrieve plan information for a sub accountRetrieve plan information for a sub account under the master account <aside>Only for the sub account which is paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans`

Insert Zoom API URL: `/accounts/{accountId}/plans`

Update Zoom API URL: `/accounts/{accountId}/plans`

Delete Zoom API URL: `/accounts/{accountId}/plans`

Field Selection Method: NotRequired

Base Path: `plan_webinar[*]`

Select Zoom API Operation: `get /accounts/{accountId}/plans`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
hosts	int64		☐	Account plan number of hosts
plan_audio_callout_countries	string		☐	Call-out countries, multiple value separated by comma
plan_audio_ddi_numbers	int64		☐	Dedicated Dial-In Numbers
plan_audio_premium_countries	string		☐	Premium countries, multiple value separated by comma
plan_audio_tollfree_countries	string		☐	Toll-free countries, multiple value separated by comma
plan_audio_type	string		☐	Additional Audio Conferencing plan type
plan_base_hosts	int64		☑	Account base plan number of hosts. For a Pro Plan, please select a value betw een 1 and 9. For a Business Plan, please select a value betw een 10 and 49. For a Education Plan, please select a value betw een 20 and 149. For a Free Trial Plan, please select a value betw een 1 and 9999.
plan_base_type	string		☑	Account base plan type
plan_recording	string		☐	Additional Cloud Recording Plan
plan_room_connector_hosts	int64		☐	Account plan number of hosts
plan_room_connector_type	string		☐	Account plan type
plan_zoom_rooms_hosts	int64		☐	Account plan number of hosts
plan_zoom_rooms_type	string		☐	Account plan type

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

Name	Data Type	Label	Required	Documentation
type	string		☐	Account plan type

4.1.3 AccountsByAccountIdBilling

Retrieve billing information for a sub accountRetrieve billing information for a sub account under the master account

Catalog: Zoom

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/billing`

Insert Zoom API URL: `/accounts/{accountId}/billing`

Update Zoom API URL: `/accounts/{accountId}/billing`

Delete Zoom API URL: `/accounts/{accountId}/billing`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /accounts/{accountId}/billing`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID

Table Function Columns

The columns of the table function `AccountsByAccountIdBilling` 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
address	string		☑	Billing Contact's address
apt	string		☐	Billing Contact's apartment/suite
city	string		☑	Billing Contact's city
country	string		☑	Billing Contact's country
email	string		☑	Billing Contact's email address
first_name	string		☑	Billing Contact's first name
last_name	string		☑	Billing Contact's last name
phone_number	string		☑	Billing Contact's phone number
state	string		☑	Billing Contact's state
zip	string		☑	Billing Contact's zip/postal code

4.1.4 AccountsByAccountIdPlans

Retrieve plan information for a sub accountRetrieve plan information for a sub account under the master account <aside>Only for the sub account which is paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans`

Insert Zoom API URL: `/accounts/{accountId}/plans`

Update Zoom API URL: `/accounts/{accountId}/plans`

Delete Zoom API URL: `/accounts/{accountId}/plans`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /accounts/{accountId}/plans`

Parameters of Table Function

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

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

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

(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
accountId	string	☑		The account ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
plan_audio_callout_countries	string		☐	Call-out countries, multiple value separated by comma
plan_audio_ddi_numbers	int64		☐	Dedicated Dial-In Numbers
plan_audio_premium_countries	string		☐	Premium countries, multiple value separated by comma
plan_audio_tollfree_countries	string		☐	Toll-free countries, multiple value separated by comma
plan_audio_type	string		☐	Additional Audio Conferencing plan type
plan_base_hosts	int64		☑	Account base plan number of hosts. For a Pro Plan, please select a value betw een 1 and 9. For a Business Plan, please select a value betw een 10 and 49. For a Education Plan, please select a value betw een 20 and 149. For a Free Trial Plan, please select a value betw een 1 and 9999.
plan_base_type	string		☑	Account base plan type
plan_recording	string		☐	Additional Cloud Recording Plan
plan_room_connector_hosts	int64		☐	Account plan number of hosts
plan_room_connector_type	string		☐	Account plan type
plan_zoom_rooms_hosts	int64		☐	Account plan number of hosts
plan_zoom_rooms_type	string		☐	Account plan type

4.1.5 patchAccountsByAccountIdBilling

Update billing information for a sub accountUpdate billing information for a sub account under the master account <aside>Only for the sub account which is a paid account and paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

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

Select Zoom API URL: `/accounts/{accountId}/billing`

Insert Zoom API URL: `/accounts/{accountId}/billing`

Update Zoom API URL: `/accounts/{accountId}/billing`

Delete Zoom API URL: `/accounts/{accountId}/billing`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /accounts/{accountId}/billing`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

4.1.6 postAccountsByAccountId_Plan_large_meetingPlans

Subscribe plans for a sub accountSubscribe plans for a sub account of the master account <aside>Can only subscribe plans for the sub account which is a free account and paid by master account</aside>

Catalog: Zoom

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

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans`

Insert Zoom API URL: `/accounts/{accountId}/plans`

Update Zoom API URL: `/accounts/{accountId}/plans`

Delete Zoom API URL: `/accounts/{accountId}/plans`

Field Selection Method: NotRequired

Base Path: `plan_large_meeting[*]`

Select Zoom API Operation: `post /accounts/{accountId}/plans`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
hosts	int64		☐	Account plan number of hosts
plan_audio_callout_countries	string		☐	Call-out countries, multiple value separated by comma
plan_audio_ddi_numbers	int64		☐	Dedicated Dial-In Numbers
plan_audio_premium_countries	string		☐	Premium countries, multiple value separated by comma

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

Name	Data Type	Label	Required	Documentation
plan_audio_tollfree_countries	string		☐	Toll-free countries, multiple value separated by comma
plan_audio_type	string		☐	Additional Audio Conferencing plan type
plan_base_hosts	int64		☑	Account base plan number of hosts. For a Pro Plan, please select a value between 1 and 9. For a Business Plan, please select a value between 10 and 49. For a Education Plan, please select a value between 20 and 149. For a Free Trial Plan, please select a value between 1 and 9999.
plan_base_type	string		☑	Account base plan type
plan_recording	string		☐	Additional Cloud Recording Plan
plan_room_connector_hosts	int64		☐	Account plan number of hosts
plan_room_connector_type	string		☐	Account plan type
plan_zoom_rooms_hosts	int64		☐	Account plan number of hosts
plan_zoom_rooms_type	string		☐	Account plan type
RESULT	string		☐	Outcome of operation as single plain text column.
type	string		☐	Account plan type

4.1.7 postAccountsByAccountId_Plan_webinarPlans

Subscribe plans for a sub accountSubscribe plans for a sub account of the master account <aside>Can only subscribe plans for the sub account which is a free account and paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans`

Insert Zoom API URL: `/accounts/{accountId}/plans`

Update Zoom API URL: `/accounts/{accountId}/plans`

Delete Zoom API URL: `/accounts/{accountId}/plans`

Field Selection Method: NotRequired

Base Path: `plan_webinar[*]`

Select Zoom API Operation: `post /accounts/{accountId}/plans`

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
hosts	int64		☐	Account plan number of hosts
plan_audio_callout_countries	string		☐	Call-out countries, multiple value separated by comma
plan_audio_ddi_numbers	int64		☐	Dedicated Dial-In Numbers
plan_audio_premium_countries	string		☐	Premium countries, multiple value separated by comma
plan_audio_tollfree_countries	string		☐	Toll-free countries, multiple value separated by comma
plan_audio_type	string		☐	Additional Audio Conferencing plan type
plan_base_hosts	int64		☑	Account base plan number of hosts. For a Pro Plan, please select a value betw een 1 and 9. For a Business Plan, please select a value betw een 10 and 49. For a Education Plan, please select a value betw een 20 and 149. For a Free Trial Plan, please select a value betw een 1 and 9999.
plan_base_type	string		☑	Account base plan type
plan_recording	string		☐	Additional Cloud Recording Plan
plan_room_connector_hosts	int64		☐	Account plan number of hosts

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

Name	Data Type	Label	Required	Documentation
plan_room_connector_type	string		☐	Account plan type
plan_zoom_rooms_hosts	int64		☐	Account plan number of hosts
plan_zoom_rooms_type	string		☐	Account plan type
RESULT	string		☐	Outcome of operation as single plain text column.
type	string		☐	Account plan type

4.1.8 postAccountsByAccountIdPlans

Subscribe plans for a sub accountSubscribe plans for a sub account of the master account <aside>Can only subscribe plans for the sub account which is a free account and paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans`

Insert Zoom API URL: `/accounts/{accountId}/plans`

Update Zoom API URL: `/accounts/{accountId}/plans`

Delete Zoom API URL: `/accounts/{accountId}/plans`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /accounts/{accountId}/plans`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
plan_audio_callout_countries	string		☐	Call-out countries, multiple value separated by comma
plan_audio_ddi_numbers	int64		☐	Dedicated Dial-In Numbers
plan_audio_premium_countries	string		☐	Premium countries, multiple value separated by comma
plan_audio_tollfree_countries	string		☐	Toll-free countries, multiple value separated by comma
plan_audio_type	string		☐	Additional Audio Conferencing plan type
plan_base_hosts	int64		☑	Account base plan number of hosts. For a Pro Plan, please select a value betw een 1 and 9. For a Business Plan, please select a value betw een 10 and 49. For a Education Plan, please select a value betw een 20 and 149. For a Free Trial Plan, please select a value betw een 1 and 9999.
plan_base_type	string		☑	Account base plan type
plan_recording	string		☐	Additional Cloud Recording Plan
plan_room_connector_hosts	int64		☐	Account plan number of hosts
plan_room_connector_type	string		☐	Account plan type
plan_zoom_rooms_hosts	int64		☐	Account plan number of hosts
plan_zoom_rooms_type	string		☐	Account plan type
RESULT	string		☐	Outcome of operation as single plain text column.

4.1.9 postAccountsByAccountIdPlansAddons

Add an additional plan for sub accountAdd an additional plan for sub account <aside>Can only add an Additional plan for the sub account which is a paid account and paid by master account</aside>

Catalog: Zoom

Schema: Billing

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

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

Select Zoom API URL: `/accounts/{accountId}/plans/addons`

Insert Zoom API URL: `/accounts/{accountId}/plans/addons`

Update Zoom API URL: `/accounts/{accountId}/plans/addons`

Delete Zoom API URL: `/accounts/{accountId}/plans/addons`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /accounts/{accountId}/plans/addons`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

4.1.10 putAccountsByAccountIdPlansAddons

Update an additional plan for sub accountUpdate an additional plan for sub account<aside>Can only update an Additional plan for the sub account which is a paid account and paid by master account</aside>

Catalog: Zoom

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

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans/addons`

Insert Zoom API URL: `/accounts/{accountId}/plans/addons`

Update Zoom API URL: `/accounts/{accountId}/plans/addons`

Delete Zoom API URL: `/accounts/{accountId}/plans/addons`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /accounts/{accountId}/plans/addons`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

4.1.11 putAccountsByAccountIdPlansBase

Update a base plan for a sub accountUpdate a base plan for a sub account <aside>Can only update a base plan for the sub account which is a paid account and paid by master account</aside>

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

Catalog: Zoom

Schema: Billing

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

Select Zoom API URL: `/accounts/{accountId}/plans/base`

Insert Zoom API URL: `/accounts/{accountId}/plans/base`

Update Zoom API URL: `/accounts/{accountId}/plans/base`

Delete Zoom API URL: `/accounts/{accountId}/plans/base`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /accounts/{accountId}/plans/base`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
accountId	string	☑		The account ID
body	string	☑		

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

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

5 Schema: CloudRecording

5.1 Tables

5.1.1 deleteMeetingsByMeetingIdRecordings

Delete a meeting's recordingsDelete a meeting's recordings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings`

Insert Zoom API URL: `/meetings/{meetingId}/recordings`

Update Zoom API URL: `/meetings/{meetingId}/recordings`

Delete Zoom API URL: `/meetings/{meetingId}/recordings`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /meetings/{meetingId}/recordings`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
action	string	☐	trash	The recording delete action (Values: trash, delete)
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, will take the last meeting instance.

Table Function Columns

The columns of the table function `deleteMeetingsByMeetingIdRecordings` 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
RESULT	string		☐	Outcome of operation as single plain text column.

5.1.2 deleteMeetingsByMeetingIdRecordingsRecordingId

Delete one meeting recording fileDelete one meeting recording file

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}`

Insert Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}`

Update Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}`

Delete Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /meetings/{meetingId}/recordings/{recordingId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
action	string	☐	trash	The recording delete action (Values: trash, delete)
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, will take the last meeting instance.
recordingId	string	☑		The recording ID

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

5.1.3 MeetingsByMeetingIdRecordings

Retrieve a meeting's all recordingsRetrieve a meeting's all recordings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings`

Insert Zoom API URL: `/meetings/{meetingId}/recordings`

Update Zoom API URL: `/meetings/{meetingId}/recordings`

Delete Zoom API URL: `/meetings/{meetingId}/recordings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}/recordings`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
account_id	string		☐	ID of the user account
duration	int64		☐	Meeting duration
host_id	string		☐	ID of the user set as host of meeting
id	string		☐	Meeting ID, also know as meeting number
recording_count	string		☐	Recording count
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
total_size	string		☐	Total size
uuid	string		☐	Meeting unique ID

5.1.4 MeetingsByMeetingIdRecordingsSettings

Retrieve a meeting recording's settingsRetrieve a meeting recording's settings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Insert Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Update Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Delete Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}/recordings/settings`

Parameters of Table Function

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

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

(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
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
approval_type	int64		☐	Approval type
on_demand	boolean		☐	Registration required
passw ord	string		☐	Passw ord protect
send_email_to_host	boolean		☐	Send an email to host w hen someone registers
share_recording	string		☐	Determine if the meeting record-ing is shared
show _social_share_buttons	boolean		☐	Show social share buttons on registration page
view er_dow nload	boolean		☐	Host video

5.1.5 patchMeetingsByMeetingIdRecordingsSettings

Update a meeting recording's settingsUpdate a meeting recording's settings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Insert Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Update Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Delete Zoom API URL: `/meetings/{meetingId}/recordings/settings`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /meetings/{meetingId}/recordings/set-tings`

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Meeting recording Settings
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

5.1.6 putMeetingsByMeetingIdRecordingsRecordingIdStatus

Recover a single recordingRecover a single recording

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}/status`

Insert Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}/status`

Update Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}/status`

Delete Zoom API URL: `/meetings/{meetingId}/recordings/{recordingId}/status`

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

Field Selection Method: NotRequired

Select Zoom API Operation: `put /meetings/{meetingId}/recordings/{recordingId}/status`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.
recordingId	string	☑		The recording ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		□	Outcome of operation as single plain text column.

5.1.7 putMeetingsByMeetingIdRecordingsStatus

Recover a meeting's recordingsRecover a meeting's recordings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/meetings/{meetingId}/recordings/status`

Insert Zoom API URL: `/meetings/{meetingId}/recordings/status`

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

Update Zoom API URL: `/meetings/{meetingId}/recordings/status`

Delete Zoom API URL: `/meetings/{meetingId}/recordings/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /meetings/{meetingId}/recordings/status`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

5.1.8 UsersByUserId_MeetingsRecordings

List all the recordingsList all the recordings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/users/{userId}/recordings`

Insert Zoom API URL: `/users/{userId}/recordings`

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

Update Zoom API URL: `/users/{userId}/recordings`

Delete Zoom API URL: `/users/{userId}/recordings`

Field Selection Method: NotRequired

Base Path: `meetings[*]`

Select Zoom API Operation: `get /users/{userId}/recordings`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
mc	string	☐	false	Query mc
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
trash	boolean	☐	False	Query trash
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
account_id	string		☐	ID of the user account
duration	int64		☐	Meeting duration
from	datetime		☐	Start Date,

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

Name	Data Type	Label	Required	Documentation
host_id	string		☐	ID of the user set as host of meeting
id	string		☐	Meeting ID, also know as meeting number
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
recording_count	string		☐	Recording count
start_time	datetime		☐	Meeting start time
to	datetime		☐	End Date
topic	string		☐	Meeting topic
total_records	int64		☐	The number of all records available across pages
total_size	string		☐	Total size
uuid	string		☐	Meeting unique ID

5.1.9 UsersByUserIdRecordings

List all the recordingsList all the recordings

Catalog: Zoom

Schema: CloudRecording

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

Select Zoom API URL: `/users/{userId}/recordings`

Insert Zoom API URL: `/users/{userId}/recordings`

Update Zoom API URL: `/users/{userId}/recordings`

Delete Zoom API URL: `/users/{userId}/recordings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/{userId}/recordings`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdRecordings. A value must be provided at 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 treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
mc	string	☐	false	Query mc
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
trash	boolean	☐	False	Query trash
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start Date,
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
to	datetime		☐	End Date
total_records	int64		☐	The number of all records available across pages

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

6 Schema: Dashboards

6.1 Tables

6.1.1 Metrics_Crc_ports_usageCrc_ports_hour_usageCrc

Retrieve CRC Port UsageGet CRC Port usage hour by hour for a specified time period <aside class='notice'>We will report a maximum of one month. For example, if "from" is set to "2017-08-05" and "to" is "2017-10-10" we will adjust "from" to "2017-09-10"</aside>.

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/crc`

Insert Zoom API URL: `/metrics/crc`

Update Zoom API URL: `/metrics/crc`

Delete Zoom API URL: `/metrics/crc`

Field Selection Method: NotRequired

Base Path: `crc_ports_usage[*].crc_ports_hour_usage[*]`

Select Zoom API Operation: `get /metrics/crc`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
to	datetime	☑		End Date

Table Function Columns

The columns of the table function `Metrics_Crc_ports_usageCrc_ports_hour_usageCrc` 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
date_time	string		☐	Date and time
hour	string		☐	
max_usage	int64		☐	
total_usage	int64		☐	

6.1.2 Metrics_MeetingsMeetings

List meetingsList live meetings or past meetings for a specified period

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings`

Insert Zoom API URL: `/metrics/meetings`

Update Zoom API URL: `/metrics/meetings`

Delete Zoom API URL: `/metrics/meetings`

Field Selection Method: NotRequired

Base Path: `meetings[*]`

Select Zoom API Operation: `get /metrics/meetings`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the cur-

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

Name	Data Type	Required	Default Value	Documentation
				rent page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned w ithin a single API call
to	datetime	☑		End Date
type	string	☐	live	The meeting type (Values: past, pastOne, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	string		☐	Meeting duration
email	string		☐	User email
end_time	datetime		☐	Meeting end time
from	datetime		☐	Start date for this report
has_3rd_party_audio	boolean		☐	
has_pstn	boolean		☐	
has_recording	boolean		☐	
has_screen_share	boolean		☐	
has_sip	boolean		☐	
has_video	boolean		☐	
has_voip	boolean		☐	
host	string		☐	User display name
id	int64		☐	Meeting ID
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token w ill be returned w henever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned w ithin a single API call.
participants	int64		☐	Meeting participant count
start_time	datetime		☐	Meeting start time
to	datetime		☐	End date for this report
topic	string		☐	Meeting topic
total_records	int64		☐	The number of all records available across pages

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

Name	Data Type	Label	Required	Documentation
user_type	string		☐	User type
uuid	guid		☐	Meeting UUID

6.1.3 Metrics_UsersIm

Retrieve IMRetrieve metrics of Zoom IM

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/im`

Insert Zoom API URL: `/metrics/im`

Update Zoom API URL: `/metrics/im`

Delete Zoom API URL: `/metrics/im`

Field Selection Method: NotRequired

Base Path: `users[*]`

Select Zoom API Operation: `get /metrics/im`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.

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

Name	Data Type	Required	Default Value	Documentation
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
calls_receive	int64		☐	
calls_send	int64		☐	
email	string		☐	User email
emoji_receive	int64		☐	
emoji_send	int64		☐	
files_receive	int64		☐	
files_send	int64		☐	
from	datetime		☐	Start date for this report
group_receive	int64		☐	
group_send	int64		☐	
images_receive	int64		☐	
images_send	int64		☐	
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
to	datetime		☐	End date for this report
total_receive	int64		☐	
total_records	int64		☐	The number of all records available across pages
total_send	int64		☐	
user_id	string		☐	User ID
user_name	string		☐	User display name
videos_receive	int64		☐	
videos_send	int64		☐	
voice_receive	int64		☐	
voice_send	int64		☐	

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

6.1.4 Metrics_WebinarsWebinars

List webinarsList live webinars or past webinars for a specified period

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars`

Insert Zoom API URL: `/metrics/webinars`

Update Zoom API URL: `/metrics/webinars`

Delete Zoom API URL: `/metrics/webinars`

Field Selection Method: NotRequired

Base Path: `webinars[*]`

Select Zoom API Operation: `get /metrics/webinars`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date

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

Name	Data Type	Required	Default Value	Documentation
type	string	☐	live	The webinar type (Values: past, pastOne, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	string		☐	Webinar duration
email	string		☐	User email
end_time	datetime		☐	Webinar end time
from	datetime		☐	Start date for this report
has_3rd_party_audio	boolean		☐	
has_pstn	boolean		☐	
has_recording	boolean		☐	
has_screen_share	boolean		☐	
has_sip	boolean		☐	
has_video	boolean		☐	
has_voip	boolean		☐	
host	string		☐	User display name
id	int64		☐	Webinar ID
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
participants	int64		☐	Webinar participant count
start_time	datetime		☐	Webinar start time
to	datetime		☐	End date for this report
topic	string		☐	Webinar topic
total_records	int64		☐	The number of all records available across pages
user_type	string		☐	User type
uuid	guid		☐	Webinar UUID

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

6.1.5 MetricsCrc

Retrieve CRC Port UsageGet CRC Port usage hour by hour for a specified time period <aside class='notice'>We will report a maximum of one month. For example, if "from" is set to "2017-08-05" and "to" is "2017-10-10" we will adjust "from" to "2017-09-10"</aside>.

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/crc`

Insert Zoom API URL: `/metrics/crc`

Update Zoom API URL: `/metrics/crc`

Delete Zoom API URL: `/metrics/crc`

Field Selection Method: NotRequired

Base Path: `crc_ports_usage[*]`

Select Zoom API Operation: `get /metrics/crc`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
date_time	string		☐	Date and time

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

6.1.6 MetricsIm

Retrieve IMRetrieve metrics of Zoom IM

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/im`

Insert Zoom API URL: `/metrics/im`

Update Zoom API URL: `/metrics/im`

Delete Zoom API URL: `/metrics/im`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/im`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date for this report
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
to	datetime		☐	End date for this report
total_records	int64		☐	The number of all records available across pages

6.1.7 MetricsMeetings

List meetingsList live meetings or past meetings for a specified period

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings`

Insert Zoom API URL: `/metrics/meetings`

Update Zoom API URL: `/metrics/meetings`

Delete Zoom API URL: `/metrics/meetings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/meetings`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
type	string	☐	live	The meeting type (Values: past, pastOne, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date for this report
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
to	datetime		☐	End date for this report
total_records	int64		☐	The number of all records available across pages

6.1.8 MetricsMeetingsByMeetingId

Retrieve meeting detailRetrieve live or past meetings detail

Catalog: Zoom

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

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings/{meetingId}`

Insert Zoom API URL: `/metrics/meetings/{meetingId}`

Update Zoom API URL: `/metrics/meetings/{meetingId}`

Delete Zoom API URL: `/metrics/meetings/{meetingId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/meetings/{meetingId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, will take the last meeting instance.
type	string	☐	live	The meeting type (Values: past, pastOne, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	string		☐	Meeting duration
email	string		☐	User email
end_time	datetime		☐	Meeting end time
has_3rd_party_audio	boolean		☐	
has_pstn	boolean		☐	
has_recording	boolean		☐	

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

Name	Data Type	Label	Required	Documentation
has_screen_share	boolean		☐	
has_sip	boolean		☐	
has_video	boolean		☐	
has_voip	boolean		☐	
host	string		☐	User display name
id	int64		☐	Meeting ID
participants	int64		☐	Meeting participant count
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
user_type	string		☐	User type
uuid	guid		☐	Meeting UUID

6.1.9 MetricsMeetingsByMeetingIdParticipants

Retrieve meeting participantsRetrieve live or past meetings participants

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings/{meetingId}/participants`

Insert Zoom API URL: `/metrics/meetings/{meetingId}/participants`

Update Zoom API URL: `/metrics/meetings/{meetingId}/participants`

Delete Zoom API URL: `/metrics/meetings/{meetingId}/participants`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /metrics/meetings/{meetingId}/participants`

Parameters of Table Function

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

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

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

(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
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token w ill be returned w henever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned w ithin a single API call
type	string	☐	live	The meeting type (Values: past, pastOne, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
connection_type	string		☐	Participant connection type
data_center	string		☐	Participant data center
device	string		☐	Participant device
domain	string		☐	Participant domain
harddisk_id	string		☐	Participant hard disk id
id	guid		☐	Participant UUID
ip_address	string		☐	Participant IP Address
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
location	string		☐	Participant location
mac_addr	string		☐	Participant MAC Address
microphone	string		☐	Participant microphone
netw ork_type	string		☐	Participant netw ork type
pc_name	string		☐	Participant PC name
recording	boolean		☐	Participant record
share_application	boolean		☐	Did participant share application
share_desktop	boolean		☐	Did participant share desktop
share_w hiteboard	boolean		☐	Did participant share w hiteboard
speaker	string		☐	Participant speaker
user_id	string		☐	Participant ID
user_name	string		☐	Participant display name

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

Name	Data Type	Label	Required	Documentation
version	string		☐	Participant version

6.1.10 MetricsMeetingsByMeetingIdParticipants_ParticipantsDetailsSharing

Retrieve sharing/recording details of meeting participantRetrieve sharing/recording details of live or past meetings participant

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Insert Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Update Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Delete Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Field Selection Method: NotRequired

Base Path: `participants[*].details[*]`

Select Zoom API Operation: `get /metrics/meetings/{meetingId}/participants/sharing`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, will take the last meeting instance.
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be re-

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

Name	Data Type	Required	Default Value	Documentation
				turned w henever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned w ithin a single API call
type	string	☐	live	The meeting type (Values: past, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
content	string		☐	Type of content shared
end_time	string		☐	End time of sharing
id	string		☐	Participant UUID
start_time	string		☐	Start time of sharing
user_id	string		☐	Participant ID
user_name	string		☐	Participant display name

6.1.11 MetricsMeetingsByMeetingIdParticipantsParticipantIdQos

Retrieve meeting participant QOSRetrieve live or past meetings participant quality of service

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings/{meetingId}/participants/{participantId}/qos`

Insert Zoom API URL: `/metrics/meetings/{meetingId}/participants/{participantId}/qos`

Update Zoom API URL: `/metrics/meetings/{meetingId}/participants/{participantId}/qos`

Delete Zoom API URL: `/metrics/meetings/{meetingId}/participants/{participantId}/qos`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/meetings/{meetingId}/participants/{participantId}/qos`

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.
participantId	string	☑		Participant ID
type	string	☐	live	The meeting type (Values: past, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
device	string		☐	Participant device
domain	string		☐	Participant domain
harddisk_id	string		☐	Participant hard disk id
ip_address	string		☐	Participant IP Address
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
location	string		☐	Participant location
mac_addr	string		☐	Participant MAC Address
pc_name	string		☐	Participant PC name
user_id	guid		☐	Participant ID
user_name	string		☐	Participant display name
user_qos_audio_input_avg_loss	string		☐	Average Loss
user_qos_audio_input_bitrate	string		☐	Bitrate
user_qos_audio_input_jitter	string		☐	Jitter
user_qos_audio_input_latency	string		☐	Latency
user_qos_audio_input_max_loss	string		☐	Max Loss

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

Name	Data Type	Label	Required	Documentation
user_qos_audio_output_avg_loss	string		☐	Average Loss
user_qos_audio_output_bitrate	string		☐	Bitrate
user_qos_audio_output_jitter	string		☐	Jitter
user_qos_audio_output_latency	string		☐	Latency
user_qos_audio_output_max_loss	string		☐	Max Loss
user_qos_cpu_usage_system_max_cpu_usage	string		☐	System Maximum CPU Usage
user_qos_cpu_usage_zoom_avg_cpu_usage	string		☐	Zoom Average CPU Usage
user_qos_cpu_usage_zoom_max_cpu_usage	string		☐	Zoom Maximum CPU Usage
user_qos_cpu_usage_zoom_min_cpu_usage	string		☐	Zoom Minimum CPU Usage
user_qos_date_time	datetime		☐	Datetime of QOS
version	string		☐	Participant version

6.1.12 MetricsMeetingsByMeetingIdParticipantsQos

List meeting participants QOSRetrieve list of live or past meetings participants quality of service

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings/{meetingId}/participants/qos`

Insert Zoom API URL: `/metrics/meetings/{meetingId}/participants/qos`

Update Zoom API URL: `/metrics/meetings/{meetingId}/participants/qos`

Delete Zoom API URL: `/metrics/meetings/{meetingId}/participants/qos`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /metrics/meetings/{meetingId}/participants/qos`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, will take the last meeting instance.
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	1	Number of items returned per page
type	string	☐	live	The meeting type (Values: past, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
device	string		☐	Participant device
domain	string		☐	Participant domain
harddisk_id	string		☐	Participant hard disk id
ip_address	string		☐	Participant IP Address
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
location	string		☐	Participant location
mac_addr	string		☐	Participant MAC Address
pc_name	string		☐	Participant PC name
user_id	guid		☐	Participant ID
user_name	string		☐	Participant display name
user_qos_audio_input_avg_loss	string		☐	Average Loss
user_qos_audio_input_bitrate	string		☐	Bitrate
user_qos_audio_input_jitter	string		☐	Jitter
user_qos_audio_input_latency	string		☐	Latency
user_qos_audio_input_max_loss	string		☐	Max Loss
user_qos_audio_output_avg_loss	string		☐	Average Loss
user_qos_audio_output_bitrate	string		☐	Bitrate

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

Name	Data Type	Label	Required	Documentation
user_qos_audio_output_jitter	string		☐	Jitter
user_qos_audio_output_latency	string		☐	Latency
user_qos_audio_output_max_loss	string		☐	Max Loss
user_qos_cpu_usage_system_max_cpu_usage	string		☐	System Maximum CPU Usage
user_qos_cpu_usage_zoom_avg_cpu_usage	string		☐	Zoom Average CPU Usage
user_qos_cpu_usage_zoom_max_cpu_usage	string		☐	Zoom Maximum CPU Usage
user_qos_cpu_usage_zoom_min_cpu_usage	string		☐	Zoom Minimum CPU Usage
user_qos_date_time	datetime		☐	Datetime of QOS
version	string		☐	Participant version

6.1.13 MetricsMeetingsByMeetingIdParticipantsSharing

Retrieve sharing/recording details of meeting participantRetrieve sharing/recording details of live or past meetings participant

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Insert Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Update Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Delete Zoom API URL: `/metrics/meetings/{meetingId}/participants/sharing`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /metrics/meetings/{meetingId}/participants/sharing`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token w ill be returned w henever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned w ithin a single API call
type	string	☐	live	The meeting type (Values: past, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Participant UUID
user_id	string		☐	Participant ID
user_name	string		☐	Participant display name

6.1.14 MetricsWebinars

List webinarsList live webinars or past webinars for a specified period

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars`

Insert Zoom API URL: `/metrics/webinars`

Update Zoom API URL: `/metrics/webinars`

Delete Zoom API URL: `/metrics/webinars`

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

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/webinars`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
type	string	☐	live	The webinar type (Values: past, pastOne, live)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date for this report
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page

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

Name	Data Type	Label	Required	Documentation
page_size	int64		☐	The number of records returned within a single API call.
to	datetime		☐	End date for this report
total_records	int64		☐	The number of all records available across pages

6.1.15 MetricsWebinarsByWebinarId

Retrieve webinar detailRetrieve live or past webinars detail

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars/{webinarId}`

Insert Zoom API URL: `/metrics/webinars/{webinarId}`

Update Zoom API URL: `/metrics/webinars/{webinarId}`

Delete Zoom API URL: `/metrics/webinars/{webinarId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/webinars/{webinarId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
type	string	☐	live	The webinar type (Values: past, live)
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	string		☐	Webinar duration
email	string		☐	User email
end_time	datetime		☐	Webinar end time
has_3rd_party_audio	boolean		☐	
has_pstn	boolean		☐	
has_recording	boolean		☐	
has_screen_share	boolean		☐	
has_sip	boolean		☐	
has_video	boolean		☐	
has_voip	boolean		☐	
host	string		☐	User display name
id	int64		☐	Webinar ID
participants	int64		☐	Webinar participant count
start_time	datetime		☐	Webinar start time
topic	string		☐	Webinar topic
user_type	string		☐	User type
uuid	guid		☐	Webinar UUID

6.1.16 MetricsWebinarsByWebinarIdParticipants

Retrieve webinar participantsRetrieve live or past webinar participants

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars/{webinarId}/participants`

Insert Zoom API URL: `/metrics/webinars/{webinarId}/participants`

Update Zoom API URL: `/metrics/webinars/{webinarId}/participants`

Delete Zoom API URL: `/metrics/webinars/{webinarId}/participants`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /metrics/webinars/{webinarId}/participants`

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

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

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

Name	Data Type	Required	Default Value	Documentation
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
type	string	☐	live	The webinar type (Values: past, live)
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
connection_type	string		☐	Participant connection type
data_center	string		☐	Participant data center
device	string		☐	Participant device
domain	string		☐	Participant domain
harddisk_id	string		☐	Participant hard disk id
id	guid		☐	Participant UUID
ip_address	string		☐	Participant IP Address
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
location	string		☐	Participant location
mac_addr	string		☐	Participant MAC Address
microphone	string		☐	Participant microphone

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

Name	Data Type	Label	Required	Documentation
network_type	string		☐	Participant network type
pc_name	string		☐	Participant PC name
recording	boolean		☐	Participant record
share_application	boolean		☐	Did participant share application
share_desktop	boolean		☐	Did participant share desktop
share_whiteboard	boolean		☐	Did participant share whiteboard
speaker	string		☐	Participant speaker
user_id	string		☐	Participant ID
user_name	string		☐	Participant display name
version	string		☐	Participant version

6.1.17 MetricsWebinarsByWebinarIdParticipants_ParticipantsDetailsSharing

Retrieve sharing/recording details of webinar participantRetrieve sharing/recording details of live or past webinar participant

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Insert Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Update Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Delete Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Field Selection Method: NotRequired

Base Path: `participants[*].details[*]`

Select Zoom API Operation: `get /metrics/webinars/{webinarId}/participants/sharing`

Parameters of Table Function

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

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

(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
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
type	string	☐	live	The webinar type (Values: past, live)
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
content	string		☐	Type of content shared
end_time	string		☐	End time of sharing
id	string		☐	Participant UUID
start_time	string		☐	Start time of sharing
user_id	string		☐	Participant ID
user_name	string		☐	Participant display name

6.1.18 MetricsWebinarsByWebinarIdParticipantsParticipantIdQos

Retrieve webinar participant QOSRetrieve live or past webinar participant quality of service

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars/{webinarId}/participants/{participantId}/qos`

Insert Zoom API URL: `/metrics/webinars/{webinarId}/participants/{participantId}/qos`

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

Update Zoom API URL: `/metrics/webinars/{webinarId}/participants/{participantId}/qos`

Delete Zoom API URL: `/metrics/webinars/{webinarId}/participants/{participantId}/qos`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/webinars/{webinarId}/participants/{participantId}/qos`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
participantId	string	☑		Participant ID
type	string	☐	live	The webinar type (Values: past, live)
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
device	string		☐	Participant device
domain	string		☐	Participant domain
harddisk_id	string		☐	Participant hard disk id
ip_address	string		☐	Participant IP Address
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
location	string		☐	Participant location
mac_addr	string		☐	Participant MAC Address
pc_name	string		☐	Participant PC name

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

Name	Data Type	Label	Required	Documentation
user_id	guid		☐	Participant ID
user_name	string		☐	Participant display name
user_qos_audio_input_avg_loss	string		☐	Average Loss
user_qos_audio_input_bitrate	string		☐	Bitrate
user_qos_audio_input_jitter	string		☐	Jitter
user_qos_audio_input_latency	string		☐	Latency
user_qos_audio_input_max_loss	string		☐	Max Loss
user_qos_audio_output_avg_loss	string		☐	Average Loss
user_qos_audio_output_bitrate	string		☐	Bitrate
user_qos_audio_output_jitter	string		☐	Jitter
user_qos_audio_output_latency	string		☐	Latency
user_qos_audio_output_max_loss	string		☐	Max Loss
user_qos_cpu_usage_system_max_cpu_usage	string		☐	System Maximum CPU Usage
user_qos_cpu_usage_zoom_avg_cpu_usage	string		☐	Zoom Average CPU Usage
user_qos_cpu_usage_zoom_max_cpu_usage	string		☐	Zoom Maximum CPU Usage
user_qos_cpu_usage_zoom_min_cpu_usage	string		☐	Zoom Minimum CPU Usage
user_qos_date_time	datetime		☐	Datetime of QOS
version	string		☐	Participant version

6.1.19 MetricsWebinarsByWebinarIdParticipantsQos

List webinar participant QOSRetrieve list of live or past webinar participants quality of service

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars/{webinarId}/participants/qos`

Insert Zoom API URL: `/metrics/webinars/{webinarId}/participants/qos`

Update Zoom API URL: `/metrics/webinars/{webinarId}/participants/qos`

Delete Zoom API URL: `/metrics/webinars/{webinarId}/participants/qos`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /metrics/webinars/{webinarId}/participants/qos`

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

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

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

Name	Data Type	Required	Default Value	Documentation
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	1	Number of items returned per page
type	string	☐	live	The webinar type (Values: past, live)
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
device	string		☐	Participant device
domain	string		☐	Participant domain
harddisk_id	string		☐	Participant hard disk id
ip_address	string		☐	Participant IP Address
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
location	string		☐	Participant location
mac_addr	string		☐	Participant MAC Address
pc_name	string		☐	Participant PC name
user_id	guid		☐	Participant ID
user_name	string		☐	Participant display name
user_qos_audio_input_avg_loss	string		☐	Average Loss

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

Name	Data Type	Label	Required	Documentation
user_qos_audio_input_bitrate	string		☐	Bitrate
user_qos_audio_input_jitter	string		☐	Jitter
user_qos_audio_input_latency	string		☐	Latency
user_qos_audio_input_max_loss	string		☐	Max Loss
user_qos_audio_output_avg_loss	string		☐	Average Loss
user_qos_audio_output_bitrate	string		☐	Bitrate
user_qos_audio_output_jitter	string		☐	Jitter
user_qos_audio_output_latency	string		☐	Latency
user_qos_audio_output_max_loss	string		☐	Max Loss
user_qos_cpu_usage_system_max_cpu_usage	string		☐	System Maximum CPU Usage
user_qos_cpu_usage_zoom_avg_cpu_usage	string		☐	Zoom Average CPU Usage
user_qos_cpu_usage_zoom_max_cpu_usage	string		☐	Zoom Maximum CPU Usage
user_qos_cpu_usage_zoom_min_cpu_usage	string		☐	Zoom Minimum CPU Usage
user_qos_date_time	datetime		☐	Datetime of QOS
version	string		☐	Participant version

6.1.20 MetricsWebinarsByWebinarIdParticipantsSharing

Retrieve sharing/recording details of webinar participantRetrieve sharing/recording details of live or past webinar participant

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Insert Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Update Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Delete Zoom API URL: `/metrics/webinars/{webinarId}/participants/sharing`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /metrics/webinars/{webinarId}/participants/sharing`

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

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

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

Name	Data Type	Required	Default Value	Documentation
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
type	string	☐	live	The webinar type (Values: past, live)
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Participant UUID
user_id	string		☐	Participant ID
user_name	string		☐	Participant display name

6.1.21 MetricsZoomrooms

List Zoom RoomsList all zoom rooms on account

Catalog: Zoom

Schema: Dashboards

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

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

Select Zoom API URL: `/metrics/zoomrooms`

Insert Zoom API URL: `/metrics/zoomrooms`

Update Zoom API URL: `/metrics/zoomrooms`

Delete Zoom API URL: `/metrics/zoomrooms`

Field Selection Method: NotRequired

Base Path: `zoom_rooms[*]`

Select Zoom API Operation: `get /metrics/zoomrooms`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
account_type	string		☐	Zoom Room email type
calender_name	string		☐	Zoom Calendar name
camera	string		☐	Zoom Room camera
device_ip	string		☐	Zoom Room device IP
email	string		☐	Zoom Room email
id	string		☐	Zoom Room ID
last_start_time	string		☐	Zoom Room last start time
microphone	string		☐	Zoom Room microphone
room_name	string		☐	Zoom Room name

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

Name	Data Type	Label	Required	Documentation
speaker	string		☐	Zoom Room speaker
status	string		☐	Zoom Room status

6.1.22 MetricsZoomrooms_Past_meetingsMeetingsByZoomroomId

Retrieve Zoom RoomRetrieve zoom room on account

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Insert Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Update Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Delete Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Field Selection Method: NotRequired

Base Path: `past_meetings.meetings[*]`

Select Zoom API Operation: `get /metrics/zoomrooms/{zoomroomId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
zoomroomId	string	☑		The Zoom Room ID

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
account_type	string		☐	Zoom Room email type
calender_name	string		☐	Zoom Calendar name
camera	string		☐	Zoom Room camera
device_ip	string		☐	Zoom Room device IP
duration	string		☐	Meeting duration
email_1	string		☐	User email
email	string		☐	Zoom Room email
end_time	datetime		☐	Meeting end time
from	datetime		☐	Start date for this report
has_3rd_party_audio	boolean		☐	
has_pstn	boolean		☐	
has_recording	boolean		☐	
has_screen_share	boolean		☐	
has_sip	boolean		☐	
has_video	boolean		☐	
has_voip	boolean		☐	
host	string		☐	User display name
id_1	int64		☐	Meeting ID
id	string		☐	Zoom Room ID
last_start_time	string		☐	Zoom Room last start time
live_meeting_duration	string		☐	Meeting duration
live_meeting_email	string		☐	User email
live_meeting_end_time	datetime		☐	Meeting end time
live_meeting_has_3rd_party_audio	boolean		☐	
live_meeting_has_pstn	boolean		☐	
live_meeting_has_recording	boolean		☐	
live_meeting_has_screen_share	boolean		☐	
live_meeting_has_sip	boolean		☐	
live_meeting_has_video	boolean		☐	
live_meeting_has_voip	boolean		☐	
live_meeting_host	string		☐	User display name
live_meeting_id	int64		☐	Meeting ID
live_meeting_participants	int64		☐	Meeting participant count
live_meeting_start_time	datetime		☐	Meeting start time
live_meeting_topic	string		☐	Meeting topic
live_meeting_user_type	string		☐	User type

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

Name	Data Type	Label	Required	Documentation
live_meeting_uuid	guid		☐	Meeting UUID
microphone	string		☐	Zoom Room microphone
participants	int64		☐	Meeting participant count
room_name	string		☐	Zoom Room name
speaker	string		☐	Zoom Room speaker
start_time	datetime		☐	Meeting start time
status	string		☐	Zoom Room status
to	datetime		☐	End date for this report
topic	string		☐	Meeting topic
user_type	string		☐	User type
uuid	guid		☐	Meeting UUID

6.1.23 MetricsZoomroomsByZoomroomId

Retrieve Zoom RoomRetrieve zoom room on account

Catalog: Zoom

Schema: Dashboards

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

Select Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Insert Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Update Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Delete Zoom API URL: `/metrics/zoomrooms/{zoomroomId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /metrics/zoomrooms/{zoomroomId}`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
zoomroomId	string	☑		The Zoom Room ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
account_type	string		☐	Zoom Room email type
calender_name	string		☐	Zoom Calendar name
camera	string		☐	Zoom Room camera
device_ip	string		☐	Zoom Room device IP
email	string		☐	Zoom Room email
id	string		☐	Zoom Room ID
last_start_time	string		☐	Zoom Room last start time
live_meeting_duration	string		☐	Meeting duration
live_meeting_email	string		☐	User email
live_meeting_end_time	datetime		☐	Meeting end time
live_meeting_has_3rd_party_audio	boolean		☐	
live_meeting_has_pstn	boolean		☐	
live_meeting_has_recording	boolean		☐	
live_meeting_has_screen_share	boolean		☐	
live_meeting_has_sip	boolean		☐	
live_meeting_has_video	boolean		☐	
live_meeting_has_voip	boolean		☐	
live_meeting_host	string		☐	User display name
live_meeting_id	int64		☐	Meeting ID
live_meeting_participants	int64		☐	Meeting participant count
live_meeting_start_time	datetime		☐	Meeting start time
live_meeting_topic	string		☐	Meeting topic
live_meeting_user_type	string		☐	User type
live_meeting_uuid	guid		☐	Meeting UUID
microphone	string		☐	Zoom Room microphone
room_name	string		☐	Zoom Room name
speaker	string		☐	Zoom Room speaker
status	string		☐	Zoom Room status

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

7 Schema: Devices

7.1 Tables

7.1.1 deleteH323DevicesByDeviceId

Delete a H.323/SIP DeviceDelete a H.323/SIP Device on your Zoom account

Catalog: Zoom

Schema: Devices

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

Select Zoom API URL: `/h323/devices/{deviceId}`

Insert Zoom API URL: `/h323/devices/{deviceId}`

Update Zoom API URL: `/h323/devices/{deviceId}`

Delete Zoom API URL: `/h323/devices/{deviceId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /h323/devices/{deviceId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteH323DevicesByDeviceId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
deviceId	string	☑		The device ID

Table Function Columns

The columns of the table function `deleteH323DevicesByDeviceId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

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

7.1.2 H323Devices

List H.323/SIP Devices.List H.323/SIP Devices on your Zoom account.

Catalog: Zoom

Schema: Devices

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

Select Zoom API URL: `/h323/devices`

Insert Zoom API URL: `/h323/devices`

Update Zoom API URL: `/h323/devices`

Delete Zoom API URL: `/h323/devices`

Field Selection Method: NotRequired

Base Path: `devices[*]`

Select Zoom API Operation: `get /h323/devices`

Table Columns

The columns of the table `H323Devices` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
encryption	string		☑	Device encryption
id	string		☐	Device ID
ip	string		☑	Device Ip
name	string(64)		☑	Device name
protocol	string		☑	Device protocol

7.1.3 patchH323DevicesByDeviceId

Update a H.323/SIP DeviceUpdate a H.323/SIP Device on your Zoom account

Catalog: Zoom

Schema: Devices

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

Select Zoom API URL: `/h323/devices/{deviceId}`

Insert Zoom API URL: `/h323/devices/{deviceId}`

Update Zoom API URL: `/h323/devices/{deviceId}`

Delete Zoom API URL: `/h323/devices/{deviceId}`

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

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /h323/devices/{deviceId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function patchH323DevicesByDeviceId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
deviceId	string	☑		The device ID

Table Function Columns

The columns of the table function `patchH323DevicesByDeviceId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

7.1.4 postH323Devices

Create a H.323/SIP DeviceCreate a H.323/SIP Device on your Zoom account

Catalog: Zoom

Schema: Devices

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

Select Zoom API URL: `/h323/devices`

Insert Zoom API URL: `/h323/devices`

Update Zoom API URL: `/h323/devices`

Delete Zoom API URL: `/h323/devices`

Field Selection Method: NotRequired

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

Select Zoom API Operation: `post /h323/devices`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function postH323Devices. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		H.323/SIP Device

Table Function Columns

The columns of the table function `postH323Devices` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
encryption	string		☑	Device encryption
id	string		☐	Device ID
ip	string		☑	Device Ip
name	string(64)		☑	Device name
protocol	string		☑	Device protocol
RESULT	string		☐	Outcome of operation as single plain text column.

8 Schema: Groups

8.1 Tables

8.1.1 deleteGroupsByGroupId

Delete a groupDelete a group under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups/{groupId}`

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

Insert Zoom API URL: `/groups/{groupId}`

Update Zoom API URL: `/groups/{groupId}`

Delete Zoom API URL: `/groups/{groupId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /groups/{groupId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

8.1.2 deleteGroupsByGroupIdMembersMemberId

Delete a group memberDelete a member from a group under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups/{groupId}/members/{memberId}`

Insert Zoom API URL: `/groups/{groupId}/members/{memberId}`

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

Update Zoom API URL: `/groups/{groupId}/members/{memberId}`

Delete Zoom API URL: `/groups/{groupId}/members/{memberId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /groups/{groupId}/members/{memberId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID
memberId	string	☑		The member ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

8.1.3 Groups

List groupsList groups under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups`

Insert Zoom API URL: `/groups`

Update Zoom API URL: `/groups`

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

Delete Zoom API URL: `/groups`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /groups`

Table Columns

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

Name	Data Type	Label	Required	Documentation
total_records	int64		☐	Total records

8.1.4 Groups_Groups

List groupsList groups under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups`

Insert Zoom API URL: `/groups`

Update Zoom API URL: `/groups`

Delete Zoom API URL: `/groups`

Field Selection Method: NotRequired

Base Path: `groups[*]`

Select Zoom API Operation: `get /groups`

Table Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Group ID
name	string		☐	Group name
total_members	int64		☐	Total number of members in this group
total_records	int64		☐	Total records

8.1.5 GroupsByGroupId

Retrieve a groupRetrieve a group under your account

Catalog: Zoom

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

Schema: Groups

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

Select Zoom API URL: `/groups/{groupId}`

Insert Zoom API URL: `/groups/{groupId}`

Update Zoom API URL: `/groups/{groupId}`

Delete Zoom API URL: `/groups/{groupId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /groups/{groupId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Group ID
name	string		☐	Group name
total_members	int64		☐	Total number of members in this group

8.1.6 GroupsByGroupIdMembers

List a group's membersList a group's members under your account

Catalog: Zoom

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

Schema: Groups

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

Select Zoom API URL: `/groups/{groupId}/members`

Insert Zoom API URL: `/groups/{groupId}/members`

Update Zoom API URL: `/groups/{groupId}/members`

Delete Zoom API URL: `/groups/{groupId}/members`

Field Selection Method: NotRequired

Base Path: `members[*]`

Select Zoom API Operation: `get /groups/{groupId}/members`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
email	string		☐	User email
first_name	string		☐	User first name
id	string		☐	User ID
last_name	string		☐	User last name

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

Name	Data Type	Label	Required	Documentation
type	int64		☐	User type

8.1.7 patchGroupsByGroupId

Update a groupUpdate a group under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups/{groupId}`

Insert Zoom API URL: `/groups/{groupId}`

Update Zoom API URL: `/groups/{groupId}`

Delete Zoom API URL: `/groups/{groupId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /groups/{groupId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
groupId	string	☑		The group ID

Table Function Columns

The columns of the table function `patchGroupsByGroupId` 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
RESULT	string		☐	Outcome of operation as single plain text column.

8.1.8 postGroups

Create a groupCreate a group under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups`

Insert Zoom API URL: `/groups`

Update Zoom API URL: `/groups`

Delete Zoom API URL: `/groups`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /groups`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		

Table Function Columns

The columns of the table function `postGroups` 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	string		☐	Group ID
name	string		☐	Group name
RESULT	string		☐	Outcome of operation as single plain text column.
total_members	int64		☐	Group member count

8.1.9 postGroupsByGroupIdMembers

Add group membersAdd members to a group under your account

Catalog: Zoom

Schema: Groups

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

Select Zoom API URL: `/groups/{groupId}/members`

Insert Zoom API URL: `/groups/{groupId}/members`

Update Zoom API URL: `/groups/{groupId}/members`

Delete Zoom API URL: `/groups/{groupId}/members`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /groups/{groupId}/members`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
groupId	string	☑		The group ID

Table Function Columns

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

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

Name	Data Type	Label	Required	Documentation
added_at	datetime		☐	
ids	string		☐	
RESULT	string		☐	Outcome of operation as single plain text column.

9 Schema: IMChat

9.1 Tables

9.1.1 ImChat_SessionsSessions

Retrieve IM Chat sessionsRetrieve IM Chat sessions for a specified period <aside>This API only supports oauth2.</aside>

Catalog: Zoom

Schema: IMChat

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

Select Zoom API URL: `/im/chat/sessions`

Insert Zoom API URL: `/im/chat/sessions`

Update Zoom API URL: `/im/chat/sessions`

Delete Zoom API URL: `/im/chat/sessions`

Field Selection Method: NotRequired

Base Path: `sessions[*]`

Select Zoom API Operation: `get /im/chat/sessions`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date
last_message_sent_time	datetime		☐	Last message sent time
name	string		☐	Meeting topic
next_page_token	string		☐	Next page token, used to paginate through large result sets. A next page token will be returned whenever the set of available result list exceeds page size. The expiration period is 15 minutes.
page_size	int64		☐	The amount of records returns within a single API call.
session_id	string		☐	IM Chat session ID
to	datetime		☐	End date
type	string		☐	IM Chat session type

9.1.2 ImChatSessions

Retrieve IM Chat sessionsRetrieve IM Chat sessions for a specified period <aside>This API only supports oauth2.</aside>

Catalog: Zoom

Schema: IMChat

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

Select Zoom API URL: `/im/chat/sessions`

Insert Zoom API URL: `/im/chat/sessions`

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

Update Zoom API URL: `/im/chat/sessions`

Delete Zoom API URL: `/im/chat/sessions`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /im/chat/sessions`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date
next_page_token	string		☐	Next page token, used to paginate through large result sets. A next page token will be returned whenever the set of available result list exceeds page size. The expiration period is 15 minutes.
page_size	int64		☐	The amount of records returns within a single API call.

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

Name	Data Type	Label	Required	Documentation
to	datetime		☐	End date

9.1.3 ImChatSessions_MessagesBySessionId

Retrieve IM Chat messagesRetrieve IM Chat messages for a specified period <aside>This API only supports oauth2.</aside>

Catalog: Zoom

Schema: IMChat

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

Select Zoom API URL: `/im/chat/sessions/{sessionId}`

Insert Zoom API URL: `/im/chat/sessions/{sessionId}`

Update Zoom API URL: `/im/chat/sessions/{sessionId}`

Delete Zoom API URL: `/im/chat/sessions/{sessionId}`

Field Selection Method: NotRequired

Base Path: `messages[*]`

Select Zoom API Operation: `get /im/chat/sessions/{sessionId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.

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

Name	Data Type	Required	Default Value	Documentation
page_size	int64	☐	30	The number of records returned within a single API call
sessionId	string	☑		IM Chat Session ID
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
action_time	datetime		☐	Action time
action	string		☐	IM Chat message action
date_time	datetime		☐	IM Chat message sent time
from	datetime		☐	Start date
message	string		☐	IM Chat message content
next_page_token	string		☐	Next page token, used to paginate through large result sets. A next page token will be returned whenever the set of available result list exceeds page size. The expiration period is 15 minutes.
page_size	int64		☐	The amount of records returns within a single API call.
sender	string		☐	IM Chat message sender
session_id	string		☐	IM Chat session ID
to	datetime		☐	End date

9.1.4 ImChatSessionsBySessionId

Retrieve IM Chat messagesRetrieve IM Chat messages for a specified period <aside>This API only supports oauth2.</aside>

Catalog: Zoom

Schema: IMChat

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

Select Zoom API URL: `/im/chat/sessions/{sessionId}`

Insert Zoom API URL: `/im/chat/sessions/{sessionId}`

Update Zoom API URL: `/im/chat/sessions/{sessionId}`

Delete Zoom API URL: `/im/chat/sessions/{sessionId}`

Field Selection Method: NotRequired

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

Select Zoom API Operation: `get /im/chat/sessions/{sessionId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
sessionId	string	☑		IM Chat Session ID
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date
next_page_token	string		☐	Next page token, used to paginate through large result sets. A next page token will be returned whenever the set of available result list exceeds page size. The expiration period is 15 minutes.
page_size	int64		☐	The amount of records returns within a single API call.
session_id	string		☐	IM Chat session ID
to	datetime		☐	End date

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

10 Schema: IMGroups

10.1 Tables

10.1.1 deleteImGroupsByGroupId

Delete an IM GroupDelete an IM Group under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups/{groupId}`

Insert Zoom API URL: `/im/groups/{groupId}`

Update Zoom API URL: `/im/groups/{groupId}`

Delete Zoom API URL: `/im/groups/{groupId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /im/groups/{groupId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

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

10.1.2 deleteImGroupsByGroupIdMembersMemberId

Delete an IM Group memberDelete a member from an IM Group under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups/{groupId}/members/{memberId}`

Insert Zoom API URL: `/im/groups/{groupId}/members/{memberId}`

Update Zoom API URL: `/im/groups/{groupId}/members/{memberId}`

Delete Zoom API URL: `/im/groups/{groupId}/members/{memberId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /im/groups/{groupId}/members/{memberId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID
memberId	string	☑		The member ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

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

10.1.3 ImGroups

List IM GroupsList IM groups under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups`

Insert Zoom API URL: `/im/groups`

Update Zoom API URL: `/im/groups`

Delete Zoom API URL: `/im/groups`

Field Selection Method: NotRequired

Base Path: `groups[*]`

Select Zoom API Operation: `get /im/groups`

Table Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	IM Group ID
name	string		☐	Group name
search_by_account	boolean		☐	Members can search others under same account
search_by_domain	boolean		☐	Members can search others in the same email domain
search_by_ma_account	boolean		☐	Members can search others under same master account, including all sub accounts
total_members	int64		☐	Total number of members in this group
type	string		☐	IM Group type

10.1.4 ImGroupsByGroupId

Retrieve an IM GroupRetrieve an IM Group under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups/{groupId}`

Insert Zoom API URL: `/im/groups/{groupId}`

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

Update Zoom API URL: `/im/groups/{groupId}`

Delete Zoom API URL: `/im/groups/{groupId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /im/groups/{groupId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Group ID
name	string		☐	Group name
search_by_account	boolean		☐	Members can search others under same account
search_by_domain	boolean		☐	Members can search others in the same email domain
search_by_ma_account	boolean		☐	Members can search others under same master account, including all sub accounts
total_members	int64		☐	Total number of members in this group
type	string		☐	IM Group type

10.1.5 ImGroupsByGroupIdMembers

List an IM Group's membersList an IM Group's members under your account

Catalog: Zoom

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

Schema: IMGroups

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

Select Zoom API URL: `/im/groups/{groupId}/members`

Insert Zoom API URL: `/im/groups/{groupId}/members`

Update Zoom API URL: `/im/groups/{groupId}/members`

Delete Zoom API URL: `/im/groups/{groupId}/members`

Field Selection Method: NotRequired

Base Path: `members[*]`

Select Zoom API Operation: `get /im/groups/{groupId}/members`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
groupId	string	☑		The group ID
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
email	string		☐	User email
first_name	string		☐	User first name
id	string		☐	User ID
last_name	string		☐	User last name

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

Name	Data Type	Label	Required	Documentation
type	int64		☐	User type

10.1.6 patchImGroupsByGroupId

Update an IM GroupUpdate an IM Group under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups/{groupId}`

Insert Zoom API URL: `/im/groups/{groupId}`

Update Zoom API URL: `/im/groups/{groupId}`

Delete Zoom API URL: `/im/groups/{groupId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /im/groups/{groupId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
groupId	string	☑		The group ID

Table Function Columns

The columns of the table function `patchImGroupsByGroupId` 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
RESULT	string		☐	Outcome of operation as single plain text column.

10.1.7 postImGroups

Create an IM GroupCreate a IM Group under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups`

Insert Zoom API URL: `/im/groups`

Update Zoom API URL: `/im/groups`

Delete Zoom API URL: `/im/groups`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /im/groups`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		

Table Function Columns

The columns of the table function `postImGroups` 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	string		☐	Group ID
name	string		☐	Group name
RESULT	string		☐	Outcome of operation as single plain text column.
search_by_account	boolean		☐	Members can search others under same account
search_by_domain	boolean		☐	Members can search others in the same email domain
search_by_ma_account	boolean		☐	Members can search others under same master account, including all sub accounts
total_members	int64		☐	Group member count

10.1.8 postImGroupsByGroupIdMembers

Add IM Group membersAdd members to an IM Group under your account

Catalog: Zoom

Schema: IMGroups

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

Select Zoom API URL: `/im/groups/{groupId}/members`

Insert Zoom API URL: `/im/groups/{groupId}/members`

Update Zoom API URL: `/im/groups/{groupId}/members`

Delete Zoom API URL: `/im/groups/{groupId}/members`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /im/groups/{groupId}/members`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
groupId	string	☑		The group ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
added_at	datetime		☐	
ids	string		☐	
RESULT	string		☐	Outcome of operation as single plain text column.

11 Schema: Meetings

11.1 Tables

11.1.1 deleteMeetingsByMeetingId

Delete a meetingDelete a meeting

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}`

Insert Zoom API URL: `/meetings/{meetingId}`

Update Zoom API URL: `/meetings/{meetingId}`

Delete Zoom API URL: `/meetings/{meetingId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /meetings/{meetingId}`

Parameters of Table Function

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

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

(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
meetingId	int64	☑		The meeting ID
occurrence_id	string	☐		The meeting occurrence ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.2 deleteMeetingsByMeetingIdPollsPollId

Delete a meeting's PollDelete a meeting's Poll

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Insert Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Update Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Delete Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /meetings/{meetingId}/polls/{pollId}`

Parameters of Table Function

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

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

(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
meetingId	int64	☑		The meeting ID
pollId	string	☑		The poll ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.3 MeetingsByMeetingId

Retrieve a meetingRetrieve a meeting's details

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}`

Insert Zoom API URL: `/meetings/{meetingId}`

Update Zoom API URL: `/meetings/{meetingId}`

Delete Zoom API URL: `/meetings/{meetingId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}`

Parameters of Table Function

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

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

(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
meetingId	int64	☑		The meeting ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
agenda	string		☐	Agenda
created_at	datetime		☐	Create time
duration	int64		☐	Meeting duration
h323_password	string		☐	H.323/SIP room system password
host_id	string		☐	ID of the user set as host of meeting
id	string		☐	Meeting ID, also know as meeting number
join_url	string		☐	Join url
password	string		☐	Meeting password
settings_alternative_hosts	string		☐	Alternative hosts emails or IDs. Multiple value separated by comma.
settings_approval_type	int64		☐	
settings_audio	string		☐	Determine how participants can join the audio portion of the meeting
settings_auto_recording	string		☐	
settings_close_registration	boolean		☐	Close registration after event date
settings_cn_meeting	boolean		☐	Host meeting in China
settings_enforce_login_domains	string		☐	Only signed-in users with specified domains can join meetings
settings_enforce_login	boolean		☐	Only signed-in users can join this meeting
settings_host_video	boolean		☐	Start video when host joins meeting
settings_in_meeting	boolean		☐	Host meeting in India
settings_join_before_host	boolean		☐	Allow participants to join the meeting before the host starts the meeting. Only used for scheduled or recurring meetings.
settings_mute_upon_entry	boolean		☐	Mute participants upon entry

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

Name	Data Type	Label	Required	Documentation
settings_participant_video	boolean		☐	Start video when participants join meeting
settings_registration_type	int64		☐	Registration type. Used for recurring meeting with fixed time only.
settings_use_pmi	boolean		☐	Use Personal Meeting ID. Only used for scheduled meetings and recurring meetings with no fixed time.
settings_waiting_room	boolean		☐	Enable waiting room
settings_watermark	boolean		☐	Add watermark when viewing shared screen
start_time	datetime		☐	Meeting start time
start_url	string		☐	Start url
timezone	string		☐	Timezone to format start_time
topic	string		☐	Meeting topic
type	int64		☐	Meeting Type
uuid	string		☐	Meeting unique ID

11.1.4 MeetingsByMeetingIdInvitation

Retrieve meeting invitationRetrieve a meeting invitation

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/invitation`

Insert Zoom API URL: `/meetings/{meetingId}/invitation`

Update Zoom API URL: `/meetings/{meetingId}/invitation`

Delete Zoom API URL: `/meetings/{meetingId}/invitation`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}/invitation`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	int64	☑		The meeting ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
invitation	string		□	Meeting invitation

11.1.5 MeetingsByMeetingIdPolls

List a meeting's polls List polls of a meeting

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/polls`

Insert Zoom API URL: `/meetings/{meetingId}/polls`

Update Zoom API URL: `/meetings/{meetingId}/polls`

Delete Zoom API URL: `/meetings/{meetingId}/polls`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}/polls`

Parameters of Table Function

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

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

(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
meetingId	int64	☑		The meeting ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
total_records	int64		☐	The number of all records available across pages

11.1.6 MeetingsByMeetingIdPollsPollId

Retrieve a meeting's pollRetrieve a meeting's poll

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Insert Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Update Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Delete Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}/polls/{pollId}`

Parameters of Table Function

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

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

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

(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
meetingId	int64	☑		The meeting ID
pollId	string	☑		The poll ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Meeting Poll ID
status	string		☐	Status of the Meeting Poll
title	string		☐	Poll Title

11.1.7 MeetingsByMeetingIdRegistrants

List a meeting's registrantsList registrants of a meeting

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/registrants`

Insert Zoom API URL: `/meetings/{meetingId}/registrants`

Update Zoom API URL: `/meetings/{meetingId}/registrants`

Delete Zoom API URL: `/meetings/{meetingId}/registrants`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /meetings/{meetingId}/registrants`

Parameters of Table Function

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

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

(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
meetingId	int64	☑		The meeting ID
occurrence_id	string	☐		The meeting occurrence ID
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
status	string	☐	approved	The registrant status (Values: pending, approved, denied)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
page_count	int64		☐	The number of items returned on this page
page_number	int64		☐	The page number of current results
page_size	int64		☐	The number of records returned within a single API call
total_records	int64		☐	The number of all records available across pages

11.1.8 PastMeetingsByMeetingIdInstances

List of ended meeting instancesList of ended meeting instances

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/past_meetings/{meetingId}/instances`

Insert Zoom API URL: `/past_meetings/{meetingId}/instances`

Update Zoom API URL: `/past_meetings/{meetingId}/instances`

Delete Zoom API URL: `/past_meetings/{meetingId}/instances`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /past_meetings/{meetingId}/instances`

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	int64	☑		The meeting ID

Table Function Columns

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

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

11.1.9 PastMeetingsByMeetingUUID

Retrieve past meeting detailsRetrieve ended meeting details

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/past_meetings/{meetingUUID}`

Insert Zoom API URL: `/past_meetings/{meetingUUID}`

Update Zoom API URL: `/past_meetings/{meetingUUID}`

Delete Zoom API URL: `/past_meetings/{meetingUUID}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /past_meetings/{meetingUUID}`

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingUUID	string	☑		The meeting UUID.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
host_id	int64		☐	Host ID
id	int64		☐	Meeting ID
participants_count	int64		☐	Number of meeting participants
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
total_minutes	int64		☐	Number of meeting minutes
type	int64		☐	Meeting type
user_email	string		☐	User email
user_name	string		☐	User display name
uuid	guid		☐	Meeting UUID

11.1.10 PastMeetingsByMeetingUUIDParticipants

Retrieve past meeting participantsRetrieve ended meeting participants

Catalog: Zoom

Schema: Meetings

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

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

Select Zoom API URL: `/past_meetings/{meetingUUID}/participants`

Insert Zoom API URL: `/past_meetings/{meetingUUID}/participants`

Update Zoom API URL: `/past_meetings/{meetingUUID}/participants`

Delete Zoom API URL: `/past_meetings/{meetingUUID}/participants`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /past_meetings/{meetingUUID}/participants`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingUUID	string	☑		The meeting UUID.
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	guid		☐	Participant UUID

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

Name	Data Type	Label	Required	Documentation
name	string		☐	Participant display name

11.1.11 patchMeetingsByMeetingId

Update a meetingUpdate a meeting's details

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}`

Insert Zoom API URL: `/meetings/{meetingId}`

Update Zoom API URL: `/meetings/{meetingId}`

Delete Zoom API URL: `/meetings/{meetingId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /meetings/{meetingId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Meeting
meetingId	int64	☑		The meeting ID

Table Function Columns

The columns of the table function `patchMeetingsByMeetingId` 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
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.12 patchMeetingsByMeetingIdLivestream

Update a meeting live streamUpdate a meeting's live stream

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/livestream`

Insert Zoom API URL: `/meetings/{meetingId}/livestream`

Update Zoom API URL: `/meetings/{meetingId}/livestream`

Delete Zoom API URL: `/meetings/{meetingId}/livestream`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /meetings/{meetingId}/livestream`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Meeting
meetingId	int64	☑		The meeting ID

Table Function Columns

The columns of the table function `patchMeetingsByMeetingIdLivestream` 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
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.13 patchMeetingsByMeetingIdLivestreamStatus

Update a meeting live stream statusUpdate a meeting's live stream status

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/livestream/status`

Insert Zoom API URL: `/meetings/{meetingId}/livestream/status`

Update Zoom API URL: `/meetings/{meetingId}/livestream/status`

Delete Zoom API URL: `/meetings/{meetingId}/livestream/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /meetings/{meetingId}/livestream/status`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Meeting
meetingId	int64	☑		The meeting ID

Table Function Columns

The columns of the table function `patchMeetingsByMeetingIdLivestreamStatus` 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
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.14 postMeetingsByMeetingIdPolls

Create a meeting's pollCreate a poll for a meeting

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/polls`

Insert Zoom API URL: `/meetings/{meetingId}/polls`

Update Zoom API URL: `/meetings/{meetingId}/polls`

Delete Zoom API URL: `/meetings/{meetingId}/polls`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /meetings/{meetingId}/polls`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Meeting poll object
meetingId	int64	☑		The meeting ID

Table Function Columns

The columns of the table function `postMeetingsByMeetingIdPolls` 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	string		☐	Meeting Poll ID
RESULT	string		☐	Outcome of operation as single plain text column.
status	string		☐	Status of the Meeting Poll
title	string		☐	Poll Title

11.1.15 postMeetingsByMeetingIdRegistrants

Add a meeting registrantRegister a participant for a meeting

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/registrants`

Insert Zoom API URL: `/meetings/{meetingId}/registrants`

Update Zoom API URL: `/meetings/{meetingId}/registrants`

Delete Zoom API URL: `/meetings/{meetingId}/registrants`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /meetings/{meetingId}/registrants`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
meetingId	int64	☑		The meeting ID
occurrence_ids	string	☐		Occurrence IDs. You can find these w ith the meeting get API. Multiple values separated by comma.

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

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	string		☐	Registrant ID
join_url	string		☐	Join URL for this registrant
registrant_id	string		☐	Registrant ID
RESULT	string		☐	Outcome of operation as single plain text column.
start_time	datetime		☐	Start time
topic	string		☐	Topic

11.1.16 postUsersByUserIdMeetings

Create a meetingCreate a meeting for a user <aside>The expiration time of start_url is two hours. But for API users, the expiration time is 90 days.</aside>

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/users/{userId}/meetings`

Insert Zoom API URL: `/users/{userId}/meetings`

Update Zoom API URL: `/users/{userId}/meetings`

Delete Zoom API URL: `/users/{userId}/meetings`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /users/{userId}/meetings`

Parameters of Table Function

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

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

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

(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
body	string	☑		Meeting object
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
agenda	string		☐	Agenda
created_at	datetime		☐	Create time
duration	int64		☐	Meeting duration
h323_password	string		☐	H.323/SIP room system password
host_id	string		☐	ID of the user set as host of meeting
id	string		☐	Meeting ID, also known as meeting number
join_url	string		☐	Join url
password	string		☐	Meeting password
RESULT	string		☐	Outcome of operation as single plain text column.
settings_alternative_hosts	string		☐	Alternative hosts emails or IDs. Multiple value separated by comma.
settings_approval_type	int64		☐	
settings_audio	string		☐	Determine how participants can join the audio portion of the meeting
settings_auto_recording	string		☐	
settings_close_registration	boolean		☐	Close registration after event date
settings_cn_meeting	boolean		☐	Host meeting in China
settings_enforce_login_domains	string		☐	Only signed-in users with specified domains can join meetings
settings_enforce_login	boolean		☐	Only signed-in users can join this meeting
settings_host_video	boolean		☐	Start video when host joins meeting
settings_in_meeting	boolean		☐	Host meeting in India
settings_join_before_host	boolean		☐	Allow participants to join the meeting before the host starts the meeting. Only used for scheduled or recurring meetings.
settings_mute_upon_entry	boolean		☐	Mute participants upon entry

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

Name	Data Type	Label	Required	Documentation
settings_participant_video	boolean		☐	Start video when participants join meeting
settings_registration_type	int64		☐	Registration type. Used for recurring meeting with fixed time only.
settings_use_pmi	boolean		☐	Use Personal Meeting ID. Only used for scheduled meetings and recurring meetings with no fixed time.
settings_waiting_room	boolean		☐	Enable waiting room
settings_watermark	boolean		☐	Add watermark when viewing shared screen
start_time	datetime		☐	Meeting start time
start_url	string		☐	Start url
timezone	string		☐	Timezone to format start_time
topic	string		☐	Meeting topic
type	int64		☐	Meeting Type
uuid	string		☐	Meeting unique ID

11.1.17 putMeetingsByMeetingIdPollsPollId

Update a meeting's pollUpdate a meeting's poll

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Insert Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Update Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Delete Zoom API URL: `/meetings/{meetingId}/polls/{pollId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /meetings/{meetingId}/polls/{pollId}`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Meeting Poll
meetingId	int64	☑		The meeting ID
pollId	string	☑		The poll ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		□	Outcome of operation as single plain text column.

11.1.18 putMeetingsByMeetingIdRegistrantsStatus

Update a meeting registrant's statusUpdate a meeting registrant's status

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/registrants/status`

Insert Zoom API URL: `/meetings/{meetingId}/registrants/status`

Update Zoom API URL: `/meetings/{meetingId}/registrants/status`

Delete Zoom API URL: `/meetings/{meetingId}/registrants/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /meetings/{meetingId}/registrants/status`

Parameters of Table Function

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

(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 with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
meetingId	int64	☑		The meeting ID
occurrence_id	string	☐		The meeting occurrence ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.19 putMeetingsByMeetingIdStatus

Update a meeting's statusUpdate a meeting's status

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/meetings/{meetingId}/status`

Insert Zoom API URL: `/meetings/{meetingId}/status`

Update Zoom API URL: `/meetings/{meetingId}/status`

Delete Zoom API URL: `/meetings/{meetingId}/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /meetings/{meetingId}/status`

Parameters of Table Function

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

(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 with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
meetingId	int64	☑		The meeting ID

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

11.1.20 UsersByUserIdMeetings

List meetingsList meetings for a user

Catalog: Zoom

Schema: Meetings

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

Select Zoom API URL: `/users/{userId}/meetings`

Insert Zoom API URL: `/users/{userId}/meetings`

Update Zoom API URL: `/users/{userId}/meetings`

Delete Zoom API URL: `/users/{userId}/meetings`

Field Selection Method: NotRequired

Base Path: `meetings[*]`

Select Zoom API Operation: `get /users/{userId}/meetings`

Parameters of Table Function

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

(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 with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

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

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
type	string	☐	live	The meeting type (Values: scheduled, live, upcoming)
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
created_at	datetime		☐	Create time
duration	int64		☐	Meeting duration
host_id	string		☐	ID of the user set as host of meeting
id	string		☐	Meeting ID, also know as meeting number
join_url	string		☐	Join url
start_time	datetime		☐	Meeting start time
timezone	string		☐	Timezone to format start_time
topic	string		☐	Meeting topic
type	int64		☐	Meeting Type
uuid	string		☐	Meeting unique ID

12 Schema: Native

12.1 Tables

12.1.1 NATIVEPLATFORMSCALARREQUESTS: Zoom Native Platform Scalar Requests

Direct access to native API.

Catalog: Zoom

Schema: Native

Alias: `npt`

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

Label: Native Platform Scalar Requests

Documentation:

The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the Zoom 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 Zoom API server.

Retrieve: true

Insert: true

Update: false

Delete: false

View Columns

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

Name	Data Type	Label	Required	Documentation
BLOB_PREFERRED	boolean	BLOB Preferred	☑	Indicator whether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☑	
DATE_STARTED	datetime	Start Date	☑	
DRY_RUN	boolean	Run without Actions	☑	
DURATION_MS	int32	Duration (ms)	☑	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	☐	
FAIL_ON_ERROR	boolean	Fail on Error	☑	Whether to raise an exception when processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	☐	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.

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

Name	Data Type	Label	Required	Documentation
HTTP_METHOD	string(30)	HTTP Method	☐	
HTTP_STATUS_CODE	int16	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	☐	
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	☐	
PAYLOAD_TEXT	string	Payload	☐	
RESULT_BLOB	byte[]	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime		☐	
RESULT_NUMBER	decimal		☐	
RESULT_TEXT	string	Result Text	☐	
SUCCESSFUL	boolean	Succesful	☑	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int32	Transaction ID	☑	Incrementing ID of the transaction.
URL	string(4000)	URL	☐	

13 Schema: PAC

13.1 Tables

13.1.1 UsersByUserId_Tsp_accountsDedicated_dial_in_numberPac

List user's PAC accountsList user's PAC accounts

Catalog: Zoom

Schema: PAC

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

Select Zoom API URL: `/users/{userId}/pac`

Insert Zoom API URL: `/users/{userId}/pac`

Update Zoom API URL: `/users/{userId}/pac`

Delete Zoom API URL: `/users/{userId}/pac`

Field Selection Method: NotRequired

Base Path: `tsp_accounts[*].dedicated_dial_in_number[*]`

Select Zoom API Operation: `get /users/{userId}/pac`

Parameters of Table Function

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

(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 with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

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

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
conference_id	int64		☐	Conference ID
country	string		☐	Country Code
listen_only_password	string		☐	Listen-Only Password, numeric value, length is less than 6
number	string		☐	Dial-in number, length is less than 16
participant_password	string		☐	Participant Password, numeric value, length is less than 6

13.1.2 UsersByUserId_Tsp_accountsGlobal_dial_in_numbersPac

List user's PAC accountsList user's PAC accounts

Catalog: Zoom

Schema: PAC

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

Select Zoom API URL: `/users/{userId}/pac`

Insert Zoom API URL: `/users/{userId}/pac`

Update Zoom API URL: `/users/{userId}/pac`

Delete Zoom API URL: `/users/{userId}/pac`

Field Selection Method: NotRequired

Base Path: `tsp_accounts[*].global_dial_in_numbers[*]`

Select Zoom API Operation: `get /users/{userId}/pac`

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

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

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

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
conference_id	int64		☐	Conference ID
country	string		☐	Country Code
listen_only_password	string		☐	Listen-Only Password, numeric value, length is less than 6
number	string		☐	Dial-in number, length is less than 16
participant_password	string		☐	Participant Password, numeric value, length is less than 6

13.1.3 UsersByUserIdPac

List user's PAC accountsList user's PAC accounts

Catalog: Zoom

Schema: PAC

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

Select Zoom API URL: `/users/{userId}/pac`

Insert Zoom API URL: `/users/{userId}/pac`

Update Zoom API URL: `/users/{userId}/pac`

Delete Zoom API URL: `/users/{userId}/pac`

Field Selection Method: NotRequired

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

Base Path: `tsp_accounts[*]`

Select Zoom API Operation: `get /users/{userId}/pac`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
conference_id	int64		□	Conference ID
listen_only_password	string		□	Listen-Only Password, numeric value, length is less than 6
participant_password	string		□	Participant Password, numeric value, length is less than 6

14 Schema: Reports

14.1 Tables

14.1.1 Report_DatesDaily

Retrieve daily reportRetrieve daily report for one month, can only get daily report for recent 6 months

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/daily`

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

Insert Zoom API URL: `/report/daily`

Update Zoom API URL: `/report/daily`

Delete Zoom API URL: `/report/daily`

Field Selection Method: NotRequired

Base Path: `dates[*]`

Select Zoom API Operation: `get /report/daily`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
month	int64	☐		Month for this report
year	int64	☐		Year for this report

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
date	datetime		☐	Date for this object
meeting_minutes	int64		☐	Number of meeting minutes on this date
meetings	int64		☐	Number of meetings on this date
month	int64		☐	Month for this report
new_users	int64		☐	Number of new users on this date
participants	int64		☐	Number of participants on this date
year	int64		☐	Year for this report

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

14.1.2 Report_Telephony_usageTelephone

Retrieve telephone reportRetrieve telephone report for a specified period <aside>Toll Report option would be removed.</aside>.

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/telephone`

Insert Zoom API URL: `/report/telephone`

Update Zoom API URL: `/report/telephone`

Delete Zoom API URL: `/report/telephone`

Field Selection Method: NotRequired

Base Path: `telephony_usage[*]`

Select Zoom API Operation: `get /report/telephone`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
type	string	☐	1	Audio type (Values: 1)

Table Function Columns

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

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

Name	Data Type	Label	Required	Documentation
call_in_number	string		☐	Call in number
country_name	string		☐	Country Name
dept	string		☐	User department
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
from	datetime		☐	Start date for this report
host_email	string		☐	User email
host_name	string		☐	User display name
meeting_id	int64		☐	Meeting ID
meeting_type	string		☐	Meeting Type
page_count	int64		☐	The number of items returned on this page
page_number	int64		☐	The page number of current results
page_size	int64		☐	The number of records returned within a single API call
phone_number	string		☐	Telephone Number
start_time	datetime		☐	Meeting start time
to	datetime		☐	End date for this report
total_records	int64		☐	The number of all records available across pages
total	decimal		☐	Total

14.1.3 Report_UsersUsers

Retrieve hosts reportRetrieve active or inactive hosts report for a specified period

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/users`

Insert Zoom API URL: `/report/users`

Update Zoom API URL: `/report/users`

Delete Zoom API URL: `/report/users`

Field Selection Method: NotRequired

Base Path: `users[*]`

Select Zoom API Operation: `get /report/users`

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
type	string	☐		Active hosts or inactive hosts (Values: active, inactive)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
dept	string		☐	User department
email	string		☐	User email
from	datetime		☐	Start date for this report
id	guid		☐	User ID
meeting_minutes	int64		☐	Number of meeting minutes for user
meetings	int64		☐	Number of meetings for user
page_count	int64		☐	The number of items returned on this page
page_number	int64		☐	The page number of current results
page_size	int64		☐	The number of records returned within a single API call
participants	int64		☐	Number of participants in meetings for user
to	datetime		☐	End date for this report

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

Name	Data Type	Label	Required	Documentation
total_meeting_minutes	int64		☐	Number of meeting minutes for this range
total_meetings	int64		☐	Number of meetings for this range
total_participants	int64		☐	Number of participants for this range
total_records	int64		☐	The number of all records available across pages
type	int64		☐	User type
user_name	string		☐	User display name

14.1.4 ReportCloudRecording

Retrieve cloud recording usage reportRetrieve cloud recording usage report for a specified period. You can only get cloud recording reports for the most recent period of 6 months. The date gap between from and to dates should be smaller or equal to 30 days.

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/cloud_recording`

Insert Zoom API URL: `/report/cloud_recording`

Update Zoom API URL: `/report/cloud_recording`

Delete Zoom API URL: `/report/cloud_recording`

Field Selection Method: NotRequired

Base Path: `cloud_recording_storage[*]`

Select Zoom API Operation: `get /report/cloud_recording`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
to	datetime	☑		End Date

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
date	datetime		☐	
free_usage	string		☐	
plan_usage	string		☐	
usage	string		☐	

14.1.5 ReportDaily

Retrieve daily reportRetrieve daily report for one month, can only get daily report for recent 6 months

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/daily`

Insert Zoom API URL: `/report/daily`

Update Zoom API URL: `/report/daily`

Delete Zoom API URL: `/report/daily`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/daily`

Parameters of Table Function

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

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

(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
month	int64	□		Month for this report
year	int64	□		Year for this report

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
month	int64		□	Month for this report
year	int64		□	Year for this report

14.1.6 ReportMeetings_Tracking_fieldsByMeetingId

Retrieve meeting details reportRetrieve ended meeting details report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/meetings/{meetingId}`

Insert Zoom API URL: `/report/meetings/{meetingId}`

Update Zoom API URL: `/report/meetings/{meetingId}`

Delete Zoom API URL: `/report/meetings/{meetingId}`

Field Selection Method: NotRequired

Base Path: `tracking_fields[*]`

Select Zoom API Operation: `get /report/meetings/{meetingId}`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
field	string		☐	Tracking fields type
id	int64		☐	Meeting ID
participants_count	int64		☐	Number of meeting participants
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
total_minutes	int64		☐	Number of meeting minutes
type	int64		☐	Meeting type
user_email	string		☐	User email
user_name	string		☐	User display name
uuid	guid		☐	Meeting UUID
value	string		☐	Tracking fields value

14.1.7 ReportMeetingsByMeetingId

Retrieve meeting details reportRetrieve ended meeting details report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/meetings/{meetingId}`

Insert Zoom API URL: `/report/meetings/{meetingId}`

Update Zoom API URL: `/report/meetings/{meetingId}`

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

Delete Zoom API URL: `/report/meetings/{meetingId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/meetings/{meetingId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
id	int64		☐	Meeting ID
participants_count	int64		☐	Number of meeting participants
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
total_minutes	int64		☐	Number of meeting minutes
type	int64		☐	Meeting type
user_email	string		☐	User email
user_name	string		☐	User display name
uuid	guid		☐	Meeting UUID

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

14.1.8 ReportMeetingsByMeetingId_QuestionsPolls

Retrieve meeting polls reportRetrieve ended meeting polls report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/meetings/{meetingId}/polls`

Insert Zoom API URL: `/report/meetings/{meetingId}/polls`

Update Zoom API URL: `/report/meetings/{meetingId}/polls`

Delete Zoom API URL: `/report/meetings/{meetingId}/polls`

Field Selection Method: NotRequired

Base Path: `questions[*]`

Select Zoom API Operation: `get /report/meetings/{meetingId}/polls`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
email	string		☐	Participant email
id	int64		☐	Meeting ID
name	string		☐	Participant display name

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

Name	Data Type	Label	Required	Documentation
start_time	datetime		☐	Meeting start time
uuid	guid		☐	Meeting UUID

14.1.9 ReportMeetingsByMeetingId_QuestionsQuestion_detailsPolls

Retrieve meeting polls reportRetrieve ended meeting polls report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/meetings/{meetingId}/polls`

Insert Zoom API URL: `/report/meetings/{meetingId}/polls`

Update Zoom API URL: `/report/meetings/{meetingId}/polls`

Delete Zoom API URL: `/report/meetings/{meetingId}/polls`

Field Selection Method: NotRequired

Base Path: `questions[*].question_details[*]`

Select Zoom API Operation: `get /report/meetings/{meetingId}/polls`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

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

Name	Data Type	Label	Required	Documentation
answer	string		☐	Given answer
email	string		☐	Participant email
id	int64		☐	Meeting ID
name	string		☐	Participant display name
question	string		☐	Asked question
start_time	datetime		☐	Meeting start time
uuid	guid		☐	Meeting UUID

14.1.10 ReportMeetingsByMeetingIdParticipants

Retrieve meeting participants reportRetrieve ended meeting participants report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/meetings/{meetingId}/participants`

Insert Zoom API URL: `/report/meetings/{meetingId}/participants`

Update Zoom API URL: `/report/meetings/{meetingId}/participants`

Delete Zoom API URL: `/report/meetings/{meetingId}/participants`

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /report/meetings/{meetingId}/participants`

Parameters of Table Function

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, will take the last meeting instance.
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
attentiveness_score	int64		☐	Participant attentiveness score
duration	int64		☐	Participant duration
id	guid		☐	Participant UUID
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
name	string		☐	Participant display name
user_email	string		☐	Participant email
user_id	string		☐	Participant ID

14.1.11 ReportMeetingsByMeetingIdPolls

Retrieve meeting polls reportRetrieve ended meeting polls report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/meetings/{meetingId}/polls`

Insert Zoom API URL: `/report/meetings/{meetingId}/polls`

Update Zoom API URL: `/report/meetings/{meetingId}/polls`

Delete Zoom API URL: `/report/meetings/{meetingId}/polls`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/meetings/{meetingId}/polls`

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

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

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

Name	Data Type	Required	Default Value	Documentation
meetingId	string	☑		The meeting ID or meeting UUID. If given meeting ID, w ill take the last meeting instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	int64		☐	Meeting ID
start_time	datetime		☐	Meeting start time
uuid	guid		☐	Meeting UUID

14.1.12 ReportTelephone

Retrieve telephone reportRetrieve telephone report for a specified period <aside>Toll Report option would be removed.</aside>.

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/telephone`

Insert Zoom API URL: `/report/telephone`

Update Zoom API URL: `/report/telephone`

Delete Zoom API URL: `/report/telephone`

Field Selection Method: NotRequired

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

Select Zoom API Operation: `get /report/telephone`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
type	string	☐	1	Audio type (Values: 1)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date for this report
page_count	int64		☐	The number of items returned on this page
page_number	int64		☐	The page number of current results
page_size	int64		☐	The number of records returned within a single API call
to	datetime		☐	End date for this report
total_records	int64		☐	The number of all records available across pages

14.1.13 ReportUsers

Retrieve hosts reportRetrieve active or inactive hosts report for a specified period

Catalog: Zoom

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

Schema: Reports

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

Select Zoom API URL: `/report/users`

Insert Zoom API URL: `/report/users`

Update Zoom API URL: `/report/users`

Delete Zoom API URL: `/report/users`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/users`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
type	string	☐		Active hosts or inactive hosts (Values: active, inactive)

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date for this report
page_count	int64		☐	The number of items returned on this page

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

Name	Data Type	Label	Required	Documentation
page_number	int64		☐	The page number of current results
page_size	int64		☐	The number of records returned within a single API call
to	datetime		☐	End date for this report
total_meeting_minutes	int64		☐	Number of meeting minutes for this range
total_meetings	int64		☐	Number of meetings for this range
total_participants	int64		☐	Number of participants for this range
total_records	int64		☐	The number of all records available across pages

14.1.14 ReportUsersByUserId_MeetingsMeetings

Retrieve meetings reportRetrieve ended meetings report for a specified period

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/users/{userId}/meetings`

Insert Zoom API URL: `/report/users/{userId}/meetings`

Update Zoom API URL: `/report/users/{userId}/meetings`

Delete Zoom API URL: `/report/users/{userId}/meetings`

Field Selection Method: NotRequired

Base Path: `meetings[*]`

Select Zoom API Operation: `get /report/users/{userId}/meetings`

Parameters of Table Function

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

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

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

(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
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
to	datetime	☑		End Date
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
from	datetime		☐	Start date for this report
id	int64		☐	Meeting ID
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
participants_count	int64		☐	Number of meeting participants
start_time	datetime		☐	Meeting start time
to	datetime		☐	End date for this report
topic	string		☐	Meeting topic
total_minutes	int64		☐	Number of meeting minutes
total_records	int64		☐	The number of all records available across pages
type	int64		☐	Meeting type
user_email	string		☐	User email
user_name	string		☐	User display name

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

Name	Data Type	Label	Required	Documentation
uuid	guid		☐	Meeting UUID

14.1.15 ReportUsersByUserIdMeetings

Retrieve meetings reportRetrieve ended meetings report for a specified period

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/users/{userId}/meetings`

Insert Zoom API URL: `/report/users/{userId}/meetings`

Update Zoom API URL: `/report/users/{userId}/meetings`

Delete Zoom API URL: `/report/users/{userId}/meetings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/users/{userId}/meetings`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
from	datetime	☑		Start Date
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call

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

Name	Data Type	Required	Default Value	Documentation
to	datetime	☑		End Date
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
from	datetime		☐	Start date for this report
next_page_token	string		☐	Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_count	int64		☐	The number of items returned on this page
page_size	int64		☐	The number of records returned within a single API call.
to	datetime		☐	End date for this report
total_records	int64		☐	The number of all records available across pages

14.1.16 ReportWebinars_Tracking_fieldsByWebinarId

Retrieve webinar details reportRetrieve ended webinar details report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}`

Insert Zoom API URL: `/report/webinars/{webinarId}`

Update Zoom API URL: `/report/webinars/{webinarId}`

Delete Zoom API URL: `/report/webinars/{webinarId}`

Field Selection Method: NotRequired

Base Path: `tracking_fields[*]`

Select Zoom API Operation: `get /report/webinars/{webinarId}`

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
field	string		☐	Tracking fields type
id	int64		☐	Meeting ID
participants_count	int64		☐	Number of meeting participants
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
total_minutes	int64		☐	Number of meeting minutes
type	int64		☐	Meeting type
user_email	string		☐	User email
user_name	string		☐	User display name
uuid	guid		☐	Meeting UUID
value	string		☐	Tracking fields value

14.1.17 ReportWebinarsByWebinarId

Retrieve webinar details reportRetrieve ended webinar details report

Catalog: Zoom

Schema: Reports

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

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

Select Zoom API URL: `/report/webinars/{webinarId}`

Insert Zoom API URL: `/report/webinars/{webinarId}`

Update Zoom API URL: `/report/webinars/{webinarId}`

Delete Zoom API URL: `/report/webinars/{webinarId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/webinars/{webinarId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
duration	int64		☐	Meeting duration
end_time	datetime		☐	Meeting end time
id	int64		☐	Meeting ID
participants_count	int64		☐	Number of meeting participants
start_time	datetime		☐	Meeting start time
topic	string		☐	Meeting topic
total_minutes	int64		☐	Number of meeting minutes
type	int64		☐	Meeting type

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

Name	Data Type	Label	Required	Documentation
user_email	string		☐	User email
user_name	string		☐	User display name
uuid	guid		☐	Meeting UUID

14.1.18 ReportWebinarsByWebinarId_QuestionsPolls

Retrieve webinar polls reportRetrieve ended webinar polls report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/polls`

Insert Zoom API URL: `/report/webinars/{webinarId}/polls`

Update Zoom API URL: `/report/webinars/{webinarId}/polls`

Delete Zoom API URL: `/report/webinars/{webinarId}/polls`

Field Selection Method: NotRequired

Base Path: `questions[*]`

Select Zoom API Operation: `get /report/webinars/{webinarId}/polls`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

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

Name	Data Type	Label	Required	Documentation
email	string		☐	Participant email
id	int64		☐	Webinar ID
name	string		☐	Participant display name
start_time	datetime		☐	Webinar start time
uuid	guid		☐	Webinar UUID

14.1.19 ReportWebinarsByWebinarId_QuestionsQa

Retrieve webinar Q&A reportRetrieve ended webinar Q&A report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/qa`

Insert Zoom API URL: `/report/webinars/{webinarId}/qa`

Update Zoom API URL: `/report/webinars/{webinarId}/qa`

Delete Zoom API URL: `/report/webinars/{webinarId}/qa`

Field Selection Method: NotRequired

Base Path: `questions[*]`

Select Zoom API Operation: `get /report/webinars/{webinarId}/qa`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the

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

Name	Data Type	Required	Default Value	Documentation
				last w ebinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
email	string		☐	Participant email
id	int64		☐	Webinar ID
name	string		☐	Participant display name
start_time	datetime		☐	Webinar start time
uuid	guid		☐	Webinar UUID

14.1.20 ReportWebinarsByWebinarId_QuestionsQuestion_detailsPolls

Retrieve webinar polls reportRetrieve ended webinar polls report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/polls`

Insert Zoom API URL: `/report/webinars/{webinarId}/polls`

Update Zoom API URL: `/report/webinars/{webinarId}/polls`

Delete Zoom API URL: `/report/webinars/{webinarId}/polls`

Field Selection Method: NotRequired

Base Path: `questions[*].question_details[*]`

Select Zoom API Operation: `get /report/webinars/{webinarId}/polls`

Parameters of Table Function

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

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

(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
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
answer	string		☐	Given answer
email	string		☐	Participant email
id	int64		☐	Webinar ID
name	string		☐	Participant display name
question	string		☐	Asked question
start_time	datetime		☐	Webinar start time
uuid	guid		☐	Webinar UUID

14.1.21 ReportWebinarsByWebinarId_QuestionsQuestion_detailsQa

Retrieve webinar Q&A reportRetrieve ended webinar Q&A report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/qa`

Insert Zoom API URL: `/report/webinars/{webinarId}/qa`

Update Zoom API URL: `/report/webinars/{webinarId}/qa`

Delete Zoom API URL: `/report/webinars/{webinarId}/qa`

Field Selection Method: NotRequired

Base Path: `questions[*].question_details[*]`

Select Zoom API Operation: `get /report/webinars/{webinarId}/qa`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
answer	string		☐	Given answer
email	string		☐	Participant email
id	int64		☐	Webinar ID
name	string		☐	Participant display name
question	string		☐	Asked question
start_time	datetime		☐	Webinar start time
uuid	guid		☐	Webinar UUID

14.1.22 ReportWebinarsByWebinarIdParticipants

Retrieve webinar participants reportRetrieve ended webinar participants report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/participants`

Insert Zoom API URL: `/report/webinars/{webinarId}/participants`

Update Zoom API URL: `/report/webinars/{webinarId}/participants`

Delete Zoom API URL: `/report/webinars/{webinarId}/participants`

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

Field Selection Method: NotRequired

Base Path: `participants[*]`

Select Zoom API Operation: `get /report/webinars/{webinarId}/participants`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
next_page_token	string	☐		Next page token is used to paginate through large result sets. A next page token will be returned whenever the set of available results exceed the current page size. The expiration period for this token is 15 minutes.
page_size	int64	☐	30	The number of records returned within a single API call
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
attentiveness_score	string		☐	Participant attentiveness score
duration	int64		☐	Participant duration
id	guid		☐	Participant UUID
join_time	datetime		☐	Participant join time
leave_time	datetime		☐	Participant leave time
name	string		☐	Participant display name
user_email	string		☐	Participant email
user_id	string		☐	Participant ID

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

14.1.23 ReportWebinarsByWebinarIdPolls

Retrieve webinar polls reportRetrieve ended webinar polls report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/polls`

Insert Zoom API URL: `/report/webinars/{webinarId}/polls`

Update Zoom API URL: `/report/webinars/{webinarId}/polls`

Delete Zoom API URL: `/report/webinars/{webinarId}/polls`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/webinars/{webinarId}/polls`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
id	int64		☐	Webinar ID
start_time	datetime		☐	Webinar start time

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

Name	Data Type	Label	Required	Documentation
uuid	guid		☐	Webinar UUID

14.1.24 ReportWebinarsByWebinarIdQa

Retrieve webinar Q&A reportRetrieve ended webinar Q&A report

Catalog: Zoom

Schema: Reports

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

Select Zoom API URL: `/report/webinars/{webinarId}/qa`

Insert Zoom API URL: `/report/webinars/{webinarId}/qa`

Update Zoom API URL: `/report/webinars/{webinarId}/qa`

Delete Zoom API URL: `/report/webinars/{webinarId}/qa`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /report/webinars/{webinarId}/qa`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
webinarId	string	☑		The webinar ID or webinar UUID. If given webinar ID, will take the last webinar instance.

Table Function Columns

The columns of the table function `ReportWebinarsByWebinarIdQa` 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	int64		☐	Webinar ID
start_time	datetime		☐	Webinar start time
uuid	guid		☐	Webinar UUID

15 Schema: TrackingField

15.1 Tables

15.1.1 deleteV2TrackingFieldsByFieldId

Delete a Tracking FieldDelete a Tracking Field on your Zoom account

Catalog: Zoom

Schema: TrackingField

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

Select Zoom API URL: `/v2/tracking_fields/{fieldId}`

Insert Zoom API URL: `/v2/tracking_fields/{fieldId}`

Update Zoom API URL: `/v2/tracking_fields/{fieldId}`

Delete Zoom API URL: `/v2/tracking_fields/{fieldId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /v2/tracking_fields/{fieldId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteV2TrackingFieldsByFieldId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
fieldId	string	☑		The Tracking Field ID

Table Function Columns

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

The columns of the table function `deleteV2TrackingFieldsByFieldId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

15.1.2 patchV2TrackingFieldsByFieldId

Update a Tracking FieldUpdate a Tracking Field on your Zoom account

Catalog: Zoom

Schema: TrackingField

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

Select Zoom API URL: `/v2/tracking_fields/{fieldId}`

Insert Zoom API URL: `/v2/tracking_fields/{fieldId}`

Update Zoom API URL: `/v2/tracking_fields/{fieldId}`

Delete Zoom API URL: `/v2/tracking_fields/{fieldId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /v2/tracking_fields/{fieldId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function patchV2TrackingFieldsByFieldId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
fieldId	string	☑		The Tracking Field ID

Table Function Columns

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

The columns of the table function `patchV2TrackingFieldsByFieldId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

15.1.3 postV2TrackingFields

Create a Tracking FieldCreate a Tracking Field on your Zoom account

Catalog: Zoom

Schema: TrackingField

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

Select Zoom API URL: `/v2/tracking_fields`

Insert Zoom API URL: `/v2/tracking_fields`

Update Zoom API URL: `/v2/tracking_fields`

Delete Zoom API URL: `/v2/tracking_fields`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /v2/tracking_fields`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function postV2TrackingFields. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Tracking Field

Table Function Columns

The columns of the table function `postV2TrackingFields` 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
field	string		☐	Tracking Field Name
id	string		☐	Tracking Field ID
required	boolean		☐	Tracking Field Required
RESULT	string		☐	Outcome of operation as single plain text column.
visible	boolean		☐	Tracking Field Visible

15.1.4 V2TrackingFields

List Tracking Fields.List Tracking Fields on your Zoom account.

Catalog: Zoom

Schema: TrackingField

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

Select Zoom API URL: `/v2/tracking_fields`

Insert Zoom API URL: `/v2/tracking_fields`

Update Zoom API URL: `/v2/tracking_fields`

Delete Zoom API URL: `/v2/tracking_fields`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /v2/tracking_fields`

Table Columns

The columns of the table `V2TrackingFields` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
total_records	int64		☐	The number of all records available across pages

15.1.5 V2TrackingFieldsByFieldId

Retrieve a tracking fieldRetrieve a tracking field

Catalog: Zoom

Schema: TrackingField

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

Select Zoom API URL: `/v2/tracking_fields/{fieldId}`

Insert Zoom API URL: `/v2/tracking_fields/{fieldId}`

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

Update Zoom API URL: `/v2/tracking_fields/{fieldId}`

Delete Zoom API URL: `/v2/tracking_fields/{fieldId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /v2/tracking_fields/{fieldId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function V2TrackingFieldsByFieldId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

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

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

Name	Data Type	Required	Default Value	Documentation
fieldId	string	☑		The Tracking Field ID

Table Function Columns

The columns of the table function `V2TrackingFieldsByFieldId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
field	string		□	Tracking Field Name
id	string		□	Tracking Field ID
required	boolean		□	Tracking Field Required
visible	boolean		□	Tracking Field Visible

16 Schema: TSP

16.1 Tables

16.1.1 deleteUsersByUserIdTspTspId

Delete a user's TSP accountDelete a user's TSP account

Catalog: Zoom

Schema: TSP

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

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

Select Zoom API URL: `/users/{userId}/tsp/{tspId}`

Insert Zoom API URL: `/users/{userId}/tsp/{tspId}`

Update Zoom API URL: `/users/{userId}/tsp/{tspId}`

Delete Zoom API URL: `/users/{userId}/tsp/{tspId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}/tsp/{tspId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
tspId	string	☑		TSP account index
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

16.1.2 patchTsp

Update account's TSP informationUpdate TSP information on account level

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/tsp`

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

Insert Zoom API URL: `/tsp`

Update Zoom API URL: `/tsp`

Delete Zoom API URL: `/tsp`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /tsp`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		TSP Account

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

16.1.3 patchUsersByUserIdTspTspId

Update a TSP accountUpdate a user's TSP account

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/users/{userId}/tsp/{tspId}`

Insert Zoom API URL: `/users/{userId}/tsp/{tspId}`

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

Update Zoom API URL: `/users/{userId}/tsp/{tspId}`

Delete Zoom API URL: `/users/{userId}/tsp/{tspId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /users/{userId}/tsp/{tspId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		TSP Account
tspId	string	☑		TSP account index
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

16.1.4 postUsersByUserId_Dial_in_numbersTsp

Add a user's TSP accountAdd a user's TSP account

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/users/{userId}/tsp`

Insert Zoom API URL: `/users/{userId}/tsp`

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

Update Zoom API URL: `/users/{userId}/tsp`

Delete Zoom API URL: `/users/{userId}/tsp`

Field Selection Method: NotRequired

Base Path: `dial_in_numbers[*]`

Select Zoom API Operation: `post /users/{userId}/tsp`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		TSP Account
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
code	string(6)		☐	Country Code
conference_code	string(16)		☑	Conference code, numeric value, length is less than 16.
country_label	string(10)		☐	Country Label, if passed, will display in place of code.
leader_pin	string(16)		☑	Leader PIN, numeric value, length is less than 16.
number	string(16)		☐	Dial-in number, length is less than 16.
RESULT	string		☐	Outcome of operation as single plain text column.
type	string		☐	Dial-in number type.

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

16.1.5 postUsersByUserIdTsp

Add a user's TSP accountAdd a user's TSP account

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/users/{userId}/tsp`

Insert Zoom API URL: `/users/{userId}/tsp`

Update Zoom API URL: `/users/{userId}/tsp`

Delete Zoom API URL: `/users/{userId}/tsp`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /users/{userId}/tsp`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
body	string	☑		TSP Account
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
conference_code	string(16)		☑	Conference code, numeric value, length is less than 16.
leader_pin	string(16)		☑	Leader PIN, numeric value, length is less than 16.

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

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

16.1.6 Tsp

Retrieve account's TSP informationRetrieve TSP information on account level

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/tsp`

Insert Zoom API URL: `/tsp`

Update Zoom API URL: `/tsp`

Delete Zoom API URL: `/tsp`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /tsp`

Table Columns

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

Name	Data Type	Label	Required	Documentation
enable	boolean		☐	Enable 3rd party audio conferencing for account users
tsp_provider	string		☐	3rd party audio conferencing provider

16.1.7 Tsp_Dial_in_numbers

Retrieve account's TSP informationRetrieve TSP information on account level

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/tsp`

Insert Zoom API URL: `/tsp`

Update Zoom API URL: `/tsp`

Delete Zoom API URL: `/tsp`

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

Field Selection Method: NotRequired

Base Path: `dial_in_numbers[*]`

Select Zoom API Operation: `get /tsp`

Table Columns

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

Name	Data Type	Label	Required	Documentation
code	string		☐	Country Code
enable	boolean		☐	Enable 3rd party audio conferencing for account users
number	string		☐	Dial-in number, length is less than 16
tsp_provider	string		☐	3rd party audio conferencing provider
type	string		☐	

16.1.8 UsersByUserId_Tsp_accountsDial_in_numbersTsp

List user's TSP accountsList user's TSP accounts

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/users/{userId}/tsp`

Insert Zoom API URL: `/users/{userId}/tsp`

Update Zoom API URL: `/users/{userId}/tsp`

Delete Zoom API URL: `/users/{userId}/tsp`

Field Selection Method: NotRequired

Base Path: `tsp_accounts[*].dial_in_numbers[*]`

Select Zoom API Operation: `get /users/{userId}/tsp`

Parameters of Table Function

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

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

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

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

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

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
code	string(6)		☐	Country Code
conference_code	string(16)		☑	Conference code, numeric value, length is less than 16.
country_label	string(10)		☐	Country Label, if passed, will display in place of code.
leader_pin	string(16)		☑	Leader PIN, numeric value, length is less than 16.
number	string(16)		☐	Dial-in number, length is less than 16.
type	string		☐	Dial-in number type.

16.1.9 UsersByUserIdTsp

List user's TSP accountsList user's TSP accounts

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/users/{userId}/tsp`

Insert Zoom API URL: `/users/{userId}/tsp`

Update Zoom API URL: `/users/{userId}/tsp`

Delete Zoom API URL: `/users/{userId}/tsp`

Field Selection Method: NotRequired

Base Path: `tsp_accounts[*]`

Select Zoom API Operation: `get /users/{userId}/tsp`

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

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

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

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
conference_code	string(16)		☑	Conference code, numeric value, length is less than 16.
leader_pin	string(16)		☑	Leader PIN, numeric value, length is less than 16.

16.1.10 UsersByUserIdTsp_Dial_in_numbersTspId

Retrieve a user's TSP accountRetrieve a user's TSP account

Catalog: Zoom

Schema: TSP

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

Select Zoom API URL: `/users/{userId}/tsp/{tspId}`

Insert Zoom API URL: `/users/{userId}/tsp/{tspId}`

Update Zoom API URL: `/users/{userId}/tsp/{tspId}`

Delete Zoom API URL: `/users/{userId}/tsp/{tspId}`

Field Selection Method: NotRequired

Base Path: `dial_in_numbers[*]`

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

Select Zoom API Operation: `get /users/{userId}/tsp/{tspId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
tspId	string	☑		TSP account index
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
code	string(6)		☐	Country Code
conference_code	string(16)		☑	Conference code, numeric value, length is less than 16.
country_label	string(10)		☐	Country Label, if passed, will display in place of code.
leader_pin	string(16)		☑	Leader PIN, numeric value, length is less than 16.
number	string(16)		☐	Dial-in number, length is less than 16.
type	string		☐	Dial-in number type.

16.1.11 UsersByUserIdTspTspId

Retrieve a user's TSP accountRetrieve a user's TSP account

Catalog: Zoom

Schema: TSP

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

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

Select Zoom API URL: `/users/{userId}/tsp/{tspId}`

Insert Zoom API URL: `/users/{userId}/tsp/{tspId}`

Update Zoom API URL: `/users/{userId}/tsp/{tspId}`

Delete Zoom API URL: `/users/{userId}/tsp/{tspId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/{userId}/tsp/{tspId}`

Parameters of Table Function

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

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

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

Name	Data Type	Required	Default Value	Documentation
tspId	string	☑		TSP account index
userId	string	☑		The user ID or email address

Table Function Columns

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

Name	Data Type	Label	Required	Documentation
conference_code	string(16)		☑	Conference code, numeric value, length is less than 16.
leader_pin	string(16)		☑	Leader PIN, numeric value, length is less than 16.

17 Schema: Users

17.1 Tables

17.1.1 deleteUsersByUserId

Delete a userDelete a user on your account

Catalog: Zoom

Schema: Users

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

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

Select Zoom API URL: `/users/{userId}`

Insert Zoom API URL: `/users/{userId}`

Update Zoom API URL: `/users/{userId}`

Delete Zoom API URL: `/users/{userId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteUsersByUserId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
action	string	☐	disassociate	Delete action type (Values: disassociate, delete)
transfer_email	string	☐		Transfer email
transfer_meeting	boolean	☐		Transfer meeting
transfer_recording	boolean	☐		Transfer recording
transfer_webinar	boolean	☐		Transfer webinar
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `deleteUsersByUserId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.2 deleteUsersByUserIdAssistants

Delete a user's assistantsDelete all of a user'sassitants

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/assistants`

Insert Zoom API URL: `/users/{userId}/assistants`

Update Zoom API URL: `/users/{userId}/assistants`

Delete Zoom API URL: `/users/{userId}/assistants`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}/assistants`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteUsersByUserIdAssistants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `deleteUsersByUserIdAssistants` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.3 deleteUsersByUserIdAssistantsAssistantId

Delete a user's assistantDelete one of a user's assistants

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/assistants/{assistantId}`

Insert Zoom API URL: `/users/{userId}/assistants/{assistantId}`

Update Zoom API URL: `/users/{userId}/assistants/{assistantId}`

Delete Zoom API URL: `/users/{userId}/assistants/{assistantId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}/assistants/{assistantId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteUsersByUserIdAssistantsAssistantId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
assistantId	string	☑		Assistant's ID
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `deleteUsersByUserIdAssistantsAssistantId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.4 deleteUsersByUserIdSchedulers

Delete a user's schedulersDelete all of a user'schedulers

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/schedulers`

Insert Zoom API URL: `/users/{userId}/schedulers`

Update Zoom API URL: `/users/{userId}/schedulers`

Delete Zoom API URL: `/users/{userId}/schedulers`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}/schedulers`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteUsersByUserIdSchedulers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `deleteUsersByUserIdSchedulers` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.5 deleteUsersByUserIdSchedulersSchedulerId

Delete a user's schedulerDelete one of a user's schedulers

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/schedulers/{schedulerId}`

Insert Zoom API URL: `/users/{userId}/schedulers/{schedulerId}`

Update Zoom API URL: `/users/{userId}/schedulers/{schedulerId}`

Delete Zoom API URL: `/users/{userId}/schedulers/{schedulerId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}/schedulers/{schedulerId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteUsersByUserIdSchedulersSchedulerId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
schedulerId	string	☑		Scheduler's ID
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `deleteUsersByUserIdSchedulersSchedulerId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.6 deleteUsersByUserIdToken

Revoke a user's SSO tokenRevoke a user's SSO token

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/token`

Insert Zoom API URL: `/users/{userId}/token`

Update Zoom API URL: `/users/{userId}/token`

Delete Zoom API URL: `/users/{userId}/token`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /users/{userId}/token`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteUsersByUserIdToken. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `deleteUsersByUserIdToken` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.7 patchUsersByUserId

Update a userUpdate a user on your account

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}`

Insert Zoom API URL: `/users/{userId}`

Update Zoom API URL: `/users/{userId}`

Delete Zoom API URL: `/users/{userId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /users/{userId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function patchUsersByUserId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		User
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `patchUsersByUserId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.8 patchUsersByUserIdSettings

Update a user's settingsUpdate a user's settings

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/settings`

Insert Zoom API URL: `/users/{userId}/settings`

Update Zoom API URL: `/users/{userId}/settings`

Delete Zoom API URL: `/users/{userId}/settings`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /users/{userId}/settings`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function patchUsersByUserIdSettings. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		User Settings
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `patchUsersByUserIdSettings` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

17.1.9 postUsers

Create a userCreate a user on your account

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users`

Insert Zoom API URL: `/users`

Update Zoom API URL: `/users`

Delete Zoom API URL: `/users`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /users`

Table Function Columns

The columns of the table function `postUsers` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
email	string		☐	User's email address
first_name	string(64)		☐	User's first name
id	string		☐	User ID
last_name	string(64)		☐	User's last name
RESULT	string		☐	Outcome of operation as single plain text column.
type	int64		☐	User's type

17.1.10 postUsersByUserIdAssistants

Add assistantsAdd assistants to a user

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/assistants`

Insert Zoom API URL: `/users/{userId}/assistants`

Update Zoom API URL: `/users/{userId}/assistants`

Delete Zoom API URL: `/users/{userId}/assistants`

Field Selection Method: NotRequired

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Select Zoom API Operation: `post /users/{userId}/assistants`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function postUsersByUserIdAssistants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		User assistant
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `postUsersByUserIdAssistants` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
add_at	datetime		☐	
ids	string		☐	User ID
RESULT	string		☐	Outcome of operation as single plain text column.

17.1.11 postUsersByUserIdPicture

Upload a user's pictureUpload a user's profile picture

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/picture`

Insert Zoom API URL: `/users/{userId}/picture`

Update Zoom API URL: `/users/{userId}/picture`

Delete Zoom API URL: `/users/{userId}/picture`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Field Selection Method: NotRequired

Select Zoom API Operation: `post /users/{userId}/picture`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function postUsersByUserIdPicture. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
pic_file	byte[]	☑		User picture file, must be a jpg/jpeg file
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `postUsersByUserIdPicture` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

17.1.12 putUsersByUserIdEmail

Update a user's emailUpdate a user's email

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/email`

Insert Zoom API URL: `/users/{userId}/email`

Update Zoom API URL: `/users/{userId}/email`

Delete Zoom API URL: `/users/{userId}/email`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Field Selection Method: NotRequired

Select Zoom API Operation: `put /users/{userId}/email`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function putUsersByUserIdEmail. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `putUsersByUserIdEmail` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

17.1.13 putUsersByUserIdPassword

Update a user's passwordUpdate a user's password

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/password`

Insert Zoom API URL: `/users/{userId}/password`

Update Zoom API URL: `/users/{userId}/password`

Delete Zoom API URL: `/users/{userId}/password`

Field Selection Method: NotRequired

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Select Zoom API Operation: `put /users/{userId}/password`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function putUsersByUserIdPassword. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `putUsersByUserIdPassword` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

17.1.14 putUsersByUserIdStatus

Update a user's statusUpdate a user's status

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/status`

Insert Zoom API URL: `/users/{userId}/status`

Update Zoom API URL: `/users/{userId}/status`

Delete Zoom API URL: `/users/{userId}/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /users/{userId}/status`

(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 putUsersByUserIdStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `putUsersByUserIdStatus` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

17.1.15 Users

List UsersList users on your account

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users`

Insert Zoom API URL: `/users`

Update Zoom API URL: `/users`

Delete Zoom API URL: `/users`

Field Selection Method: NotRequired

Base Path: `users[*]`

Select Zoom API Operation: `get /users`

(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 Users. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
status	string	☐	active	User status (Values: active, inactive, pending)

Table Function Columns

The columns of the table function `Users` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
created_at	datetime		☐	User create time
dept	string		☐	Department
email	string		☑	User's email address
first_name	string(64)		☐	User's first name
id	string		☐	User ID
last_client_version	string		☐	User last login client version
last_login_time	datetime		☐	User last login time
last_name	string(64)		☐	User's last name
pmi	string		☐	Personal Meeting ID
timezone	string		☐	Time Zone
type	int64		☑	User's type

17.1.16 Users_Group_idsByUserId

Retrieve a userRetrieve a user on your account

Catalog: Zoom

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}`

Insert Zoom API URL: `/users/{userId}`

Update Zoom API URL: `/users/{userId}`

Delete Zoom API URL: `/users/{userId}`

Field Selection Method: NotRequired

Base Path: `group_ids[*]`

Select Zoom API Operation: `get /users/{userId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Users_Group_idsByUserId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
login_type	string	☐		Values: 0, 1, 99, 100, 101.
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `Users_Group_idsByUserId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
account_id	string		☐	
cms_user_id	string		☐	
created_at	datetime		☐	User create time
dept	string		☐	Department
email	string		☑	User's email address
first_name	string(64)		☐	User's first name

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
host_key	string		☐	
id	string		☐	User ID
language	string		☐	
last_client_version	string		☐	User last login client version
last_login_time	datetime		☐	User last login time
last_name	string(64)		☐	User's last name
personal_meeting_url	string		☐	
pic_url	string		☐	
pmi	string		☐	Personal Meeting ID
timezone	string		☐	Time Zone
type	int64		☑	User's type
use_pmi	boolean		☐	
value	string		☐	
vanity_url	string		☐	
verified	int64		☐	

17.1.17 Users_Im_group_idsByUserId

Retrieve a userRetrieve a user on your account

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}`

Insert Zoom API URL: `/users/{userId}`

Update Zoom API URL: `/users/{userId}`

Delete Zoom API URL: `/users/{userId}`

Field Selection Method: NotRequired

Base Path: `im_group_ids[*]`

Select Zoom API Operation: `get /users/{userId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Users_Im_group_idsByUserId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
login_type	string	☐		Values: 0, 1, 99, 100, 101.
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `Users_Im_group_idsByUserId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
account_id	string		☐	
cms_user_id	string		☐	
created_at	datetime		☐	User create time
dept	string		☐	Department
email	string		☑	User's email address
first_name	string(64)		☐	User's first name
host_key	string		☐	
id	string		☐	User ID
language	string		☐	
last_client_version	string		☐	User last login client version
last_login_time	datetime		☐	User last login time
last_name	string(64)		☐	User's last name
personal_meeting_url	string		☐	
pic_url	string		☐	
pmi	string		☐	Personal Meeting ID
timezone	string		☐	Time Zone
type	int64		☑	User's type
use_pmi	boolean		☐	
value	string		☐	
vanity_url	string		☐	
verified	int64		☐	

17.1.18 Users_UsersGroup_ids

List UsersList users on your account

Catalog: Zoom

Schema: Users

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users`

Insert Zoom API URL: `/users`

Update Zoom API URL: `/users`

Delete Zoom API URL: `/users`

Field Selection Method: NotRequired

Base Path: `users[*].group_ids[*]`

Select Zoom API Operation: `get /users`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Users_UsersGroup_ids. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
status	string	☐	active	User status (Values: active, inactive, pending)

Table Function Columns

The columns of the table function `Users_UsersGroup_ids` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
created_at	datetime		☐	User create time
dept	string		☐	Department
email	string		☑	User's email address
first_name	string(64)		☐	User's first name

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
id	string		☐	User ID
last_client_version	string		☐	User last login client version
last_login_time	datetime		☐	User last login time
last_name	string(64)		☐	User's last name
pmi	string		☐	Personal Meeting ID
timezone	string		☐	Time Zone
type	int64		☑	User's type
value	string		☐	

17.1.19 Users_Usersim_group_ids

List UsersList users on your account

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users`

Insert Zoom API URL: `/users`

Update Zoom API URL: `/users`

Delete Zoom API URL: `/users`

Field Selection Method: NotRequired

Base Path: `users[*].im_group_ids[*]`

Select Zoom API Operation: `get /users`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Users_Usersim_group_ids. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
status	string	☐	active	User status (Values: active, inactive, pending)

Table Function Columns

The columns of the table function `Users_UsersIm_group_ids` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
created_at	datetime		☐	User create time
dept	string		☐	Department
email	string		☑	User's email address
first_name	string(64)		☐	User's first name
id	string		☐	User ID
last_client_version	string		☐	User last login client version
last_login_time	datetime		☐	User last login time
last_name	string(64)		☐	User's last name
pmi	string		☐	Personal Meeting ID
timezone	string		☐	Time Zone
type	int64		☑	User's type
value	string		☐	

17.1.20 UsersByUserId

Retrieve a userRetrieve a user on your account

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}`

Insert Zoom API URL: `/users/{userId}`

Update Zoom API URL: `/users/{userId}`

Delete Zoom API URL: `/users/{userId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/{userId}`

(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 UsersByUserId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
login_type	string	☐		Values: 0, 1, 99, 100, 101.
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `UsersByUserId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
account_id	string		☐	
cms_user_id	string		☐	
created_at	datetime		☐	User create time
dept	string		☐	Department
email	string		☑	User's email address
first_name	string(64)		☐	User's first name
host_key	string		☐	
id	string		☐	User ID
language	string		☐	
last_client_version	string		☐	User last login client version
last_login_time	datetime		☐	User last login time
last_name	string(64)		☐	User's last name
personal_meeting_url	string		☐	
pic_url	string		☐	
pmi	string		☐	Personal Meeting ID
timezone	string		☐	Time Zone
type	int64		☑	User's type
use_pmi	boolean		☐	
vanity_url	string		☐	

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
verified	int64		☐	

17.1.21 UsersByUserIdAssistants

List a user's assistantsList a user's assistants

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/assistants`

Insert Zoom API URL: `/users/{userId}/assistants`

Update Zoom API URL: `/users/{userId}/assistants`

Delete Zoom API URL: `/users/{userId}/assistants`

Field Selection Method: NotRequired

Base Path: `assistants[*]`

Select Zoom API Operation: `get /users/{userId}/assistants`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdAssistants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `UsersByUserIdAssistants` 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
email	string		☐	User email address. Must have id or email, if given id, the email is ignored.
id	string		☐	User ID

17.1.22 UsersByUserIdPermissions

Retrieve a user's permissionsRetrieve a user's permissions

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/permissions`

Insert Zoom API URL: `/users/{userId}/permissions`

Update Zoom API URL: `/users/{userId}/permissions`

Delete Zoom API URL: `/users/{userId}/permissions`

Field Selection Method: NotRequired

Base Path: `permissions[*]`

Select Zoom API Operation: `get /users/{userId}/permissions`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdPermissions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function `UsersByUserIdPermissions` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
TEXT	string		☐	

17.1.23 UsersByUserIdSchedulers

List a user's schedulersList a user's schedulers

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/schedulers`

Insert Zoom API URL: `/users/{userId}/schedulers`

Update Zoom API URL: `/users/{userId}/schedulers`

Delete Zoom API URL: `/users/{userId}/schedulers`

Field Selection Method: NotRequired

Base Path: `assistants[*]`

Select Zoom API Operation: `get /users/{userId}/schedulers`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdSchedulers. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
userId	string	☑		The user ID or email address

Table Function Columns

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function `UsersByUserIdSchedulers` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
email	string		☐	User email address.
id	string		☐	User ID

17.1.24 UsersByUserIdSettings

Retrieve a user's settingsRetrieve a user's settings

Catalog: Zoom

Schema: Users

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/settings`

Insert Zoom API URL: `/users/{userId}/settings`

Update Zoom API URL: `/users/{userId}/settings`

Delete Zoom API URL: `/users/{userId}/settings`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/{userId}/settings`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdSettings. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
login_type	string	☐		Values: 0, 1, 99, 100, 101.
userId	string	☑		The user ID or email address

Table Function Columns

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

The columns of the table function `UsersByUserIdSettings` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
email_notification_alternat-ive_host_reminder	boolean		☐	When an alternative host is set or removed from a meeting
email_notification_cancel_meet-ing_reminder	boolean		☐	When a meeting is cancelled
email_notification_jbh_reminder	boolean		☐	When attendees join meeting be-fore host
feature_cn_meeting	boolean		☐	CN meeting feature
feature_in_meeting	boolean		☐	IN meeting feature
feature_large_meeting_capacity	int64		☐	Large meeting capacity, can be 100, 200, 300 or 500, depends on if having related large meeting capacity plan subscription or not
feature_large_meeting	boolean		☐	Large meting feature
feature_meeting_capacity	int64		☐	User's meeting capacity
feature_w ebinar_capacity	int64		☐	Webinar capacity, can be 100, 500, 1000, 3000, 5000 or 10000, depends on if having related w e-binar capacity plan subscription or not
feature_w ebinar	boolean		☐	Webinar feature
in_meeting_allow _live_streaming	boolean		☐	Allow live streaming
in_meeting_annotation	boolean		☐	Annotation
in_meeting_attendee_on_hold	boolean		☐	Allow host to put attendee on hold
in_meeting_attention_tracking	boolean		☐	Attention tracking
in_meeting_auto_saving_chat	boolean		☐	Auto saving chats
in_meeting_breakout_room	boolean		☐	Breakout room
in_meeting_chat	boolean		☐	Chat
in_meeting_closed_caption	boolean		☐	Closed caption
in_meeting_co_host	boolean		☐	Co-host
in_meeting_custom_live_streaming	boolean		☐	Custom live streaming
in_meeting_custom_service_in-structions	string		☐	Custom service instructions
in_meeting_e2e_encryption	boolean		☐	End-to-end encryption
in_meeting_entry_exit_chime	string		☐	Play sound on join/leave
in_meeting_far_end_camera_con-trol	boolean		☐	Far end camera control
in_meeting_feedback	boolean		☐	Feedback to Zoom
in_meeting_file_transfer	boolean		☐	File transfer
in_meeting_group_hd	boolean		☐	Group HD video
in_meeting_non_verbal_feedback	boolean		☐	Non-verbal feedback
in_meeting_polling	boolean		☐	Polling
in_meeting_private_chat	boolean		☐	Private chat

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
in_meeting_record_play_voice	boolean		☐	Record and play their own voice
in_meeting_remote_control	boolean		☐	Remote control
in_meeting_remote_support	boolean		☐	Remote support
in_meeting_share_dual_camera	boolean		☐	Share dual camera (Deprecated)
in_meeting_virtual_background	boolean		☐	Virtual background
in_meeting_waiting_room	boolean		☐	Waiting room
in_meeting_workplace_by_facebook	boolean		☐	Workplace by facebook
recording_auto_delete_cmr_days	int64		☐	A specified number of days of auto delete cloud recordings
recording_auto_delete_cmr	boolean		☐	Auto delete cloud recordings
recording_auto_recording	string		☐	Automatic recording
recording_cloud_recording	boolean		☐	Cloud recording
recording_local_recording	boolean		☐	Local recording
recording_record_audio_file	boolean		☐	Record an audio only file
recording_record_gallery_view	boolean		☐	Record the gallery view
recording_record_speaker_view	boolean		☐	Record the active speaker view
recording_recording_audio_transcript	boolean		☐	Audio transcript
recording_save_chat_text	boolean		☐	Save chat text from the meeting
recording_show_timestamp	boolean		☐	Show timestamp on video
schedule_meeting_audio_type	string		☐	Determine how participants can join the audio portion of the meeting
schedule_meeting_force_pmi_jbh_password	boolean		☐	Require a password for Personal Meetings if attendees can join before host
schedule_meeting_host_video	boolean		☐	Host video
schedule_meeting_join_before_host	boolean		☐	Join before host
schedule_meeting_participants_video	boolean		☐	Participants video
schedule_meeting_pstn_password_protected	boolean		☐	Generate and require password for participants joining by phone
telephony_audio_conference_info	string(2048)		☐	3rd party audio conference info
telephony_show_international_numbers_link	boolean		☐	Show international numbers link on the invitation email
telephony_third_party_audio	boolean		☐	3rd party audio conference

17.1.25 UsersByUserIdToken

Retrieve a user's tokenRetrieve a user's token

Catalog: Zoom

Schema: Users

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/token`

Insert Zoom API URL: `/users/{userId}/token`

Update Zoom API URL: `/users/{userId}/token`

Delete Zoom API URL: `/users/{userId}/token`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/{userId}/token`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdToken. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
type	string	☐		User token type (Values: token, zpk, zak)
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `UsersByUserIdToken` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
token	string		☐	User ID

17.1.26 UsersEmail

Check a user's emailCheck if the user email exists

Catalog: Zoom

Schema: Users

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/email`

Insert Zoom API URL: `/users/email`

Update Zoom API URL: `/users/email`

Delete Zoom API URL: `/users/email`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/email`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersEmail. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
email	string	☑		Zoom w ork email

Table Function Columns

The columns of the table function `UsersEmail` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
existed_email	boolean		☐	

17.1.27 UsersVanityName

Check a user's personal meeting room nameCheck if the user's personal meeting room name exists

Catalog: Zoom

Schema: Users

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/vanity_name`

Insert Zoom API URL: `/users/vanity_name`

Update Zoom API URL: `/users/vanity_name`

Delete Zoom API URL: `/users/vanity_name`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/vanity_name`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersVanityName. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
vanity_name	string	☑		Personal meeting room name

Table Function Columns

The columns of the table function `UsersVanityName` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
existed	boolean		☐	

17.1.28 UsersZpk

Verify a user's zpk (DeprecatedCheck if the zpk is expired. The zpk is used to authenticate a user.

Catalog: Zoom

Schema: Users

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/zpk`

Insert Zoom API URL: `/users/zpk`

Update Zoom API URL: `/users/zpk`

Delete Zoom API URL: `/users/zpk`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /users/zpk`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersZpk. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
zpk	string	☑		User zpk

Table Function Columns

The columns of the table function `UsersZpk` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
expire_in	int64		☐	

18 Schema: Webhooks

18.1 Tables

18.1.1 deleteWebhooksByWebhookId

Delete a webhookDelete a webhook

Catalog: Zoom

Schema: Webhooks

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webhooks/{webhookId}`

Insert Zoom API URL: `/webhooks/{webhookId}`

Update Zoom API URL: `/webhooks/{webhookId}`

Delete Zoom API URL: `/webhooks/{webhookId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /webhooks/{webhookId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function deleteWebhooksByWebhookId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webhookId	string	☑		The webhook ID

Table Function Columns

The columns of the table function `deleteWebhooksByWebhookId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

18.1.2 patchWebhooksByWebhookId

Update a webhookUpdate a webhook

Catalog: Zoom

Schema: Webhooks

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webhooks/{webhookId}`

Insert Zoom API URL: `/webhooks/{webhookId}`

Update Zoom API URL: `/webhooks/{webhookId}`

Delete Zoom API URL: `/webhooks/{webhookId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /webhooks/{webhookId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function patchWebhooksByWebhookId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Webhook
webhookId	string	☑		The webhook ID

Table Function Columns

The columns of the table function `patchWebhooksByWebhookId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

18.1.3 patchWebhooksOptions

Switch webhook versionSwitch webhook version

Catalog: Zoom

Schema: Webhooks

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webhooks/options`

Insert Zoom API URL: `/webhooks/options`

Update Zoom API URL: `/webhooks/options`

Delete Zoom API URL: `/webhooks/options`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /webhooks/options`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function patchWebhooksOptions. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		

Table Function Columns

The columns of the table function `patchWebhooksOptions` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

18.1.4 postWebhooks

Create a webhookCreate a webhook for a account

Catalog: Zoom

Schema: Webhooks

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webhooks`

Insert Zoom API URL: `/webhooks`

Update Zoom API URL: `/webhooks`

Delete Zoom API URL: `/webhooks`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /webhooks`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function postWebhooks. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a predefined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Webhook

Table Function Columns

The columns of the table function `postWebhooks` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
auth_password	string(64)		☑	Webhook auth password
auth_user	string(128)		☑	Webhook auth user name
created_at	datetime		☐	Webhook create time
RESULT	string		☐	Outcome of operation as single plain text column.
url	string(256)		☑	Webhook endpoint
webhook_id	string		☐	Webhook Id

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

18.1.5 Webhooks

List webhooksList webhooks for a account

Catalog: Zoom

Schema: Webhooks

This is a read-only table. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webhooks`

Insert Zoom API URL: `/webhooks`

Update Zoom API URL: `/webhooks`

Delete Zoom API URL: `/webhooks`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webhooks`

Table Columns

The columns of the table `Webhooks` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
total_records	int64		☐	The number of all records available across pages

18.1.6 WebhooksByWebhookId

Retrieve a webhookRetrieve a webhook

Catalog: Zoom

Schema: Webhooks

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webhooks/{webhookId}`

Insert Zoom API URL: `/webhooks/{webhookId}`

Update Zoom API URL: `/webhooks/{webhookId}`

Delete Zoom API URL: `/webhooks/{webhookId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webhooks/{webhookId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WebhooksByWebhookId. A value must be provided at 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 treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webhookId	string	☑		The webhook ID

Table Function Columns

The columns of the table function `WebhooksByWebhookId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
auth_password	string(64)		☑	Webhook auth password
auth_user	string(128)		☑	Webhook auth user name
created_at	datetime		☐	Webhook create time
url	string(256)		☑	Webhook endpoint
webhook_id	string		☐	Webhook Id

19 Schema: Webinars

19.1 Tables

19.1.1 deleteWebinarsByWebinarId

Delete a webinarDelete a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}`

Insert Zoom API URL: `/webinars/{webinarId}`

Update Zoom API URL: `/webinars/{webinarId}`

Delete Zoom API URL: `/webinars/{webinarId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /webinars/{webinarId}`

(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 deleteWebinarsByWebinarId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
occurrence_id	string	☐		The meeting occurrence ID
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `deleteWebinarsByWebinarId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.2 deleteWebinarsByWebinarIdPanelists

Remove a webinar's panelistsRemove all panelists from a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/panelists`

Insert Zoom API URL: `/webinars/{webinarId}/panelists`

Update Zoom API URL: `/webinars/{webinarId}/panelists`

Delete Zoom API URL: `/webinars/{webinarId}/panelists`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /webinars/{webinarId}/panelists`

(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 deleteWebinarsByWebinarIdPanelists. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `deleteWebinarsByWebinarIdPanelists` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.3 deleteWebinarsByWebinarIdPanelistsPanelistId

Remove a webinar panelistRemove a panelist from a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/panelists/{panelistId}`

Insert Zoom API URL: `/webinars/{webinarId}/panelists/{panelistId}`

Update Zoom API URL: `/webinars/{webinarId}/panelists/{panelistId}`

Delete Zoom API URL: `/webinars/{webinarId}/panelists/{panelistId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /webinars/{webinarId}/panelists/{panelistId}`

(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 deleteWebinarsByWebinarIdPanelistsPanelistId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
panelistId	int64	☑		The panelist ID
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `deleteWebinarsByWebinarIdPanelistsPanelistId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.4 deleteWebinarsByWebinarIdPollsPollId

Delete a webinar's PollDelete a webinar's Poll

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Insert Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Update Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Delete Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `delete /webinars/{webinarId}/polls/{pollId}`

(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 deleteWebinarsByWebinarIdPollsPollId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
pollId	string	☑		The poll ID
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `deleteWebinarsByWebinarIdPollsPollId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.5 PastWebinarsByWebinarIdInstances

List of ended webinar instancesList of ended webinar instances

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/past_webinars/{webinarId}/instances`

Insert Zoom API URL: `/past_webinars/{webinarId}/instances`

Update Zoom API URL: `/past_webinars/{webinarId}/instances`

Delete Zoom API URL: `/past_webinars/{webinarId}/instances`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /past_webinars/{webinarId}/instances`

(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 PastWebinarsByWebinarIdInstances. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `PastWebinarsByWebinarIdInstances` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
DUMMY	string(1)		☑	Default column added since the specification specifies that no data is returned.

19.1.6 patchWebinarsByWebinarId

Update a webinarUpdate a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}`

Insert Zoom API URL: `/webinars/{webinarId}`

Update Zoom API URL: `/webinars/{webinarId}`

Delete Zoom API URL: `/webinars/{webinarId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `patch /webinars/{webinarId}`

(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 patchWebinarsByWebinarId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Webinar
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `patchWebinarsByWebinarId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.7 postUsersByUserIdWebinars

Create a webinarCreate a webinar for a user <aside>The expiration time of start_url is two hours. But for API users, the expiration time is 90 days.</aside>

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/webinars`

Insert Zoom API URL: `/users/{userId}/webinars`

Update Zoom API URL: `/users/{userId}/webinars`

Delete Zoom API URL: `/users/{userId}/webinars`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /users/{userId}/webinars`

(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 postUsersByUserIdWebinars. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		User
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `postUsersByUserIdWebinars` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
agenda	string		☐	Webinar agenda
created_at	datetime		☐	Create time
duration	int64		☐	Webinar duration
host_id	string		☐	ID of the user set as host of Webinar
id	string		☐	Webinar ID, also known as Webinar number
join_url	string		☐	Join url
RESULT	string		☐	Outcome of operation as single plain text column.
settings_allow_multiple_devices	boolean		☐	Allow attendees to join from multiple devices
settings_alternative_hosts	string		☐	Alternative hosts emails or IDs. Multiple values separated by comma.
settings_approval_type	int64		☐	
settings_audio	string		☐	Determine how participants can join the audio portion of the meeting
settings_auto_recording	string		☐	
settings_close_registration	boolean		☐	Close registration after event date

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
settings_enforce_login_domains	string		☐	Only signed-in users with specified domains can join meetings
settings_enforce_login	boolean		☐	Only signed-in users can join this meeting
settings_hd_video	boolean		☐	Default to HD Video
settings_host_video	boolean		☐	Start video when host joins webinar
settings_on_demand	boolean		☐	Make the webinar on-demand
settings_panelists_video	boolean		☐	Start video when panelists join webinar
settings_practice_session	boolean		☐	Enable Practice Session
settings_registration_type	int64		☐	Registration type. Used for recurring webinar with fixed time only.
settings_show_share_button	boolean		☐	Show social share buttons on registration page
start_time	datetime		☐	Webinar start time
start_url	string		☐	Start url
timezone	string		☐	Timezone to format start_time
topic	string		☐	Webinar topic
type	int64		☐	Webinar Type
uuid	string		☐	Webinar unique ID

19.1.8 postWebinarsByWebinarIdPanelists

Add a webinar panelistAdd panelist to webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/panelists`

Insert Zoom API URL: `/webinars/{webinarId}/panelists`

Update Zoom API URL: `/webinars/{webinarId}/panelists`

Delete Zoom API URL: `/webinars/{webinarId}/panelists`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /webinars/{webinarId}/panelists`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function postWebinarsByWebinarIdPanelists. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `postWebinarsByWebinarIdPanelists` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	string		☐	Panelist ID
join_url	string		☐	Join URL for this panelist
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.9 postWebinarsByWebinarIdPolls

Create a webinar's pollCreate a poll for a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/polls`

Insert Zoom API URL: `/webinars/{webinarId}/polls`

Update Zoom API URL: `/webinars/{webinarId}/polls`

Delete Zoom API URL: `/webinars/{webinarId}/polls`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /webinars/{webinarId}/polls`

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 postWebinarsByWebinarIdPolls. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Webinar poll object
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `postWebinarsByWebinarIdPolls` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	string		☐	Webinar Poll ID
RESULT	string		☐	Outcome of operation as single plain text column.
status	string		☐	Status of the Webinar Poll
title	string		☐	Poll Title

19.1.10 postWebinarsByWebinarIdRegistrants

Add a webinar registrantAdd a registrant for a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/registrants`

Insert Zoom API URL: `/webinars/{webinarId}/registrants`

Update Zoom API URL: `/webinars/{webinarId}/registrants`

Delete Zoom API URL: `/webinars/{webinarId}/registrants`

Field Selection Method: NotRequired

Select Zoom API Operation: `post /webinars/{webinarId}/registrants`

(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 postWebinarsByWebinarIdRegistrants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
occurrence_ids	string	☐		Occurrence IDs, could get this value from Webinar Get API. Multiple value separated by comma.
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `postWebinarsByWebinarIdRegistrants` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	string		☐	Registrant ID
join_url	string		☐	Join URL for this registrant
registrant_id	string		☐	Registrant ID
RESULT	string		☐	Outcome of operation as single plain text column.
start_time	datetime		☐	Start time
topic	string		☐	Topic

19.1.11 putWebinarsByWebinarIdPollsPollId

Update a webinar's pollUpdate a webinar's poll

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Insert Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Update Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Delete Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /webinars/{webinarId}/polls/{pollId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function putWebinarsByWebinarIdPollsPollId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		Webinar Poll
pollId	string	☑		The poll ID
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `putWebinarsByWebinarIdPollsPollId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.12 putWebinarsByWebinarIdRegistrantsStatus

Update a webinar registrant's statusUpdate a webinar registrant's status

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/registrants/status`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Insert Zoom API URL: `/webinars/{webinarId}/registrants/status`

Update Zoom API URL: `/webinars/{webinarId}/registrants/status`

Delete Zoom API URL: `/webinars/{webinarId}/registrants/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /webinars/{webinarId}/registrants/status`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function putWebinarsByWebinarIdRegistrantsStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
occurrence_id	string	☐		The meeting occurrence ID
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `putWebinarsByWebinarIdRegistrantsStatus` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.13 putWebinarsByWebinarIdStatus

Update a webinar's statusUpdate a webinar's status

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/status`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Insert Zoom API URL: `/webinars/{webinarId}/status`

Update Zoom API URL: `/webinars/{webinarId}/status`

Delete Zoom API URL: `/webinars/{webinarId}/status`

Field Selection Method: NotRequired

Select Zoom API Operation: `put /webinars/{webinarId}/status`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function putWebinarsByWebinarIdStatus. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
body	string	☑		
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `putWebinarsByWebinarIdStatus` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
RESULT	string		☐	Outcome of operation as single plain text column.

19.1.14 UsersByUserIdWebinars

List webinarsList webinars for a user

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/users/{userId}/webinars`

Insert Zoom API URL: `/users/{userId}/webinars`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Update Zoom API URL: `/users/{userId}/webinars`

Delete Zoom API URL: `/users/{userId}/webinars`

Field Selection Method: NotRequired

Base Path: `webinars[*]`

Select Zoom API Operation: `get /users/{userId}/webinars`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function UsersByUserIdWebinars. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
userId	string	☑		The user ID or email address

Table Function Columns

The columns of the table function `UsersByUserIdWebinars` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
created_at	datetime		☐	Create time
duration	int64		☐	Meeting duration
host_id	string		☐	ID of the user set as host of webinar
id	string		☐	Webinar ID, also know as webinar number
join_url	string		☐	Join url
timezone	string		☐	Timezone to format start_time
topic	string		☐	Meeting topic
type	int64		☐	Meeting Type
uuid	string		☐	Webinar unique ID

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

19.1.15 WebinarsByWebinarId

Retrieve a webinarRetrieve a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}`

Insert Zoom API URL: `/webinars/{webinarId}`

Update Zoom API URL: `/webinars/{webinarId}`

Delete Zoom API URL: `/webinars/{webinarId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webinars/{webinarId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WebinarsByWebinarId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `WebinarsByWebinarId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
agenda	string		☐	Webinar agenda
created_at	datetime		☐	Create time
duration	int64		☐	Webinar duration

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Name	Data Type	Label	Required	Documentation
host_id	string		☐	ID of the user set as host of w e-binar
id	string		☐	Webinar ID, also know as w e-binar number
join_url	string		☐	Join url
settings_allow _multiple_devices	boolean		☐	Allow attendees to join from mul-tiple devices
settings_alternative_hosts	string		☐	Alternative hosts emails or IDs. Multiple values separated by comma.
settings_approval_type	int64		☐	
settings_audio	string		☐	Determine how participants can join the audio portion of the meet-ing
settings_auto_recording	string		☐	
settings_close_registration	boolean		☐	Close registration after event date
settings_enforce_login_domains	string		☐	Only signed-in users w ith spe-cified domains can join meetings
settings_enforce_login	boolean		☐	Only signed-in users can join this meeting
settings_hd_video	boolean		☐	Default to HD Video
settings_host_video	boolean		☐	Start video w hen host joins w e-binar
settings_on_demand	boolean		☐	Make the w ebinar on-demand
settings_panelists_video	boolean		☐	Start video w hen panelists join w ebinar
settings_practice_session	boolean		☐	Enable Practice Session
settings_registration_type	int64		☐	Registration type. Used for re-curring w ebinar w ith fixed time only.
settings_show _share_button	boolean		☐	Show social share buttons on registration page
start_time	datetime		☐	Webinar start time
start_url	string		☐	Start url
timezone	string		☐	Timezone to format start_time
topic	string		☐	Webinar topic
type	int64		☐	Webinar Type
uuid	string		☐	Webinar unique ID

19.1.16 WebinarsByWebinarIdPanelists

List a webinar's panelistsList panelists for a webinar

Catalog: Zoom

Schema: Webinars

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/panelists`

Insert Zoom API URL: `/webinars/{webinarId}/panelists`

Update Zoom API URL: `/webinars/{webinarId}/panelists`

Delete Zoom API URL: `/webinars/{webinarId}/panelists`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webinars/{webinarId}/panelists`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WebinarsByWebinarIdPanelists. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `WebinarsByWebinarIdPanelists` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
total_records	int64		☐	Total records

19.1.17 WebinarsByWebinarIdPolls

List a webinar's polls List polls of a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Select Zoom API URL: `/webinars/{webinarId}/polls`

Insert Zoom API URL: `/webinars/{webinarId}/polls`

Update Zoom API URL: `/webinars/{webinarId}/polls`

Delete Zoom API URL: `/webinars/{webinarId}/polls`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webinars/{webinarId}/polls`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WebinarsByWebinarIdPolls. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `WebinarsByWebinarIdPolls` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
total_records	int64		☐	The number of all records available across pages

19.1.18 WebinarsByWebinarIdPollsPollId

Retrieve a webinar's pollRetrieve a webinar's poll

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

Select Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Insert Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Update Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Delete Zoom API URL: `/webinars/{webinarId}/polls/{pollId}`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webinars/{webinarId}/polls/{pollId}`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WebinarsByWebinarIdPollsPollId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
pollId	string	☑		The poll ID
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `WebinarsByWebinarIdPollsPollId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
id	string		☐	Webinar Poll ID
status	string		☐	Status of the Webinar Poll
title	string		☐	Poll Title

19.1.19 WebinarsByWebinarIdRegistrants

List a webinar's registrantsList registrants for a webinar

Catalog: Zoom

Schema: Webinars

This is a read-only table function. The Zoom API may not support changing the data or the Invantive SQL driver for Zoom does not cover it. In the latter case, please use the table `NativePlatformScalarRequests` to upload data to the Zoom API.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Select Zoom API URL: `/webinars/{webinarId}/registrants`

Insert Zoom API URL: `/webinars/{webinarId}/registrants`

Update Zoom API URL: `/webinars/{webinarId}/registrants`

Delete Zoom API URL: `/webinars/{webinarId}/registrants`

Field Selection Method: NotRequired

Select Zoom API Operation: `get /webinars/{webinarId}/registrants`

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function WebinarsByWebinarIdRegistrants. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be treated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example with `select * from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select * from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
occurrence_id	string	☐		The meeting occurrence ID
page_number	int64	☐	1	Current page number of returned records
page_size	int64	☐	30	The number of records returned within a single API call
status	string	☐	approved	The registrant status (Values: pending, approved, denied)
webinarId	int64	☑		The webinar ID

Table Function Columns

The columns of the table function `WebinarsByWebinarIdRegistrants` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
page_count	int64		☐	The number of items returned on this page
page_number	int64		☐	The page number of current results
page_size	int64		☐	The number of records returned within a single API call
total_records	int64		☐	The number of all records available across pages

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Index

- A -

account_id 46, 51, 207, 209, 213
account_name 15
account_type 15, 86, 88, 90
accountId 16, 17, 18, 19, 24, 25, 26, 28, 30, 32, 33, 34, 35, 37, 39, 40, 41, 42
Accounts 15
AccountsByAccountId 16
AccountsByAccountId_DomainsManagedDomains 17
AccountsByAccountId_Plan_large_meetingPlans 28
AccountsByAccountId_Plan_webinarPlans 30
AccountsByAccountIdBilling 32
AccountsByAccountIdManagedDomains 18
AccountsByAccountIdPlans 33
AccountsByAccountIdSettings 19
action 44, 45, 107, 192
action_time 107
add_at 201
added_at 103, 117
add-odata-mandatory-filters 2
address 32
agenda 120, 135, 236, 246
analysis-enforce-row-uniqueness 2
answer 157, 171, 172
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
approval_type 47
apt 32
assistantId 195
attentiveness_score 158, 173
auth_password 227, 229
auth_user 227, 229

- B -

BLOB Preferred 141
BLOB_PREFERRED 141
body 25, 26, 27, 34, 35, 37, 39, 40, 41, 42, 48, 49, 50, 93, 94, 101, 102, 103, 115, 116, 117, 130, 131, 132, 133, 134, 135, 137, 138, 139, 178, 179, 182, 183, 184, 186, 199, 200, 201, 225, 226, 227, 235, 236, 238, 239, 240, 241, 242, 243
BOL_RESPONSE_CACHE_MAX_AGE_SEC 141
bulk-delete-page-size-rows 2
bulk-insert-page-size-bytes 2
bulk-insert-page-size-rows 2

- C -

calender_name 86, 88, 90
call_in_number 148
calls_receive 58
calls_send 58
camera 86, 88, 90
city 32
cms_user_id 207, 209, 213
conference_code 184, 186, 188, 189, 190, 191
conference_id 143, 144, 145
connection_type 67, 78
content 69, 80
Content Type 141
CONTENT_TYPE 141
country 32, 143, 144
country_label 184, 188, 190
country_name 148
created_at 15, 16, 27, 120, 135, 140, 206, 207, 209, 210, 212, 213, 227, 229, 236, 244, 246

- D -

data_center 67, 78
date 146, 151
DATE_ENDED 141
DATE_STARTED 141
date_time 55, 62, 107
deleteAccountsByAccountId 24
deleteGroupsByGroupId 95
deleteGroupsByGroupIdMembersMemberId 96
deleteH323DevicesByDeviceId 92
deleteImGroupsByGroupId 110
deleteImGroupsByGroupIdMembersMemberId 111
deleteMeetingsByMeetingId 118
deleteMeetingsByMeetingIdPollsPollId 119
deleteMeetingsByMeetingIdRecordings 44
deleteMeetingsByMeetingIdRecordingsRecordingId 45
deleteUsersByUserId 192
deleteUsersByUserIdAssistants 194

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

deleteUsersByUserIdAssistantsAssistantId 195
deleteUsersByUserIdSchedulers 196
deleteUsersByUserIdSchedulersSchedulerId 197
deleteUsersByUserIdToken 198
deleteUsersByUserIdTspTspId 181
deleteV2TrackingFieldsByFieldId 177
deleteWebhooksByWebhookId 224
deleteWebinarsByWebinarId 230
deleteWebinarsByWebinarIdPanelists 231
deleteWebinarsByWebinarIdPanelistsPanelistId 232
deleteWebinarsByWebinarIdPollsPollId 233
dept 148, 149, 206, 207, 209, 210, 212, 213
device 67, 70, 72, 78, 81, 83
device_ip 86, 88, 90
deviceId 92, 93
download-error-400-bad-request-max-tries 2
download-error-400-bad-request-sleep-initial-ms 2
download-error-400-bad-request-sleep-max-ms 2
download-error-400-bad-request-sleep-multiplicator 2
download-error-408-request-timeout-max-tries 2
download-error-408-request-timeout-sleep-initial-ms 2
download-error-408-request-timeout-sleep-max-ms 2
download-error-408-request-timeout-sleep-multiplicator 2
download-error-422-bad-request-max-tries 2
download-error-422-bad-request-sleep-initial-ms 2
download-error-422-bad-request-sleep-max-ms 2
download-error-422-bad-request-sleep-multiplicator 2
download-error-429-too-many-requests-max-tries 2
download-error-429-too-many-requests-sleep-initial-ms 2
download-error-429-too-many-requests-sleep-max-ms 2
download-error-429-too-many-requests-sleep-multiplicator 2
download-error-502-server-unavailable-max-tries 2
download-error-502-server-unavailable-sleep-initial-ms 2
download-error-502-server-unavailable-sleep-max-ms 2
download-error-502-server-unavailable-sleep-multiplicator 2
download-error-503-server-unavailable-max-tries 2
download-error-503-server-unavailable-sleep-initial-ms 2
download-error-503-server-unavailable-sleep-max-ms 2
download-error-503-server-unavailable-sleep-multiplicator 2
download-error-504-gateway-timeout-max-tries 2
download-error-504-gateway-timeout-sleep-initial-ms 2
download-error-504-gateway-timeout-sleep-max-ms 2
download-error-504-gateway-timeout-sleep-multiplicator 2
download-error-590-network-connect-timeout-max-tries 2
download-error-590-network-connect-timeout-sleep-initial-ms 2
download-error-590-network-connect-timeout-sleep-max-ms 2
download-error-590-network-connect-timeout-sleep-multiplicator 2
download-error-599-network-connect-timeout-max-tries 2
download-error-599-network-connect-timeout-sleep-initial-ms 2
download-error-599-network-connect-timeout-sleep-max-ms 2
download-error-599-network-connect-timeout-sleep-multiplicator 2
download-error-argument-exception-max-tries 2
download-error-argument-exception-sleep-initial-ms 2
download-error-argument-exception-sleep-max-ms 2
download-error-argument-exception-sleep-multiplicator 2
download-error-internet-down-max-tries 2
download-error-internet-down-sleep-initial-ms 2
download-error-internet-down-sleep-max-ms 2
download-error-internet-down-sleep-multiplicator 2
download-error-io-exception-max-tries 2
download-error-io-exception-sleep-initial-ms 2
download-error-io-exception-sleep-max-ms 2
download-error-io-exception-sleep-multiplicator 2
download-error-json-exception-max-tries 2
download-error-json-exception-sleep-initial-ms 2
download-error-json-exception-sleep-max-ms 2
download-error-json-exception-sleep-multiplicator 2
download-error-other-exception-max-tries 2
download-error-other-exception-sleep-initial-ms 2
download-error-other-exception-sleep-max-ms 2
download-error-other-exception-sleep-multiplicator 2
download-error-socket-exception-max-tries 2
download-error-socket-exception-sleep-initial-ms 2
download-error-socket-exception-sleep-max-ms 2
download-error-socket-exception-sleep-multiplicator 2
download-error-web-exception-max-tries 2
download-error-web-exception-sleep-initial-ms 2
download-error-web-exception-sleep-max-ms 2
download-error-web-exception-sleep-multiplicator 2

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

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
Driver 1
DRY_RUN 141
DUMMY 126, 234
duration 46, 51, 56, 60, 65, 77, 88, 120, 127, 135, 140, 148, 153, 154, 158, 163, 166, 167, 173, 236, 241, 246
Duration (ms) 141
DURATION_MS 141

- E -

email 32, 56, 58, 60, 65, 77, 86, 88, 90, 99, 113, 149, 156, 157, 169, 170, 171, 172, 201, 206, 207, 209, 210, 212, 213, 215, 217, 221
email_1 88
email_notification_alternative_host_reminder 19, 218
email_notification_cancel_meeting_reminder 19, 218
email_notification_cloud_recording_avaliable_reminder 19
email_notification_jbh_reminder 19, 218
email_notification_low_host_count_reminder 19
emoji_receive 58
emoji_send 58
enable 187
encryption 93, 94
End Date 141
end_time 56, 60, 65, 69, 77, 80, 88, 127, 148, 153, 154, 163, 166, 167
Error Message Code 141
Error Message Text 141
ERROR_MESSAGE_CODE 141
ERROR_MESSAGE_TEXT 141
existed 222
existed_email 221
expire_in 223

- F -

Fail on Error 141
FAIL_ON_ERROR 141
feature_cn_meeting 218
feature_in_meeting 218
feature_large_meeting 218
feature_large_meeting_capacity 218
feature_meeting_capacity 19, 218
feature_webinar 218
feature_webinar_capacity 218
field 153, 166, 179, 180
fieldId 177, 178, 180
files_receive 58
files_send 58
first_name 32, 99, 113, 201, 206, 207, 209, 210, 212, 213
force-case-sensitive-identifiers 2
forced-casing-identifiers 2
free_usage 151
from 51, 53, 55, 56, 58, 60, 62, 63, 64, 75, 88, 90, 104, 105, 107, 108, 148, 149, 151, 160, 161, 163, 165

- G -

group_receive 58
group_send 58
groupId 95, 96, 98, 99, 101, 103, 110, 111, 112, 113, 115, 117
Groups 97
Groups_Groups 98
GroupsByGroupId 98
GroupsByGroupIdMembers 99

- H -

h323_password 120, 135
H323Devices 93
harddisk_id 67, 70, 72, 78, 81, 83
has_3rd_party_audio 56, 60, 65, 77, 88
has_pstn 56, 60, 65, 77, 88
has_recording 56, 60, 65, 77, 88
has_screen_share 56, 60, 65, 77, 88
has_sip 56, 60, 65, 77, 88
has_video 56, 60, 65, 77, 88
has_voip 56, 60, 65, 77, 88
host 56, 60, 65, 77, 88
host_email 148

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

host_id 46, 51, 120, 127, 135, 140, 236, 244, 246
host_key 207, 209, 213
host_name 148
hosts 28, 30, 35, 37
hour 55
HTTP Disk Cache Maximum Age (sec) 141
HTTP Memory Cache Maximum Age (sec) 141
HTTP Method 141
HTTP Status Code 141
HTTP_DISK_CACHE_MAX_AGE_SEC 141
HTTP_DISK_CACHE_SAVE 141
HTTP_DISK_CACHE_USE 141
HTTP_MEMORY_CACHE_MAX_AGE_SEC 141
HTTP_MEMORY_CACHE_SAVE 141
HTTP_MEMORY_CACHE_USE 141
HTTP_METHOD 141
HTTP_STATUS_CODE 141
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

- I -

id_1 88
ids 103, 117, 201
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
images_receive 58
images_send 58
ImChat_SessionsSessions 104
ImChatSessions 105
ImChatSessions_MessagesBySessionId 107
ImChatSessionsBySessionId 108
ImGroups 112
ImGroupsByGroupId 112
ImGroupsByGroupIdMembers 113
in_meeting_alert_guest_join 19
in_meeting_allow_live_streaming 19, 218
in_meeting_allow_show_zoom_windows 19
in_meeting_annotation 19, 218
in_meeting_anonymous_question_answer 19
in_meeting_attendee_on_hold 19, 218
in_meeting_attention_tracking 19, 218
in_meeting_auto_answer 19
in_meeting_auto_saving_chat 19, 218
in_meeting_breakout_room 19, 218
in_meeting_chat 19, 218
in_meeting_closed_caption 19, 218
in_meeting_co_host 19, 218
in_meeting_custom_live_streaming 19, 218
in_meeting_custom_service_instructions 19, 218
in_meeting_dscp_audio 19
in_meeting_dscp_marking 19
in_meeting_dscp_video 19
in_meeting_e2e_encryption 19, 218
in_meeting_entry_exit_chime 218
in_meeting_far_end_camera_control 19, 218
in_meeting_feedback 19, 218
in_meeting_file_transfer 19, 218
in_meeting_group_hd 19, 218
in_meeting_non_verbal_feedback 218
in_meeting_original_audio 19
in_meeting_p2p_connetion 19
in_meeting_p2p_ports 19
in_meeting_polling 19, 218
in_meeting_ports_range 19
in_meeting_post_meeting_feedback 19
in_meeting_private_chat 19, 218
in_meeting_record_play_voice 218
in_meeting_remote_control 19, 218
in_meeting_remote_support 218
in_meeting_screen_sharing 19
in_meeting_sending_default_email_invites 19
in_meeting_share_dual_camera 218
in_meeting_show_meeting_control_toolbar 19
in_meeting_stereo_audio 19
in_meeting_use_html_format_email 19
in_meeting_virtual_background 19, 218
in_meeting_waiting_room 218
in_meeting_watermark 19
in_meeting_webinar_question_answer 19
in_meeting_whiteboard 19
in_meeting_workplace_by_facebook 19, 218
integration_box 19
integration_dropbox 19

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

integration_google_calendar 19
integration_google_drive 19
integration_kubi 19
integration_microsoft_one_drive 19
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-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
invitation 122
ip 93, 94
ip_address 67, 70, 72, 78, 81, 83

- J -

join_time 67, 70, 72, 78, 81, 83, 158, 173
join_url 120, 134, 135, 140, 236, 238, 240, 244, 246
join-set-points-per-request 2

- L -

language 207, 209, 213
last_client_version 206, 207, 209, 210, 212, 213
last_login_time 206, 207, 209, 210, 212, 213
last_message_sent_time 104
last_name 32, 99, 113, 201, 206, 207, 209, 210, 212, 213
last_start_time 86, 88, 90
leader_pin 184, 186, 188, 189, 190, 191
leave_time 67, 70, 72, 78, 81, 83, 158, 173
limit-partition-calls-left 2
listen_only_password 143, 144, 145
live_meeting_duration 88, 90
live_meeting_email 88, 90
live_meeting_end_time 88, 90
live_meeting_has_3rd_party_audio 88, 90
live_meeting_has_pstn 88, 90
live_meeting_has_recording 88, 90
live_meeting_has_screen_share 88, 90
live_meeting_has_sip 88, 90
live_meeting_has_video 88, 90
live_meeting_has_voip 88, 90
live_meeting_host 88, 90
live_meeting_id 88, 90
live_meeting_participants 88, 90
live_meeting_start_time 88, 90
live_meeting_topic 88, 90
live_meeting_user_type 88, 90
live_meeting_uuid 88, 90
location 67, 70, 72, 78, 81, 83
login_type 207, 209, 213, 218
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

- M -

mac_addr 67, 70, 72, 78, 81, 83
max_usage 55
maximum-discovered-column-count 2
maximum-length-identifiers 2
max-odata-filters 2
max-url-length-accepted 2
max-url-length-desired 2
mc 51, 53
meeting_id 148
meeting_minutes 146, 149
meeting_type 148
meetingId 44, 45, 46, 47, 48, 49, 50, 65, 67, 69, 70, 72, 74, 118, 119, 120, 122, 123, 124, 125, 126, 130, 131, 132, 133, 134, 137, 138, 139, 153, 154, 156, 157, 158, 159
meetings 146, 149
MeetingsByMeetingId 120
MeetingsByMeetingIdInvitation 122
MeetingsByMeetingIdPolls 123
MeetingsByMeetingIdPollsPollId 124
MeetingsByMeetingIdRecordings 46
MeetingsByMeetingIdRecordingsSettings 47
MeetingsByMeetingIdRegistrants 125
meetingUUID 127, 128
memberId 96, 111
message 107
metadata-cache-max-age-sec 2
Metrics_Crc_ports_usageCrc_ports_hour_usageCrc 55
Metrics_MeetingsMeetings 56
Metrics_UsersIm 58
Metrics_WebinarsWebinars 60
MetricsCrc 62

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

MetricsIm 63
MetricsMeetings 64
MetricsMeetingsByMeetingId 65
MetricsMeetingsByMeetingIdParticipants 67
MetricsMeetingsByMeetingIdParticipants_ParticipantsDetailsSharing 69
MetricsMeetingsByMeetingIdParticipantsParticipantIdQos 70
MetricsMeetingsByMeetingIdParticipantsQos 72
MetricsMeetingsByMeetingIdParticipantsSharing 74
MetricsWebinars 75
MetricsWebinarsByWebinarId 77
MetricsWebinarsByWebinarIdParticipants 78
MetricsWebinarsByWebinarIdParticipants_ParticipantsDetailsSharing 80
MetricsWebinarsByWebinarIdParticipantsParticipantIdQos 81
MetricsWebinarsByWebinarIdParticipantsQos 83
MetricsWebinarsByWebinarIdParticipantsSharing 85
MetricsZoomrooms 86
MetricsZoomrooms_Past_meetingsMeetingsByZoomroomId 88
MetricsZoomroomsByZoomroomId 90
microphone 67, 78, 86, 88, 90
month 146, 152

- N -

name 93, 94, 98, 102, 104, 112, 116, 128, 156, 157, 158, 169, 170, 171, 172, 173
Native Platform Scalar Requests 141
NATIVEPLATFORMSCALARREQUESTS 141
network_type 67, 78
new_users 146
next_page_token 51, 53, 56, 58, 60, 63, 64, 67, 69, 72, 74, 75, 78, 80, 83, 85, 104, 105, 107, 108, 128, 158, 163, 165, 173
npt 141
number 143, 144, 184, 187, 188, 190

- O -

oauth-unauthorized-max-tries 2
oauth-unauthorized-sleep-initial-ms 2
oauth-unauthorized-sleep-max-ms 2
oauth-unauthorized-sleep-multiplicator 2
occurrence_id 118, 125, 138, 230, 242, 250
occurrence_ids 134, 240
on_demand 47
options_meeting_connectors 16
options_pay_mode 16
options_room_connectors 16
options_share_mc 16
options_share_rc 16
ORIG_SYSTEM_GROUP 141
ORIG_SYSTEM_REFERENCE 141
Original System Group 141
Original System Reference 141
owner_email 15, 16, 27
owner_id 16, 27

- P -

page_count 51, 53, 56, 58, 60, 63, 64, 75, 125, 148, 149, 160, 161, 163, 165, 250
page_number 15, 86, 88, 90, 99, 113, 125, 140, 148, 149, 160, 161, 206, 210, 212, 244, 250
page_size 15, 51, 53, 56, 58, 60, 63, 64, 67, 69, 72, 74, 75, 78, 80, 83, 85, 86, 88, 90, 99, 104, 105, 107, 108, 113, 125, 128, 140, 148, 149, 158, 160, 161, 163, 165, 173, 206, 210, 212, 244, 250
page-size-rows 2
panelistId 232
participant_password 143, 144, 145
participantId 70, 81
participants 56, 60, 65, 77, 88, 146, 149
participants_count 127, 153, 154, 163, 166, 167
partition-slot-based-rate-limit-length-ms 2
partition-slot-based-rate-limit-slots 2
password 47, 120, 135
PastMeetingsByMeetingIdInstances 126
PastMeetingsByMeetingUUID 127
PastMeetingsByMeetingUUIDParticipants 128
PastWebinarsByWebinarIdInstances 234
patchAccountsByAccountIdBilling 34
patchAccountsByAccountIdOptions 25
patchAccountsByAccountIdSettings 26
patchGroupsByGroupId 101
patchH323DevicesByDeviceId 93
patchImGroupsByGroupId 115
patchMeetingsByMeetingId 130
patchMeetingsByMeetingIdLivestream 131
patchMeetingsByMeetingIdLivestreamStatus 132
patchMeetingsByMeetingIdRecordingsSettings 48
patchTsp 182
patchUsersByUserId 199
patchUsersByUserIdSettings 200
patchUsersByUserIdTspTspId 183
patchV2TrackingFieldsByFieldId 178
patchWebhooksByWebhookId 225
patchWebhooksOptions 226
patchWebinarsByWebinarId 235

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

Payload 141
PAYLOAD_TEXT 141
pc_name 67, 70, 72, 78, 81, 83
personal_meeting_url 207, 209, 213
phone_number 32, 148
pic_file 202
pic_url 207, 209, 213
plan_audio_callout_countries 28, 30, 33, 35, 37, 39
plan_audio_ddi_numbers 28, 30, 33, 35, 37, 39
plan_audio_premium_countries 28, 30, 33, 35, 37, 39
plan_audio_tollfree_countries 28, 30, 33, 35, 37, 39
plan_audio_type 28, 30, 33, 35, 37, 39
plan_base_hosts 28, 30, 33, 35, 37, 39
plan_base_type 28, 30, 33, 35, 37, 39
plan_recording 28, 30, 33, 35, 37, 39
plan_room_connector_hosts 28, 30, 33, 35, 37, 39
plan_room_connector_type 28, 30, 33, 35, 37, 39
plan_usage 151
plan_zoom_rooms_hosts 28, 30, 33, 35, 37, 39
plan_zoom_rooms_type 28, 30, 33, 35, 37, 39
pmi 206, 207, 209, 210, 212, 213
pollId 119, 124, 137, 233, 241, 249
postAccounts 27
postAccountsByAccountId_Plan_large_meetingPlans 35
postAccountsByAccountId_Plan_webinarPlans 37
postAccountsByAccountIdPlans 39
postAccountsByAccountIdPlansAddons 40
postGroups 102
postGroupsByGroupIdMembers 103
postH323Devices 94
postImGroups 116
postImGroupsByGroupIdMembers 117
postMeetingsByMeetingIdPolls 133
postMeetingsByMeetingIdRegistrants 134
postUsers 201
postUsersByUserId_Dial_in_numbersTsp 184
postUsersByUserIdAssistants 201
postUsersByUserIdMeetings 135
postUsersByUserIdPicture 202
postUsersByUserIdTsp 186
postUsersByUserIdWebinars 236
postV2TrackingFields 179
postWebhooks 227
postWebinarsByWebinarIdPanelists 238
postWebinarsByWebinarIdPolls 239
postWebinarsByWebinarIdRegistrants 240
pre-request-delay-ms 2
protocol 93, 94
putAccountsByAccountIdPlansAddons 41
putAccountsByAccountIdPlansBase 42
putMeetingsByMeetingIdPollsPollId 137
putMeetingsByMeetingIdRecordingsRecordingIdStatus 49
putMeetingsByMeetingIdRecordingsStatus 50
putMeetingsByMeetingIdRegistrantsStatus 138
putMeetingsByMeetingIdStatus 139
putUsersByUserIdEmail 203
putUsersByUserIdPassword 204
putUsersByUserIdStatus 205
putWebinarsByWebinarIdPollsPollId 241
putWebinarsByWebinarIdRegistrantsStatus 242
putWebinarsByWebinarIdStatus 243

- Q -

question 157, 171, 172

- R -

recording 67, 78
recording_account_user_access_recording 19
recording_auto_delete_cmr 19, 218
recording_auto_delete_cmr_days 19, 218
recording_auto_recording 19, 218
recording_cloud_recording 19, 218
recording_cloud_recording_download 19
recording_cloud_recording_download_host 19
recording_count 46, 51
recording_local_recording 19, 218
recording_record_audio_file 19, 218
recording_record_gallery_view 19, 218
recording_record_speaker_view 19, 218
recording_recording_audio_transcript 19, 218
recording_save_chat_text 19, 218
recording_show_timestamp 19, 218
recordingId 45, 49
registrant_id 134, 240
Report_DatesDaily 146
Report_Telephony_usageTelephone 148
Report_UsersUsers 149
ReportCloudRecording 151
ReportDaily 152
ReportMeetings_Tracking_fieldsByMeetingId 153
ReportMeetingsByMeetingId 154
ReportMeetingsByMeetingId_QuestionsPolls 156
ReportMeetingsByMeetingId_QuestionsQuestion_detailsPolls 157
ReportMeetingsByMeetingIdParticipants 158
ReportMeetingsByMeetingIdPolls 159
ReportTelephone 160

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

ReportUsers 161
ReportUsersByUserId_MeetingsMeetings 163
ReportUsersByUserIdMeetings 165
ReportWebinars_Tracking_fieldsByWebinarId 166
ReportWebinarsByWebinarId 167
ReportWebinarsByWebinarId_QuestionsPolls 169
ReportWebinarsByWebinarId_QuestionsQa 170
ReportWebinarsByWebinarId_QuestionsQuestion_detailsPolls 171
ReportWebinarsByWebinarId_QuestionsQuestion_detailsQa 172
ReportWebinarsByWebinarIdParticipants 173
ReportWebinarsByWebinarIdPolls 175
ReportWebinarsByWebinarIdQa 176
requested-page-size 2
requests-parallel-max 2
required 179, 180
Response Cache Maximum Age (sec) 141
RESULT 24, 25, 26, 27, 34, 35, 37, 39, 40, 41, 42, 44, 45, 48, 49, 50, 92, 93, 94, 95, 96, 101, 102, 103, 110, 111, 115, 116, 117, 118, 119, 130, 131, 132, 133, 134, 135, 137, 138, 139, 177, 178, 179, 181, 182, 183, 184, 186, 192, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 224, 225, 226, 227, 230, 231, 232, 233, 235, 236, 238, 239, 240, 241, 242, 243
Result BLOB 141
Result Text 141
RESULT_BLOB 141
RESULT_DATE_TIME_UTC 141
RESULT_NUMBER 141
RESULT_TEXT 141
room_name 86, 88, 90
Run without Actions 141

- S -

Save HTTP Disk Cache 141
Save HTTP Memory Cache 141
schedule_meeting_audio_type 218
schedule_meeting_force_pmi_jbh_password 218
schedule_meeting_host_video 218
schedule_meeting_join_before_host 218
schedule_meeting_participants_video 218
schedule_meeting_pstn_password_protected 218
schedule_meting_audio_type 19
schedule_meting_enforce_login 19
schedule_meting_enforce_login_domains 19
schedule_meting_enforce_login_with_domains 19
schedule_meting_force_pmi_jbh_password 19
schedule_meting_host_video 19
schedule_meting_join_before_host 19
schedule_meting_not_store_meeting_topic 19
schedule_meting_participant_video 19
schedulerId 197
search_by_account 112, 116
search_by_domain 112, 116
search_by_ma_account 112, 116
seats 15
security_admin_change_name_pic 19
security_hide_billing_info 19
security_import_photos_from_devices 19
send_email_to_host 47
sender 107
session_id 104, 107, 108
sessionId 107, 108
settings_allow_multiple_devices 236, 246
settings_alternative_hosts 120, 135, 236, 246
settings_approval_type 120, 135, 236, 246
settings_audio 120, 135, 236, 246
settings_auto_recording 120, 135, 236, 246
settings_close_registration 120, 135, 236, 246
settings_cn_meeting 120, 135
settings_enforce_login 120, 135, 236, 246
settings_enforce_login_domains 120, 135, 236, 246
settings_hd_video 236, 246
settings_host_video 120, 135, 236, 246
settings_in_meeting 120, 135
settings_join_before_host 120, 135
settings_mute_upon_entry 120, 135
settings_on_demand 236, 246
settings_panelists_video 236, 246
settings_participant_video 120, 135
settings_practice_session 236, 246
settings_registration_type 120, 135, 236, 246
settings_show_share_button 236, 246
settings_use_pmi 120, 135
settings_waiting_room 120, 135
settings_watermark 120, 135
share_application 67, 78
share_desktop 67, 78
share_recording 47
share_whiteboard 67, 78
show_social_share_buttons 47
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

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

simulate-http-500-errors 2
simulate-http-500-errors-percentage 2
simulate-http-502-errors 2
simulate-http-502-errors-percentage 2
simulate-http-503-errors 2
simulate-http-503-errors-percentage 2
simulate-http-protocol-errors 2
simulate-http-protocol-errors-percentage 2
simulate-http-timeout-errors 2
simulate-http-timeout-errors-percentage 2
slot-based-rate-limit-length-ms 2
slot-based-rate-limit-slots 2
speaker 67, 78, 86, 88, 90
standardize-identifiers 2
standardize-identifiers-casing 2
Start Date 141
start_time 46, 51, 56, 60, 65, 69, 77, 80, 88, 120, 127, 134, 135, 140, 148, 153, 154, 156, 157, 159, 165, 166, 167, 169, 170, 171, 172, 175, 176, 236, 240, 246
start_url 120, 135, 236, 246
state 32
status 17, 86, 88, 90, 124, 125, 133, 206, 210, 212, 239, 249, 250
subscription_end_time 15
subscription_start_time 15
Succesful 141
SUCCESSFUL 141
swagger-specification-file 2
swagger-specification-http-disk-cache-max-age-sec 2
swagger-specification-url 2

- T -

telephony_audio_conference_info 19, 218
telephony_show_international_numbers_link 218
telephony_third_party_audio 19, 218
TEXT 216
Timeout (sec) 141
TIMEOUT_SEC 141
timezone 120, 135, 140, 206, 207, 209, 210, 212, 213, 236, 244, 246
title 124, 133, 239, 249
to 51, 53, 55, 56, 58, 60, 62, 63, 64, 75, 88, 90, 104, 105, 107, 108, 148, 149, 151, 160, 161, 163, 165
token 220
topic 46, 51, 56, 60, 65, 77, 88, 120, 127, 134, 135, 140, 153, 154, 163, 166, 167, 236, 240, 244, 246
total 148
total_meeting_minutes 149, 161
total_meetings 149, 161
total_members 98, 102, 112, 116
total_minutes 127, 153, 154, 163, 166, 167
total_participants 149, 161
total_receive 58
total_records 17, 18, 51, 53, 56, 58, 60, 63, 64, 75, 97, 98, 123, 125, 148, 149, 160, 161, 163, 165, 180, 229, 247, 248, 250
total_send 58
total_size 46, 51
total_usage 55
Transaction ID 141
TRANSACTION_ID 141
transfer_email 192
transfer_meeting 192
transfer_recording 192
transfer_webinar 192
trash 51, 53
Tsp 187
Tsp_Dial_in_numbers 187
tsp_provider 187
tspId 181, 183, 190, 191
type 28, 30, 35, 37, 56, 60, 64, 65, 67, 69, 70, 72, 74, 75, 77, 78, 80, 81, 83, 85, 99, 104, 112, 113, 120, 127, 135, 140, 148, 149, 153, 154, 160, 161, 163, 166, 167, 184, 187, 188, 190, 201, 206, 207, 209, 210, 212, 213, 220, 236, 244, 246

- U -

url 141, 227, 229
usage 151
Use HTTP Disk Cache 141
Use HTTP Memory Cache 141
use_pmi 207, 209, 213
use-batch-insert 2
use-http-disk-cache-read 2
use-http-disk-cache-write 2
use-http-memory-cache-read 2
use-http-memory-cache-write 2
user_email 127, 153, 154, 158, 163, 166, 167, 173
user_id 58, 67, 69, 70, 72, 74, 78, 80, 81, 83, 85, 158, 173
user_name 58, 67, 69, 70, 72, 74, 78, 80, 81, 83, 85, 127, 149, 153, 154, 163, 166, 167
user_qos_audio_input_avg_loss 70, 72, 81, 83
user_qos_audio_input_bitrate 70, 72, 81, 83
user_qos_audio_input_jitter 70, 72, 81, 83
user_qos_audio_input_latency 70, 72, 81, 83
user_qos_audio_input_max_loss 70, 72, 81, 83
user_qos_audio_output_avg_loss 70, 72, 81, 83
user_qos_audio_output_bitrate 70, 72, 81, 83
user_qos_audio_output_jitter 70, 72, 81, 83
user_qos_audio_output_latency 70, 72, 81, 83

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

user_qos_audio_output_max_loss 70, 72, 81, 83
user_qos_cpu_usage_system_max_cpu_usage 70, 72, 81, 83
user_qos_cpu_usage_zoom_avg_cpu_usage 70, 72, 81, 83
user_qos_cpu_usage_zoom_max_cpu_usage 70, 72, 81, 83
user_qos_cpu_usage_zoom_min_cpu_usage 70, 72, 81, 83
user_qos_date_time 70, 72, 81, 83
user_type 56, 60, 65, 77, 88
userId 51, 53, 135, 140, 143, 144, 145, 163, 165, 181, 183, 184, 186, 188, 189, 190, 191, 192, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 207, 209, 213, 215, 216, 217, 218, 220, 236, 244
Users 206
Users_Group_idsByUserId 207
Users_lm_group_idsByUserId 209
Users_UsersGroup_ids 210
Users_UsersIm_group_ids 212
UsersByUserId 213
UsersByUserId_MeetingsRecordings 51
UsersByUserId_Tsp_accountsDedicated_dial_in_numberPac 143
UsersByUserId_Tsp_accountsDial_in_numbersTsp 188
UsersByUserId_Tsp_accountsGlobal_dial_in_numbersPac 144
UsersByUserIdAssistants 215
UsersByUserIdMeetings 140
UsersByUserIdPac 145
UsersByUserIdPermissions 216
UsersByUserIdRecordings 53
UsersByUserIdSchedulers 217
UsersByUserIdSettings 218
UsersByUserIdToken 220
UsersByUserIdTsp 189
UsersByUserIdTsp_Dial_in_numbersTspId 190
UsersByUserIdTspTspId 191
UsersByUserIdWebinars 244
UsersEmail 221
UsersVanityName 222
UsersZpk 223
uuid 46, 51, 56, 60, 65, 77, 88, 120, 127, 135, 140, 153, 154, 156, 157, 159, 163, 166, 167, 169, 170, 171, 172, 175, 176, 236, 244, 246

- V -

V2TrackingFields 180
V2TrackingFieldsByFieldId 180
value 153, 166, 207, 209, 210, 212
vanity_name 222
vanity_url 16, 207, 209, 213
verified 207, 209, 213
version 67, 70, 72, 78, 81, 83
videos_receive 58
videos_send 58
viewer_download 47
visible 179, 180
voice_receive 58
voice_send 58

- W -

webhook_id 227, 229
webhookId 224, 225, 229
Webhooks 229
WebhooksByWebhookId 229
webinarId 77, 78, 80, 81, 83, 85, 166, 167, 169, 170, 171, 172, 173, 175, 176, 230, 231, 232, 233, 234, 235, 238, 239, 240, 241, 242, 243, 246, 247, 248, 249, 250
WebinarsByWebinarId 246
WebinarsByWebinarIdPanelists 247
WebinarsByWebinarIdPolls 248
WebinarsByWebinarIdPollsPollId 249
WebinarsByWebinarIdRegistrants 250

- Y -

year 146, 152

- Z -

zip 32
Zoom 1, 15, 16, 17, 18, 19, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 37, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 53, 55, 56, 58, 60, 62, 63, 64, 65, 67, 69, 70, 72, 74, 75, 77, 78, 80, 81, 83, 85, 86, 88, 90, 92, 93, 94, 95, 96, 97, 98, 99, 101, 102, 103, 104, 105, 107, 108, 110, 111, 112, 113, 115, 116, 117, 118, 119, 120, 122, 123, 124, 125, 126, 127, 128, 130, 131, 132, 133, 134, 135, 137, 138, 139, 140, 141, 143, 144, 145, 146, 148, 149, 151, 152, 153, 154, 156, 157, 158, 159, 160, 161, 163, 165, 166, 167, 169, 170, 171, 172, 173, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 186, 187, 188, 189, 190, 191, 192, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 209, 210, 212, 213, 215, 216, 217, 218, 220, 221, 222, 223, 224, 225, 226, 227, 229, 230, 231, 232, 233, 234, 235, 236, 238, 239, 240, 241, 242, 243, 244, 246, 247, 248, 249, 250

(C) Copyright 2004-2025 Invantive Software B.V., the Netherlands. All rights reserved.

zoom_rooms_auto_start_stop_scheduled_meetings
19
zoom_rooms_cmr_for_instant_meeting 19
zoom_rooms_force_private_meeting 19
zoom_rooms_hide_host_information 19
zoom_rooms_list_meetings_with_calendar 19
zoom_rooms_start_airplay_manually 19
zoom_rooms_ultrasonic 19
zoom_rooms_upcoming_meeting_alert 19
zoom_rooms_weekly_system_restart 19
zoom_rooms_zr_post_meeting_feedback 19
zoomroomId 88, 90
zpk 223

(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
28/05/2026, 20:20
ISBN 978-94-6302-000-8: