Webex WFO API Reference Guide

For Deployments with New WFM

First Published: July 20, 2021
Last Updated: January 17, 2025

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0882

Contents

Getting Started

Webex WFO includes RESTful APIs that allow you to interact with the application in different ways. This section explains how to use Webex WFO APIs. Familiarize yourself with the authorization process before moving on to any other APIs.

Authorization

The authorization process is comparable to logging in to Webex WFO. Therefore, it must be completed before you can make any API requests. To authenticate yourself as an authorized user of Webex WFO and its APIs, you must send your formatted credentials as a POST request to the Authorize API endpoint (/api/rest/authorize). See Authorize API for more information.

NOTE Your session times out after two hours of inactivity.

For information about how to create a credentials for this request, see Create an API user for Classic Webex WFO.

Authorization Request

The following table and JSON example show what your POST request should include and how it should be formatted.

Name	Req?	Type	Description
locale	N	String	User's country localization code.
userId	Y	String	ID of the tenant's user.
password	Y	String	User's password.
language	Y	String	User's language. Default = en
tenantId	N	String	The ID of the tenant the user belongs to. Include the tenant ID if the user has the same login credentials (user ID and password) for more than one tenant.

JSON Request

```
{
          "locale": string,
          "userId": string,
          "password": string,
          "language": string,
          "tenantId": integer
}
```

Using the Authorization Response

When your credentials are accepted, you will receive a response that includes a session ID. The session ID must be included as a header in all subsequent API calls. This is how Webex WFO verifies that you are logged in and authorized to use the APIs.

> **NOTE** Some API development tools like Postman will save the session ID and pass it back on subsequent requests without the need for intervention. Other API tools require you to set the header manually.

This session ID must be formatted in a key-value pair with "cookie" as the key and the session ID added after "hazelcast.sessionId=" as seen in the following key-value pair.

```
"cookie": "hazelcast.sessionId=<The session ID as it appears in the Authorize
   API response>"
```

Destroying the Session

On application shutdown, you should destroy the HTTPS session (log out).

Log out

- Issue an HTTPS DELETE request to:

  ```
  https://<domainname>.com/api/rest/authorize
  ```

Working with Text Files

When you import text data using a Webex WFO API, the text should be UTF-8 encoded. This allows the inclusion of special characters that are not available with ASCII encoding.

Create an API user for Classic Webex WFO

If you plan to use Webex WFO APIs for third-party integrations, Cisco recommends that you create an API user account. You configure an API user the same as a system administrator, but the purpose of this user is to enable third-party applications to authenticate with Webex WFO.

NOTE To create this user, you must have the default system administrator role or a role with equivalent permissions. This role consumes one System Administrator license.

For more information on authorizing and authenticating a user with Webex WFO, see Authorize API .

If you run into any issues, contact Technical Support for assistance.

Create an API user account

1. Start Webex WFO and log in as a system administrator.

   NOTE If you are logging into the product for the first time, create an additional Administrative account.

2. Navigate to Application Management > Global > User Configuration > Users.
3. Select the **Create a new user** option and complete the following fields in the User Information section.

   - **First Name** and **Last Name**—Enter a first and last name that identifies the purpose of the user. For example, enter **API** as the First Name and **User** as the Last Name.
   - **User Name**—Enter a valid email address. This is the user's Webex WFO user name that will be used to log in.

4. Enter a password that meets your tenant's password complexity requirements in the **Password** field.
5. In the **Roles** field, assign the user to the default system Administrator role.
6. Set the **Team** field to Default Team. You can set this value to an alternate team is you prefer.
7. Select the Associated Groups and Teams that this account should have scoping over. You can select individual teams, a Group name, or a Tenant name as a group. Selecting a Tenant name highlights all teams within that group.

11. In the WFM Views field, assign **EnterpriseView** as the main view. This allows the user to see all users.
12. Click **Save** in the upper, right corner of the user interface.

Related topics

View the *Webex WFO API Reference Guide*.

Access WFM APIs

Webex WFO New WFM uses an interactive website to document APIs that is available for all customer environments. This topic details the procedures on how to access WFM APIs. This includes how to enable required permissions, get an authentication token, authorize a session, and end a session.

> **NOTE** The rate limit for WFM APIs per user token is 500 calls per minute and 10,000 calls per hour. This rate limit does not apply to WFM historical APIs.

The URL to your organization's environment follows the template below where [customerURL] is unique to your organization.

```
https://[customerURL].com/api/docs/index.html
```

Page locations

- WFM > Permissions
- User Settings > Personal access tokens

Procedures

Enable access to New WFM APIs

1. Under the list of **Roles**, select the role you want to enable permissions for.
2. Select the **Functions** tab.
3. Under **Open Calabrio WFM** > **API access**, enable one or more of the following permissions.
    - Read forecast
    - Read organization
    - Read request
    - Read schedule
    - Write forecast
    - Write organization
    - Write request

- Write schedule

4. Under **Open Calabrio WFM** > **Web**, enable the **Personal access tokens** permission.

Generate a token

Generating an access token is a one time action. After creating your token, save it in an easy to access location for future use. If you lose your token, you can regenerate a new one.

1. Navigate to **User Settings** > **Personal access tokens**.
2. Enter a unique name in the **Name of application** text field.
3. Click **Generate token**.
4. Copy the token.

NOTE The token does not work if the user creating the token has a leaving date that has passed or if the user is deleted.

Authorize

1. Navigate to your organization's URL for Webex WFO WFM APIs.
2. Click **Authorize**. The Available authorizations window appears.
3. Paste the token you copied in the previous procedure in the **Value** text field.
4. Click **Authorize**.
5. Click **Close** to close the window. You can now execute queries and commands for the WFM APIs available.

End your authorized session

1. Click **Authorize**.
2. Click **logout**.

Historical Data WFM APIs

Webex WFO supports historical data APIs for all current and new versions of WFM (TWFM and New WFM). These historical data APIs enable users to import external WFM historical data from integrated contact center platforms. With these integrations you can forecast, set agent performance follow ups, and more.

> **IMPORTANT** To ensure proper support when integrating with the historical data APIs, please engage with Webex WFO Professional Services to set up a managed service plan to support your organization with the development and deployment of your Webex WFO integration.

The URL to your organization's environment follows the template below where [customerURL] is unique to your organization.

URLs for historical data APIs follow the format detailed below.

```
https://[customerURL].com/historical-data/api/v1
```

URLs for historical data API documentation follow the format detailed below.

```
https://[customerURL].com/historical-data/docs/index.html
```

Authorize

1. Navigate to your organization's URL for Webex WFO WFM API for historical data.
2. Click **Authorize**. The Available authorizations window appears.
3. Paste the tenant key that you received from your Webex WFO contact person.
4. Click **Authorize**.
5. Click **Close** to close the window. You can now execute the APIs available by using the data source key given to you by your Webex WFO contact person.

End your authorized session

1. Click **Authorize**.
2. Click **logout**.

Authorize API

The Authorize API enables users to be authorized to log in to Webex WFO and to receive information associated with an authorized user.

NOTE Your session times out after two hours of inactivity.

GET Protocol/URI

URI	`/api/rest/authorize`
Method	GET

Request Fields

Name	Req?	Type	Description
locale	N	String	User's country localization code.
userId	Y	String	ID of the tenant's user.
password	Y	String	User's password.
language	Y	String	User's language. Default = en
tenantId	N	String	The ID of the tenant the user belongs to. Include the tenant ID if the user has the same login credentials (user ID and password) for more than one tenant.

JSON File Examples

Request

The following is an example of a formatted JSON request.

```
{
        "locale": string,
```

```
        "userId": string,
        "password": string,
        "language": string,
        "tenantId": integer
}
```

Response

The following is an example of a formatted JSON response.

```
{
        "sessionId":"HZ9A274F889D034FD19E399D5321EDFFB0",
        "id":2,
        "userId":"user@test.com",
        "firstName":"bill",
        "lastName":"smith",
        "teamId":3,
        "teamName":"Default Team",
        "groupId":2,
        "groupName":"Default Group",
        "lang":"en",
        "country":"",
        "sessionTimeout":"60",
        "timezoneString":"America\/Chicago",
        "deactivated":false,
        "licenseCompliance":true,
        "tenantId":1,
        "passwordExpirationDate":null,
        "userIq":false,
        "activatedDate": 0,
        "permissions":[
              "ViewOrg",
              "UpdateOrg",
              "LiveScreenMonitoring",
```

```
                "AdministerPasswordPolicy",
                        ...
        ]
        "groupsInScope": [],
        "teamsInScope": [],
        "serviceQueuesInScope":[]
}
```

POST Protocol and URI

URI	`/api/rest/authorize`
Method	POST

Request Fields

Name	Req?	Type	Description
locale	N	String	User's country localization code.
userId	Y	String	ID of the tenant's user.
password	Y	String	User's password.
language	Y	String	User's language. Default = en
tenantId	N	String	The ID of the tenant the user belongs to. Include the tenant ID if the user has the same login credentials (user ID and password) for more than one tenant.

JSON File Examples

Request

The following is an example of a formatted JSON request.

```
{
        "locale": string,
        "userId": string,
```

```
        "password": string,
        "language": string,
        "tenantId": integer
}
```

Response

The following is an example of a formatted JSON response.

```
{
        "sessionId":"HZ9A274F889D034FD19E399D5321EDFFB0",
        "id":2,
        "userId":"user@test.com",
        "firstName":"bill",
        "lastName":"smith",
        "teamId":3,
        "teamName":"Default Team",
        "groupId":2,
        "groupName":"Default Group",
        "lang":"en",
        "country":"",
        "sessionTimeout":"60",
        "timezoneString":"America\/Chicago",
        "deactivated":false,
        "licenseCompliance":true,
        "tenantId":1,
        "passwordExpirationDate":null,
        "userIq":false,
        "activatedDate": 0,
        "permissions":[
              "ViewOrg",
              "UpdateOrg",
              "LiveScreenMonitoring",
              "AdministerPasswordPolicy",
```

```
                    ...
            ]
            "groupsInScope": [],
            "teamsInScope": [],
            "serviceQueuesInScope":[]
}
```

Import contacts in bulk

Third parties use the Bulk Contact Import APIs to merge and insert metadata and recordings in a multi-part HTTPS request.

There are two Bulk Import APIs:

- Bulk Contact Import API—Used by third parties; allows insertion of both metadata and recordings in a multi-part HTTPS request

  > **NOTE** You can also use the Bulk Contact Import API to import contacts in bulk through the Data Server. See Using the Data Server for bulk contact import.

- Real-time Contact API—Used by Webex WFO Smart Desktop recording client; inserts metadata first and then uploads recordings separately later, based on the response from the metadata insertion. This allows the client to delay uploading recordings and allows contacts to be inserted while the call is still in progress. However, this forces the client to track IDs to upload the recordings for the correct contact.

Protocol and URI

Bulk contact import API

URI	`/api/upload/contacts`
Method	POST
Permissions	Administer Tenant
Content Type	multipart/form-data

Real-time bulk contact import API (metadata)

URI	`/api/rest/wfo/contact/import`

Method	POST
Permissions	Capture Contacts, Record Voice/Record Screen
Content Type	application/JSON

Supported formats

The following formats can be included in a multipart request.

Type	Description
CSV	A comma-separated file used to assign metadata.
JSON	The same metadata format as the Real-time API, but can be used for bulk import.
WAV	An audio recording format.
WEBM	A combined audio and video recording format.
WEBMA	An audio-only (WebM container) format.
WEBMV	A video-only (WebM container) format.
SPX	An audio format.
WMV	A combined audio and video recording format (or video only if paired with audio in the same contact).
OPUS	An audio format.

Request and response fields

The CSV and JSON files include fields defined in the following table. Not all fields are used in both types of files. The file the field applies to is indicated in the description.

Name	Req?	Description
AgentId	Y	Used in CSV and JSON. The Agent ID in one of three formats. Processing figures out

Name	Req?	Description
		which format is used based on parsing the contents. ■ Person ID. A unique identifier from WfoPerson.id. This number is also used in the User Export spreadsheet (Application Management > Global > Users > Import and Export > Export > User ID column). It is not the same as the ID in the Webex WFO user profile. ■ AD Login. A domain\username (requires "\"). ■ Email address. An email address (requires "@"). When using a CSV to upload contacts, the agent ID is required. If you are using JSON to make the request, the agent ID is optional. In the latter case, the agent ID is set to the ID of the authenticated user initiating the upload. Max characters = 254 Default = none
AssocCallId	N	Used in CSV and JSON. An ID that ties contacts together. For example, a transferred call from one agent to another each have the same ID. Max characters = 52 Default = NULL
Audio.Location	N	Used in CSV and JSON. In the Audio sub-object. The key (file) name of the recording in the multipart request. This can be any supported recording format (audio/screen/combined). Only a single audio file per contact is allowed. The key name must have a valid extension that matches the media type of the recording. The extension identifies the file as an audio or screen recording, or both. Max characters = 128 Default = None

Name	Req?	Description
Audio.StartTimeMs	N	Used in CSV and JSON. In the Audio sub-object. The start time in milliseconds GMT since 1970-01-01 (UNIX time) of the audio recording. This is used to determine the audio offset from when the contact starts. Max characters = long Default = ContactStartTimeMs
CalledAddress	N	Used in CSV. The called phone number. Max characters = 64 Default = Empty string
Called	N	Used in JSON. The called phone number. Max characters = 64 Default = NULL
CallId	N	Used in CSV and JSON. An ID that identifies a contact. Max characters = 128 Default = NULL
Calling	N	Used in JSON. The calling phone number. Max characters = 64 Default = NULL
CallingAddress	N	Used in CSV. The calling phone number. Max characters = 64

Name	Req?	Description
		Default = Empty string
ClientTimeZone	N	Used in CSV and JSON. The time zone in UTC format. Windows Time is also supported. The Desktop Recording client sends Windows Time, which is mapped to Olson time. Max characters = 255 Default = Customer's time zone as defined in Webex WFO **EXAMPLE** –06:00
ContactStartTimeMs	N	Used in CSV and JSON. The start time in milliseconds GMT since 1970-01-01 (UNIX time). A value in this field is required, so if the value is missing, the API uses the current upload time. Note that this likely results in a poor user experience, with many contacts that have the same timestamp. **IMPORTANT** If you are importing contacts with Excel, you must format the Start Time column to display milliseconds (consult the Excel user documentation for more information). Otherwise, Excel truncates milliseconds, resulting in a false time and preventing recordings from importing correctly. Max characters = long Default = current upload time **EXAMPLE** 1447100000000 - 11/09/2015 20:13:20 GMT
Direction	N	Used in CSV and JSON. The direction of the call, inbound or outbound. 1 = outbound 0 = inbound

Name	Req?	Description
		Max characters = 1 Default = NULL
Line	N	Used in CSV and JSON. The agent's line/extension. Max characters = 64 Default = NULL
metadata.<custom metadata field name>	N	Used in CSV. The custom metadata fields to populate. The field will be created if it does not exist. Any column beginning with "metadata" will be treated as a custom metadata field. **EXAMPLE** To set "accountNumber", create a column named "metadata.accountNumber". Max characters field name = 39 Max characters of custom metadata value = 2056
CustomMetadata	N	Used in JSON. The custom metadata fields to populate. The field will be created if it does not exist. The object contains data in the form of name/value pairs. **EXAMPLE** "accountNumber":"123456" Max characters field name = 39 Max characters of custom metadata value = 2056
Recording1	Y	Used in CSV. The key (file) name of the recording in the multipart request. This can be any supported recording format (audio/screen/combined). Only a single audio file per contact is allowed. The key name must have a valid extension that matches the media type of the recording. The extension

Name	Req?	Description
		identifies the file as an audio or screen recording, or both. Max characters = 128 Default = None
Recording2	N	Used in CSV. The key (file) name of the recording in the multipart request. This can be any supported recording format (audio/screen/combined). Only a single audio file per contact is allowed. The key name must have a valid extension that matches the media type of the recording. The extension identifies the file as an audio or screen recording, or both. Max characters = 128 Default = None
Recording3	N	Used in CSV. The key (file) name of the recording in the multipart request. This can be any supported recording format (audio/screen/combined). Only a single audio file per contact is allowed. The key name must have a valid extension that matches the media type of the recording. The extension identifies the file as an audio or screen recording, or both. **NOTE** There must be an audio file or the import fails. Max characters = 128 Default = None
Recording2Offset	N	Used in CSV. The offset of Recording2 from Recording1. **EXAMPLE** An audio file (WAV) that starts 5 seconds after the screen file (WEBM) has an offset of 5000 (5000 = 5 seconds). Max characters = Long

Name	Req?	Description
		Default = 0
Screen.Location	N	Used in CSV and JSON. In the Screen sub-object. The key (file) name of the recording in the multipart request. This can be any supported recording format (audio/screen/combined). Only a single audio file per contact is allowed. The key name must have a valid extension that matches the media type of the recording. The extension identifies the file as an audio or screen recording, or both. Max characters = 128 Default = None
Screen.StartTimeMs	N	Used in CSV and JSON. In the Screen sub-object. The start time in milliseconds GMT since 1970-01-01 (UNIX time) of the screen recording. This is used to figure out the screen offset from when the contact starts. Max characters = long Default = ContactStartTimeMs

CSV file examples

CSV can be uploaded as part of a multipart upload request. Some rules regarding the format are as follows.

- The number of columns is variable. For example, if you always want to use the customer's time zone, you do not have to include the TimeZone column in the CSV.
- The columns included in the CSV can be in any order.
- The number of columns in each row must match the number of header columns.
- If a value has a comma, it must be surrounded by quotes.
- If a value is not known for a specific contact, but the header exists, use a empty string for that column.

> **NOTE** A successful response is formatted in JSON, because the response adds some status for each contact and recording.

Full example

This example shows a file that uses every field possible for a CSV file.

```
AgentId,ContactStartTimeMs,TimeZone,AssocCallId,CallId,CalledAddress,Line,CallingAddress,Direction,Recording1,Recording2,Recording2Offset,metadata.accountNumber
abc/bunkowm,1447100000000,America/Chicago,103585664793210000,30611848,1801,1800,1800,1,call1.webmv,call1.wav,5000,1234567890
mark.bunkowske@abc.com,1447110000000,America/Chicago,103585664793220000,30611848,1801,1800,1800,1,call2.wav,,,987654321
```

Short example

This example shows only the fields required for a CSV file.

```
AgentId,ContactStartTimeMs,Recording1
2,1447100000000,call1.wav
2,1447110000000,call2.wav
```

JSON file examples

Contact information can be imported into Webex WFO in JSON format as an alternative to CSV format.

Full Example

The following is an example of a formatted JSON file.

```
{
   "AgentId":"john.smith@acme.com",
   "AssocCallId":"103585664793254280",
   "CallId":"30611848",
   "CalledAddress":"1801",
   "CallingAddress":"1800",
   "ClientTimeZone":"Central Standard Time",
   "ContactStartTimeMs":1447075073000,
   "Direction":1,
   "Audio":[
      {
         "Location":"25.wav",
         "StartTimeMs":1447075080000
      }
   ],
   "Screen":[
         "Location":"25.webm",
         "StartTimeMs":1447075075000
   ],
   "CustomMetadata":{
         "accountNumber":"123456",
```

```
            "department":"sales"
        }
    }
```

Short example

This example shows only the fields required for a JSON file.

```
{
   "AgentId":"acme\smithj",
   "ContactStartTimeMs":1447075073000,
   "Audio":[
      {
         "Location":"25.wav"
      }
   ]
}
```

ZIP format

The ZIP format is handled differently than CSV or JSON, in that it is a collection of files that are processed as if they were individual files within the multipart request.

- The name of the file is the key that needs to be referenced in the CSV/JSON.
- Any folder structure in the ZIP file is flattened and ignored.

For example, a multipart request looks like the following.

```
batch.zip
   batch.csv (contains 2 rows, for call1 and call2)
   call1.wav
   call2.wav
```

This multipart request is processed as if the files were all in the ZIP or all individually in the multipart request.

Notes

- The order of files does not matter.
- An upload for a contact that contains a recording file name but does not include that recording will fail to be inserted.

- An upload that contains a recording the is not referenced in a CSV or JSON will ignore that recording.

IMPORTANT You must have Tenant Administrator access, access to the data server to use for the Bulk Contact Import, and the Bulk Import permission checked for your role.

Using the Data Server for bulk contact import

You can use the Bulk Contact Import API to upload contacts in bulk through the Data Server.

Using the Bulk Contact Import API requires the following:

- Webex WFO Administrator role with the Bulk Import permission enabled.
- Read/write access to the Data Server.

To upload contacts in bulk through the Data Server:

PREREQUISITE For a bulk import to successfully upload files using a data server, the data server must be configured within Webex WFO. This can be a Data Server that is already being used for any other purpose or a new Data Server. If you are configuring a new Data Server for bulk import, see the topic, “Data Server Configuration” in the *Webex WFO User Guide*.

1. To use a data server that is already configured in Webex WFO navigate to the **Data Server Configuration** page (Application Management > System Configuration > Data Server Configuration) and select that data server.
2. In the **Regional Data Server ACD Sync Settings** section, ensure that **Enable Capture** is selected and that the **Generic (Default)** ACD is assigned.
3. Click **Save**.
4. Create a CSV (not JSON) file that contains all required fields, plus any optional ones that you want to add.
5. Prefix the file with the word CONTACT. The word is case-sensitive, and you must type it in upper-case.

   EXAMPLE
   Your CSV file is named ExampleContacts.csv. You must rename it with the CONTACT prefix as follows:

   CONTACT.ExampleContacts.csv

6. Place the CSV file and all associated media files in the GIS <tenant> folder on the Data Server. This folder is in the location defined by the Regional Data Server GIS File Location field on the Data Server Configuration page.

   EXAMPLE C:\Program Files\Common Files\Webex WFO\Data Server\gis\<tenant>

Bulk Organization Import API

The Bulk Organization Import API is used by third parties to upload their contact center's organizational structure (groups, teams, and people) into Webex WFO.

The API only inserts and updates groups, teams, and people. It never deletes, so it is safe to import a subset of the organizational structure. The import is repeatable because the API modifies only those fields that change. It is safe to import the same file multiple times.

The import handles groups, teams, and persons as follows.

> **IMPORTANT** If you are uploading a CSV file for a Bulk Organization Import for groups or teams, the column headers must match the corresponding request and response fields in the tables below exactly. You cannot reorder or omit any of the columns, even if they are not required for each individual entry.

Groups API Protocol and URI

URI	`/api/upload/groups`
Method	POST
Permissions	Bulk Import
Content Type	multipart/form-data

Groups. A group is identified by the groupName column. If the value in groupName matches an existing group, that group is used. If groupName is new, a group is created with that name and is activated.

Groups API Request and Response Fields

Name	Req?	Description
id	Y	The group's ID in the ACD. You can identify the ID number by exporting group data and viewing it in the resulting CSV file. **NOTE** If the group is a new group the ID must be set to 0 (zero), or the import will fail.
name	Y	The name of the group. Max characters = 60 Default = none
activatedDate	Y	The date the group was activated in the format MM/DD/YYYY. Default = current date
deactivatedDate	Y	The date the group was deactivated in the format MM/DD/YYYY.

Teams API Protocol and URI

URI	`/api/upload/teams`
Method	POST
Permissions	Bulk Import
Content Type	multipart/form-data

Teams. A team is identified by the teamName column. If the value in teamName matches an existing team, that team is used. If the group this team is under has changed, that relationship is updated. If teamName is new, a team is created with that name under the group specified and is activated.

Teams API Request and Response Fields

Name	Req?	Description
id	Y	The team's ID in the ACD. You can identify the ID number by exporting team data and viewing it in the resulting CSV file. **NOTE** If the team is a new team the ID must be set to 0 (zero), or the import will fail.
name	Y	The name of the team. Max characters = 60
activatedDate	Y	The date the group was activated in the format MM/DD/YYYY. Default = current date
deactivatedDate	Y	The date the group was deactivated in the format MM/DD/YYYY.
parentGroupId	Y	The name of the group to which the team belongs.
productivityCompilation	Y	Indicates if the team's statistics are to be included when the Capture service compiles productivity statistics (True/False). Default = true

Persons API Protocol and URI

URI	`/api/upload/persons`
Method	POST
Permissions	Bulk Import
Content Type	multipart/form-data

Persons. A person is identified by email, adLogin, or acdId. If any of these values match an existing person, that person is used. If that person's team membership, name, time zone, locale, or active status has changed, it is updated accordingly. If a matching person is found, the API never blanks out one of the three identifiers

if a value already exists. For example, if a match is found on email address but the imported CSV file does not contain an ACD ID for the user, but we already have one in the database, the existing value is not overwritten. If all three of the person identifiers are new, a new person is created and is given the default agent role.

Persons API Request and Response Fields

Name	Req?	Description
personId	Y	The person's ID in Webex WFO. You can identify the ID number by exporting person data and viewing it in the resulting CSV file. Use 0 (zero) if this is a new person.
email	Y	The user's email address. Max characters = 254 Default = none
firstName	Y	The user's first name. Max characters = 50 Default = none
lastName	Y	The user's last name. Max characters = 50 Default = none
timeZone	N	The agent's time zone. Default = Customer's time zone For a full list of time zone names see Importing and Exporting Users. **EXAMPLE** America/Chicago
adLogin	N	The user's Active Directory login name. Max characters = 50 Default = none

Name	Req?	Description
memberGroup	N	The team to which the user belongs.
activated	N	The date the user was activated in the format MM/DD/YYYY. Default = current date
deactivated	N	The date the user was deactivated in the format MM/DD/YYYY. **NOTE** To activate a deactivated user when importing users, set this date to 12/31/2999.
locale	N	The language used by the user. Default = en
roles	N	The roles assigned to the user. Multiple roles are separated by semicolons.
scopegroups	N	The groups within the user's scope. Multiple groups are separated by semicolons.
scopeteams	N	The teams within the user's scope. Multiple teams are separated by semicolons.
enableScheduling	N	Indicates if the user can be scheduled in True or False format (WFM only). Default = false
views	N	The views that the user is assigned in WFM (WFM only). Default = none
mainView	N	The user's main view in WFM (WFM only). Default = none
companyStartDate	N	The date the user started with the company in the format MM/DD/YYYY (WFM only). Default = none

Name	Req?	Description
companyEndDate	N	The date the user left the company in the format MM/DD/YYYY (WFM only). Default = none
departmentStartDate	N	The date the user started with the department in the format MM/DD/YYYY (WFM only). Default = none
employeeId	N	The user's employee ID (WFM only). Default = none
rank	N	The user's rank within the contact center (WFM only). Default = none
qmViews	N	The views assigned to the user in QM (QM only). Default = none
isReconcileOnly	N	When selected, the agent uses gateway recording and all the agent's calls require post-call reconciliation (QM only). Default = none
scopetenant	N	When set to TRUE, the user has scope over all groups, teams, and users. When set to FALSE, the user's scope is as configured in the Scope Teams and Scope Groups fields (True/False). Default = none
acdId	Y	The user's ID within the ACD. Default = none **NOTE** This field is required if you selected the "Enable scheduling for this user" check box.
acdServerId	Y	The ID of the ACD that is the source of data for the user.

Name	Req?	Description
		Default = none **NOTE** This field is required if you selected the "Enable scheduling for this user" check box.

CSV File Examples

Groups CSV Example

```
1 id,name,activatedDate,deactivatedDate
2 0,Sales,01/04/2018,01/01/2019
3 0,Finance,01/04/2018,0180182019
```

Teams CSV Example

```
1 id,name,activatedDate,deactivatedDate,parentGroupId,productivityCompilation
2 0,Car Sales,01/04/2018,01/01/2019,Sales,True
3 0,Parts Sales,01/04/2018,01/01/2019,Sales,False
```

Bulk Report Data Export API

The Bulk Report Data Export API allows you to retrieve data from reports in Data Explorer. If the data you request is too large to return instantly, it provides you with a request ID to retrieve data once it is ready. After configuring filters in the report in Data Explorer, you can adjust the report parameters in the API call.

NOTE You can export a maximum of 1,000 records in each export request. If you need to export more than 1,000 records, create multiple, more restricted exports.

GET Bulk Report Data Export API

The GET method for the Bulk Report Data Export API allows you to check on the status of a previous request or retrieve the data from an asynchronous request using a request ID.

NOTE A request ID is contained in the response to a GET or POST action if the requested data is too large or is not able to be returned quickly.

GET Bulk Report Data Export API Protocol and URI

URI	/api/rest/dataexplorer/report/export/request/{requestId}
Method	GET
Permission	ReportAuthoring
Content Type	Text/CSV
Parameters	requestId

GET Bulk Report Data Export API Request Fields

Name	Required	Description
Request ID	Y	The Request ID that was returned when the POST action was taken

GET Bulk Report Data Export API JSON File Example

The following is an example of a formatted JSON request.

```
{
   "schemaName": "com.stytch.rest.api.v4.data.report.action.ExportActionData",
   "action": "EXPORT_CSV", "state": {
   "schemaName": "com.stytch.rest.api.v4.data.report.ReportSpecData",
   "qname": "RP1_217_0_getk05bhup57b3nmd3ill"
         },
         "paramValues": {
         "qnameContent": "RP1_217_0_getk05bhup57b3nmd3ill",
   "encodedParamValues": "[]"
         },
         "pageSpecification": [
         {
         "offset": 0,
         "limit": 100
   },
         {
         "offset": 0,
         "limit": 100
         }
   ]
}
```

POST Bulk Report Data Export API

The POST method for the Bulk Report Data Export API returns the data from a report in Data Explorer. If the data requested is small enough to be returned quickly, the response body includes the report data in CSV format. If the requested data is too large or cannot be returned quickly, the response body contains request ID which can be used to retrieve the data later.

IMPORTANT This API is deprecated. It is available only to existing Cisco customers who are already using it. If you want to use this API, please contact Cisco Support.

Protocol and URI

URI	/api/rest/dataexplorer/report/{reportId}/export
Method	POST
Permission	ReportAuthoring
Content Type	Text/CSV
Parameters	reportId

POST Request Fields

Name	Description
Schema Name	The schema name for a part of the API. These should be copied from the JSON File Example.
Action	The task the API should complete. In this case, "Export_CSV."
QName	The qualified name. This information can be found by opening the report in Data Explorer and then opening your browser console. Search for the string **RENDER_REACT_REPORT** in the console to find the QName.
Encoded Parameter Values	The values that you will be changing from the default report. The parameters that are entered here must first be configured in the report in Data Explorer. See Configuring a Report Filter.
Offset	The distance from the top, left-most cell in the table to the cell that the API should start returning data from. The first instance of offset in the body request indicates rows, and the second instance indicates columns.
Limit	The largest number of cells the API should return. The first instance of limit in the body request indicates rows, and the second instance indicates columns.

JSON File Example

```
{
   "schemaName": "com.stytch.rest.api.v4.data.report.action.ExportActionData",
   "action": "EXPORT_CSV", "state": {
      "schemaName": "com.stytch.rest.api.v4.data.report.ReportSpecData",
      "qname": "RP1_22_0_8o448kr6ll3sbkgvvfkj7"
   },
   "paramValues": {
      "schemaName":
   "com.stytch.rest.api.v4.data.parameters.EncodedParameterValuesData",
      "encodedParamValues": "[{‘name’:‘Team’,‘value’:{‘isAll’:‘false’,‘values’:
   [122.0],‘containsNull’:‘false’}}]",
      "qnameContent": "RP1_217_0_getk05bhup57b3nmd3ill"
   },
   "pageSpecification": [
      {
         "offset": 0,
         "limit": 100
      },
      {
         "offset": 0,
         "limit": 100
      }
   ]
```

Response Fields

There are two possible responses to this API. If the data can be returned quickly, a synchronous response contains the requested data in CSV format. If the requested data cannot be returned quickly, the response will contain a JSON response with a request ID and the status of the request. These outputs includes fields defined in the following tables.

Synchronous Response

Field	Description
Content Type	The format of the content. Only CSV format is available.
Content Extent	The total number of rows available and the total number of columns available.
Body	The report data in CSV format.

Asynchronous Response

Field	Description
Request ID	The format of the content. Only CSV format is available.
Status	The current state of the request. **NOTE** A message that cannot be processed immediately goes into a message queue and is stored as a job in the database, so it can be tracked. The status field displays the status of the job at the time of the call. One of the following statuses will be displayed. ■ Queued—The request is successfully captured in the Message Queue and database. ■ Processing—The queued messaged is being processed but has not reached one of the final states. ■ Failed—The request could not be completed. ■ Canceled—The request was stopped. ■ Expired—The time allowed for the job ran out before it was processed. ■ Blacklisted—The number of calls has exceeded the limit, and this call will not be returned.

File Example

The following is an example of a formatted request.

JSON Request

```
{
      "schemaName":
    "com.stytch.rest.api.v4.data.report.action.ExportActionData",
      "action": "EXPORT_CSV",
      "state": {
         "schemaName":
   "com.stytch.rest.api.v4.data.report.ReportSpecData",
         "qname": "RP1_217_0_getk05bhup57b3nmd3ill"
         },
      "paramValues": {
         "qnameContent": "RP1_217_0_getk05bhup57b3nmd3ill",
         "encodedParamValues": "[]"
      },
      "pageSpecification": [
         {
            "offset": 0,
            "limit": 100
         },
         {
            "offset": 0,
            "limit": 100
         }
      ]
   }
```

Configuring a Report Filter

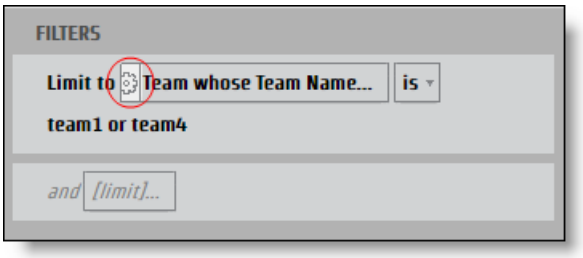
One of the features of the Bulk Report Data Export API is the ability to change the filters on a report when calling the API. You can configure the filters by editing the text of the POST request. In order to change a filter using the API, you must make a few changes to the report in Data Explorer beforehand.

Configure the Parameter in Data Explorer

To set up a filter so that it can be updated through the API call, you must edit the filter's parameters in Data Explorer.

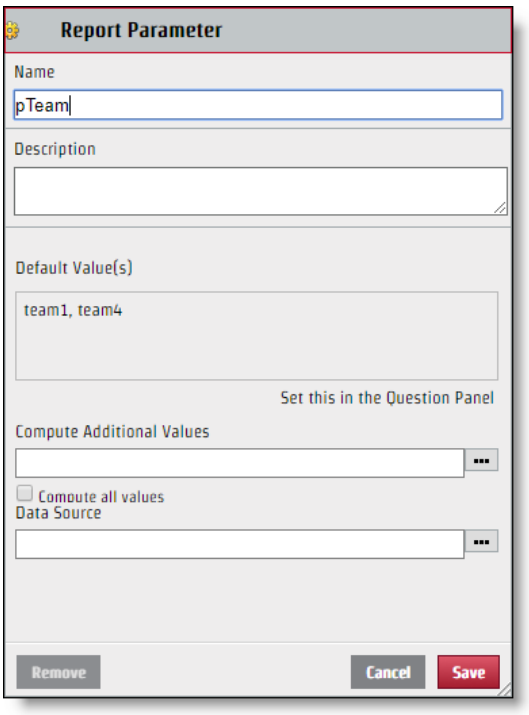
Configure the parameter

1. Open the report you want to export from the Reports list in Data Explorer.
2. Click **Edit** to open the Question Panel.
3. In the Filters section of the Question Panel, click the Parameter icon beside the filter you want to update. The Parameter icon looks like part of a gear, and it is located at the left side of the Filter field. The following image shows the Parameter icon circled in red.



The **Report Parameter** dialog box opens.

4. Enter a name in the **Name** field for the Parameter that can be used as a JSON key.

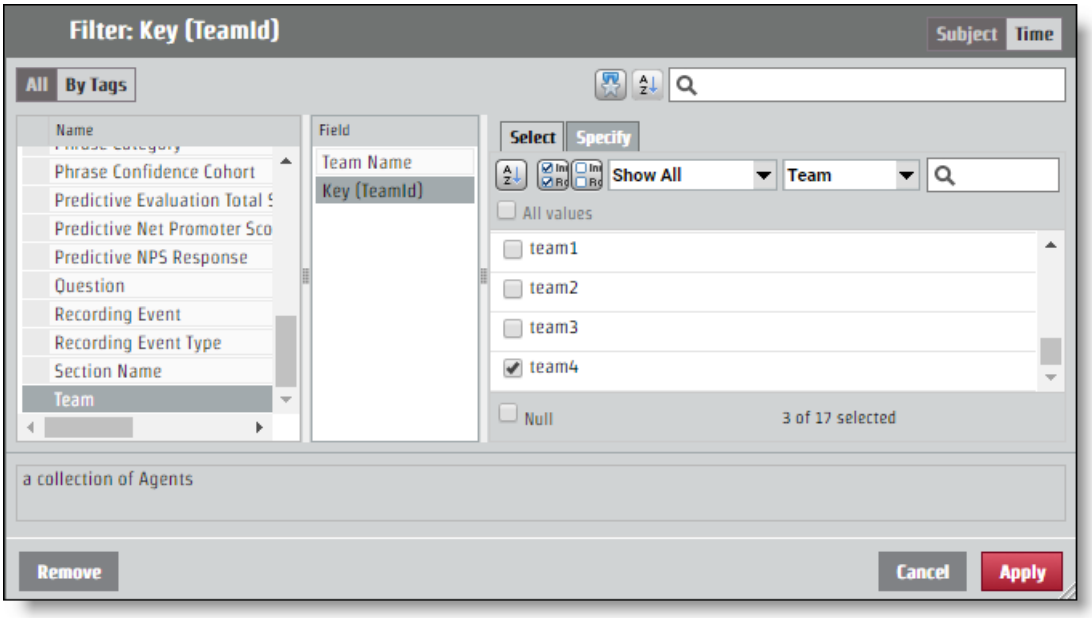


5. Click **Save**.

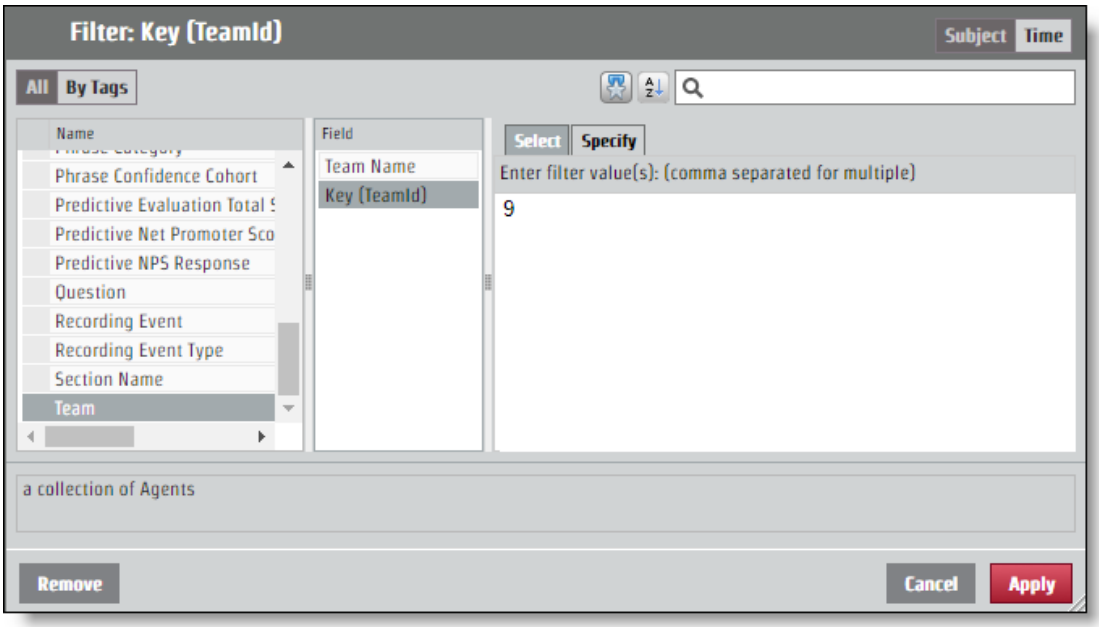
Once the filter is configured, it can be edited through the API request. Though the Parameter name you create is used for the key name in the JSON key value pair, the value that is displayed in the Select section of the Filter dialog box is not necessarily the value that you should use for the JSON file. To ammend the JSON file correctly, you need to identify the specific name of the value.

Identify the specific value

1. Click the **Filter** field with values to identify from the Question panel to open the Filter dialog box.
2. Select only the check box beside the value you want to identify.



3. Open the **Specify** tab. The specific value appears in the Filter Value CSV pane.



Configure JSON for the API Request

The key value pair that you determined from naming the parameter and identifying the value needs to be added to the JSON request. Changes to the filter are made through the **encodedParamValues** field.

If your report returns information for all teams, but you want information from only the team with the specific value "9", the encodedParamValues JSON should look like the following example. In this example, the parameter was named "pTeam," so that is the key in the JSON file.

```
...
"encodedParamValues": "[{\"name\":\"pTeam\",\"value\":
   {\"isAll\":\"false\",\"values\":[9.0],\"containsNull\":\"false\"}}]",
...
```

Configuring Range Filters

The process for configuring a parameter to return information from a certain time range works just the same as editing the encodedParamValues field for any other filter. There are several specific time periods like year, month, and quarter. The specific time periods have a **Specify** tab that works like any other filter value. There are also options for ranges: Relative Date and Date Range. Each start time option requires different values in the JSON file.

Date Range

The Date Range option uses exact dates to determine the range. The API uses UNIX Epoch time, so any date and time that you want to use as a parameter for a range must be converted.

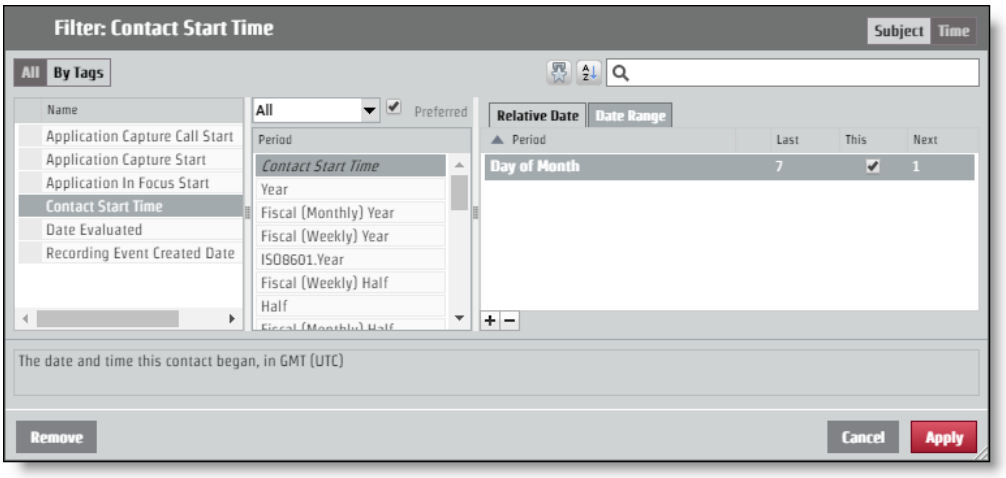
> **BEST PRACTICE** A report that doesn't specify an end range will pull all information from the range start time forward. For recurring reports, you can use this feature and take the time of the last item in the report, which will be formatted in UNIX Epoch time and use that number as the start of the new range. In this way, you can get all of the information you need and never have to translate times into UNIX Epoch time.

For a request that configures a time range of June 15, 2017, the encodedParamValues should look like the following example. In this example, the parameter name is "pDate".

```
...
"encodedParamValues": "[{\"name\":\"pDate\",\"value\":
   {\"containsNull\":\"false\",\"values\":
   [{\"periodType\":\"DayOfMonthPeriod\",\"referenceTime\":1497528000000,\"range
   Start\":0,\"rangeEnd\":1}]}}]",
...
```

Relative Date

The Relative Date option creates a range without specific dates. This includes time periods that are relative to the current date. In the Filter dialog box, you can select a time period and determine whether you want some number of time periods before the current one, this current time period, and some number of future time periods. In the following example, the time range is set for the last seven days, today, and tomorrow.



In order to request this layout through the API request, the encodedParamValues should look like the following example. In this example, the parameter name is "pContactStartTime."

```
...
"encodedParamValues": "[{\"name\":\"pContactStartTime\",\"value\":
   {\"containsNull\":\"false\",\"value\":
   [{\"periodType\":\"DayOfMonthPeriod\",\"referenceTime\":\"NL\",\"rangeStar
   t\":-7,\"rangeEnd\":-1},
   {\"periodType\":\"DayOfMonthPeriod\",\"referenceTime\":\"NL\",\"rangeStart\":
   0,\"rangeEnd\":0},
   {\"periodType\":\"DayOfMonthPeriod\",\"referenceTime\":\"NL\",\"rangeStart\":
   1,\"rangeEnd\":1}]}}]",
...
```

Categories and Phrases API

Use this API to work with category and phrase lists from .

These methods are used for the Categories and Phrases API:

- Get—Retrieves category and phrase information
- Post—Creates categories and phrases
- Put—Updates category and phrase information
- Delete—Deletes category and phrase information

DELETE Categories and Phrases API

This method deletes category and phrase information.

DELETE Protocol and URIs

URIs:	
Category	`/api/rest/cas/phraselist`
Phrase	`/api/rest/cas/phrase`
Method	DELETE
Permissions	Admin Desktop Analytics Admin Audio Analytics Admin LVSCR Analytics Admin Text Analytics Administer Tenant
Content Type	multipart/form-data

EXAMPLE

Deleting a category: `/api/rest/cas/phraselist?id=13`

Deleting a phrase: `/api/rest/cas/phrase?id=7`

The system responds with the following:

```
{"deleted":1}
```

GET Categories and Phrases API

This method retrieves category and phrase information.

GET Protocol and URIs

URIs:	
Category	`/api/rest/cas/phraselist`
Phrase	`/api/rest/cas/phrase`
Method	GET
Permissions	Admin Desktop Analytics Admin Audio Analytics Admin LVSCR Analytics Admin Text Analytics Administer Tenant
Content Type	multipart/form-data

GET Response Fields

Name	Description
categories	Detail information regarding Analytics categories.
id	Category ID.
description	Description of the category.
languageId	Character string specifying the language. **EXAMPLE** US English = en-us

Name	Description
phrases	Detail information regarding Analytics phrases.
phrase	Phrase to be searched.
phonetic	Phonetic spelling of the phrase to be searched.
soundsLike	Alternate versions of the phrase that sound like the target phrase.
confidence	Confidence value indicating the probability that the hit is correct.
id	ID number of the phrase.
categoryId	Category ID number for the phrase.
state	Agent's contact state.

POST Category and Phrases API

This method creates category and phrase information.

POST Protocol and URIs

URIs:	
Category	`/api/rest/cas/phraselist`
Phrase	`/api/rest/cas/phrase`
Method	POST
Permissions	Admin Desktop Analytics Admin Audio Analytics Admin Lvcsr Analytics Admin Text Analytics Administer Tenant
Content Type	multipart/form-data

POST Response Fields

Name	Description
category	Detail information regarding Analytics categories.
description	Description of the category.
languageId	Character string specifying the language. **EXAMPLE** US English = en-us

PUT Categories and Phrases API

This method updates category and phrase information.

PUT Protocol and URIs

URIs:	
Category	`/api/rest/cas/phraselist`
Phrase	`/api/rest/cas/phrase`
Method	PUT
Permissions	Admin Desktop Analytics Admin Audio Analytics Admin LVSCR Analytics Admin Text Analytics Administer Tenant
Content Type	multipart/form-data

PUT Response Fields

Name	Description
categories	Detail information regarding Analytics categories.

Name	Description
id	Category ID.
description	Description of the category.
languageId	Character string specifying the language. **EXAMPLE** US English = en-us
phrases	Detail information regarding Analytics phrases.
phrase	Phrase to be searched.
phonetic	Phonetic spelling of the phrase to be searched.
soundsLike	Alternate versions of the phrase that sound like the target phrase.
confidence	Confidence value indicating the probability that the hit is correct.
id	ID number of the phrase.
categoryId	Category ID number for the phrase.
state	Agent's contact state.

GET/POST/PUT JSON File Example

The following is an example of a formatted JSON response file for retrieving, creating, and updating categories and phrases:

```
{
        "categories": [{
                    "id": number,
                    "category": string,
                    "description": string,
                    "languageId": "en-us"
              },
              ....
        ],
        "phrases": [{
```

```
                        "phrase": string,
                        "phonetic": string,
                        "soundslike": string,
                        "confidence": number,
                        "id": number,
                        "categoryId": number,
                        "state": number
                },
                ...
        ],
        "confidence": number
}
```

Contact Device API

Use the Contact Device API to retrieve a list of devices.

Contact Device API Protocol and URI

URI	/api/rest/recording/contactDevice
Method	GET
Permissions	Administer QM

Parameters

Field	Req?	Type	Description
includeUnconfiguredDevices	N	Boolean	When set to true, unconfigured devices are included. Default = false Accepted values. ■ true ■ false
itemsPerPage	N	Number	The number of items that appear per page.
sortColumn	N	Text	personId
sortDirection	N		Options ■ ascending ■ descending

Contact Device API JSON File Examples

Response File Example

```
[{
```

```
        "id": 1,
        "name": "123456789",
        "contactDeviceTypeId": 0,
        "recordingTypeId": 1,
        "configured": true,
        "personId": null,
        "telephonyGroupId": 2,
        "recordingGroupId": 2,
        "signalingGroupId": 3,
        "loggedInPersonId": null,
        "virtualDevice": null,
        "isRecordingTones": false,
        "isStereo": false,
        "monitoringServerId": null,
        "contactDeviceLines": [{
                "id": 49,
                "contactDeviceId": 1,
                "extension": "8013",
                "partition": "Route Partition"
            }
        ]
    }
```

Contact Device Bulk API

Use the Contact Device Bulk API to add devices in bulk.

Contact Device Bulk API Protocol and URI

URI	/api/rest/recording/contactDeviceBulk
Method	POST
Permissions	Administer QM

See Contact Device API for more information.

Contact Device BULK API JSON File Examples

Response File Example

```
{
    "add": [
        {
            "configured": true,
            "contactDeviceLines": [
                {"contactDeviceId": 239, "extension": "7003", "id": 424032, "partition": ""},
                {"contactDeviceId": 239, "extension": "7103", "id": 424033, "partition": ""}
            ],
            "contactDeviceTypeId": 0,
            "id": 239,
            "isRecordingTones": false,
            "isStereo": false,
            "loggedInPersonId": 19,
            "monitoringServerId": null,
            "name": "SEPWIN7X64003",
            "personId": 19,
            "recordingGroupId": 1,
            "recordingTypeId": 1,
            "signalingGroupId": 1,
            "telephonyGroupId": 1,
            "virtualDevice": null
        },
        {
            "configured": true,
            "contactDeviceLines": [
                {"contactDeviceId": 238, "extension": "7004", "id": 412459, "partition": ""},
```

```
                {"contactDeviceId": 238, "extension": "7104", "id": 412458,
"partition": ""}
            ],
            "contactDeviceTypeId": 0,
            "id": 238,
            "isRecordingTones": false,
            "isStereo": false,
            "loggedInPersonId": 29,
            "monitoringServerId": null,
            "name": "SEPWIN7X64004",
            "personId": 29,
            "recordingGroupId": 1,
            "recordingTypeId": 1,
            "signalingGroupId": 1,
            "telephonyGroupId": 1,
            "virtualDevice": null
        }
    ],
    "delete": [
        {
            "configured": true,
            "contactDeviceLines": [
                {"contactDeviceId": 108, "extension": "8004", "id": 424040,
"partition": ""},
                {"contactDeviceId": 108, "extension": "8104", "id": 424041,
"partition": ""}
            ],
            "contactDeviceTypeId": 0,
            "id": 108,
            "isRecordingTones": false,
            "isStereo": false,
            "loggedInPersonId": 30,
            "monitoringServerId": null,
            "name": "SEPWIN8X64004",
            "personId": 30,
            "recordingGroupId": 1,
            "recordingTypeId": 1,
            "signalingGroupId": 1,
            "telephonyGroupId": 1,
            "virtualDevice": null
        },
        {
            "configured": true,
            "contactDeviceLines": [
                {"contactDeviceId": 99, "extension": "8003", "id": 424037,
"partition": ""},
                {"contactDeviceId": 99, "extension": "8103", "id": 424036,
"partition": ""}
            ],
            "contactDeviceTypeId": 0,
            "id": 99,
            "isRecordingTones": false,
            "isStereo": false,
            "loggedInPersonId": 22,
            "monitoringServerId": null,
            "name": "SEPWIN8X64003",
```

```
            "personId": 22,
            "recordingGroupId": 1,
            "recordingTypeId": 1,
            "signalingGroupId": 1,
            "telephonyGroupId": 1,
            "virtualDevice": null
        }
    ],
    "update": [
        {
            "configured": true,
            "contactDeviceLines": [{"contactDeviceId": 88, "extension": "9504",
"id": 412466, "partition": ""}],
            "contactDeviceTypeId": 0,
            "id": 88,
            "isRecordingTones": false,
            "isStereo": false,
            "loggedInPersonId": 31,
            "monitoringServerId": null,
            "name": "CSFWIN10X64004",
            "personId": 31,
            "recordingGroupId": 1,
            "recordingTypeId": 1,
            "signalingGroupId": 1,
            "telephonyGroupId": 1,
            "virtualDevice": null
        },
        {
            "configured": true,
            "contactDeviceLines": [{"contactDeviceId": 86, "extension": "9503",
"id": 412511, "partition": ""}],
            "contactDeviceTypeId": 0,
            "id": 86,
            "isRecordingTones": false,
            "isStereo": false,
            "loggedInPersonId": 23,
            "monitoringServerId": null,
            "name": "CSFWIN10X64003",
            "personId": 23,
            "recordingGroupId": 1,
            "recordingTypeId": 1,
            "signalingGroupId": 1,
            "telephonyGroupId": 1,
            "virtualDevice": null
        }
    ]
}
```

Contact API

The Contact API allows users to search for specific contacts, then link to Webex WFO to play them back in the Media Player.

Protocol and URI

URI	`/api/rest/recording/contact/<id>`
Method	GET
Permissions	Search Contacts
Content Type	multipart/form-data
Parameters	ID = identifier for an existing contact. This resource URI locates a specific contact ID. If the {id} does not exist, the response returns an error code instead of an empty list. beginTime = <beginning timestamp for the contacts>

Request Fields

NOTE If the beginTime parameter is used, then the endTime parameter must also be used in the URI query.

Parameter	Description
agent	The user's ID. The accepted value is an integer ID. The parameter ("agent") is case sensitive and must be all lowercase. If a user uses the parameter with a capital A, it returns every agent. OPTIONAL
ani	The automatic number identification (ANI) for a call. In other words, ANI identifies the number of the calling party. The accepted value is string with asterisk (*) or question mark (?) wildcards.

Parameter	Description
	This parameter can appear zero or more times in a single query. **EXAMPLE** `contact?ani=1234&ani=4567` When you provide multiple values for a parameter, the query combines these values with OR (that is, the previous example searches for contacts with either ani=1234 OR ani=4567). OPTIONAL.
assocCallId	The Associated Contact ID that ties together contacts based on a customer experience. OPTIONAL.
beginDate	Return only records that start on or after the specified date. FORMAT: `YYYY-MM-DD`. If no range is specified and beginDate is not specified, the default is 60 days before today.
beginTime	Used with beginDate to return records that start on or after the specified date and time. FORMAT: `HH:MM`.
callId	A contact ID from a third-party integration. The accepted value is string. OPTIONAL.
canEvaluate	Indicates whether the user can evaluate this contact. The accepted value is true. True returns the contact, if the user can evaluate this contact when logged into Webex WFO. OPTIONAL. If you do not include this parameter, the query does not filter the contacts. VALUE: True, if the contact can be evaluated by the logged-in user.
dateEvaluatedEnd	The end date of the range of filtered contacts that were evaluated. FORMAT: `YYYY-MM-DD`.
dateEvaluatedStart	The start date of the range of filtered contacts that were evaluated. FORMAT: `YYYY-MM-DD`.
dnis	The dialed number identification server (DNIS) for the call. In other words, the called number. The accepted value is string with asterisk (*) or question mark (?) wildcards. This parameter can appear zero or more times in a single query. When

Parameter	Description
	you provide multiple values for a parameter, the query combines these values with OR. OPTIONAL.
endDate	Return only records that start before the specified date. FORMAT: `YYYY-MM-DD`. If no range is specified and the endDate is not specified, the default is today.
endTime	Used with endDate to return records that start before the specified date. FORMAT: `HH:MM:SS`.
exclude	Returns a Uniform Resource Identifier (URI) for the specified value. Click the URI to see the data associated with the specified value. OPTIONAL. VALUES: ■ event—Returns a URI that points the event data. ■ metadata—Returns a URI that points to the metadata. ■ metadata.key1—Returns a URI that points to the key1 attribute within the metadata object.
expand	Returns all data associated with the specified value instead of a URI. OPTIONAL. VALUES: ■ event—Expands all events to include all event data, not just the URI. ■ metadata—Expands all metadata, not just the URI. ■ metadata.key1—Only expands the key1 attribute within the metadata object.
firstName	The agent's first name. The accepted values is string with any number of asterisk (*) or question mark (?) wildcards. OPTIONAL.
group	A group's ID. The accepted value is integer ID. OPTIONAL.
hasRecording	Return only contacts associated with a recording. OPTIONAL. VALUES: ■ true ■ false

Parameter	Description
	If you do not include this parameter, the query does not filter contacts by recordings.
hr	Whether the contact evaluation has been marked for human resources (hr). OPTIONAL. VALUES: ■ true ■ false ■ 1 (true) ■ 0 (false)
inProgress	Whether the contact evaluation is in progress. OPTIONAL. VALUES: ■ true ■ false ■ 1 (true) ■ 0 (false)
lastName	The agent's last name. The accepted values is string with any number of asterisk (*) or question mark (?) wildcards. This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. OPTIONAL.
limit	A URL parameter that can be used to increase the total results and to get larger result sets. OPTIONAL. **IMPORTANT** Use of this parameter can negatively effect performance. Cisco recommends 100 or fewer.
line	The extension for the call (from the perspective of the agent who is recording the call). The accepted value is string with asterisk (*) or question mark (?) wildcards. This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these

Parameter	Description
	values with OR. OPTIONAL.
metadata	The metadata field. OPTIONAL. VALUES: ■ <key>~<operator>~<value> Where <key> is the name of the metadata field configured in Monitoring and Recording Administrator and must match the configured metadata name exactly. The <operator> is one of the following: ■ equals ■ beginsWith ■ endsWith ■ contains The <value> can include a string with asterisk (*) or question mark (?) wildcards. This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. If you do not specify a value, the query returns all contacts that have metadata for the specified key.
needsApproval	Whether the contact evaluation needs approval. OPTIONAL. VALUES: ■ true ■ false ■ 1 (true) ■ 0 (false)
number	Any number used in the contact (ANI or DNIS). The accepted value is string with asterisk (*) or question mark (?) wildcards. This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. OPTIONAL.

Parameter	Description
phoneNumber	Any number used in the contact (ANI, DNIS, or Line). The accepted value is string with asterisk (*) or question mark (?) wildcards. This parameter can appear zero or one time in a single query. OPTIONAL.
range	A header parameter that allows you to page through results. Where, "items=<1-200>". OPTIONAL. **BEST PRACTICE** The range parameter can be used in conjunction with the limit query parameter, which increases the total results.
scored	Whether the contact has been scored. OPTIONAL. VALUES: ■ true ■ false ■ 1 (true) ■ 0 (false)
searchStats	Returns the total number of available records that match as the "count". Where accepted values are: ■ true ■ false OPTIONAL.
silenceDuration	Return only contacts where the duration (in milliseconds) of recorded silence is equal to or greater than the value specified. **EXAMPLE** silenceDuration=5000 returns contacts where there are one or more instances of recorded silence equal to or greater than 5 seconds. OPTIONAL.
silenceEvents	Return only contacts where the number of silence events are equal to or

Parameter	Description
	greater than the value specified. **EXAMPLE** silenceEvents=5 returns contacts where there are 5 or more silence events. OPTIONAL
tagged	Whether the contact was tagged. The accepted Boolean values are: ■ true ■ false ■ 1 (true) ■ 0 (false) OPTIONAL.
talkOverDuration	Return only contacts where the duration (in milliseconds) of recorded talk over events are equal to or greater than the value specified. **EXAMPLE** talkOverDuration=5000 returns contacts where there are one or more instances of recorded talk over events equal to or greater than 5 seconds. OPTIONAL.
talkOverEvents	Return only contacts where the number of talk over events is equal to or greater than the value specified. **EXAMPLE** talkOverEvents=5 returns contacts in which there are 5 or more talk over events. OPTIONAL.
team	A team's ID. The accepted value is integer ID. OPTIONAL.
training	Whether the contact evaluation has been marked for training. OPTIONAL. VALUES: ■ true

Parameter	Description
	■ false ■ 1 (true) ■ 0 (false)
type	The type of contact. The type parameter filters contacts based on upload states. OPTIONAL. VALUES: ■ quality ■ archive If you do not include this parameter, the query does not filter on upload states. **NOTE** The archive user role only has global scope when you specify the archive type.

Performing a Combination Search

The following request returns a list of all contacts recorded for quality purposes on or after 01/01/2018 (GMT).

```
GET ~/api/rest/recording/contact?beginDate=2018-01-01&type=quality
```

Performing a Search Using a Metadata Key

The following request returns a list of all contacts with the metadata called phone.

```
GET ~/api/rest/recording/contact?beginDate=2018-01-01&metadata=phone
```

The following returns a list of all contacts with the metadata called customerNo.

```
GET ~/api/rest/recording/contact?beginDate=2018-01-
01&metadata=phone~contains~*
```

Performing a Search Using a Metadata Key With a Specific Value

The following request returns a list of all contacts with the metadata value of 555-1234.

```
GET ~/api/rest/recording/contact?beginDate=2018-01-
01&metadata=phone~equals~555-1234
```

Performing a Search For Silence Events

The following request returns a list contacts that contain two or more silence events of 10 seconds or greater.

```
GET~/api/rest/recording/contact?silenceEvents=2&silenceDuration=10000
```

JSON File Example


```
{
  "hr" : false,
  "dnis" : "2111",
  "callDuration" : 15000,
  "ani" : "1581",
  "icmCallId" : "19291093",
"assocCallId" : 290275383991253,
  "evalForm" : {
    "name" : "form1",
    "$ref" : "/api/rest/recording/evalform/5"
  },
    "team" : {
    "name" : "team1",
    "$ref" : "/api/rest/recording/team/1",
 "displayId" : "0.4"
  },
  "qualityReason" : {
    "text" : "First Call of Day",
    "reasonId" : 1,
    "key" : "rec_reason_what_first"
  },
  "agent" : {
    "lastName" : "Bunkowske",
    "username" : "bunkowm",
    "$ref" : "/api/rest/recording/person/1",
    "firstName" : "Mark",
    "displayId" : "0.1"
  },
  "evaluation" : {
    "stateId" : 1,
    "score" : 90,
    "$ref" : "/api/rest/recording/contact/1/eval/1"
  },
  "training" : false,
  "id" : 1,
  "startTime" : 1239308710000,
```

```
  "tz" : "America/Chicago",
  "recordingUrl" : /api/rest/recording/contact/1/recording",
  "audioUploaded" : true,
  "archiveWF" : {
    "$ref" : "/api/rest/recording/workflow/265"
  },
  "group" : {
    "name" : "group1",
    "$ref" : "/api/rest/recording/group/1"
  },
  "evaluator" : {
    "lastName" : "Sillars",
    "username" : "sillarj",
    "$ref" : "/api/rest/recording/person/2",
    "firstName" : "Jay",
    "displayId" : "0.2"
  },
  "screenUploaded" : true,
  "metadata" : {
    "$ref" : "/api/rest/recording/contact/1/metadata/"   },
  "qualityWF" : {
    "$ref" : "/api/rest/recording/workflow/266"
  }
}
```


JSON File Example searchStats Parameter = true


```
{
"avgCallDuration": "7300",
"avgScore": null,
"count": "689"
}
```

Contact Basic Search API

The Contact Basic Search API returns details about an in-progress or most recently completed call. The returned information consists of the most recent contact that matches the parameters of the search. The most recent contact might currently be in progress.

Protocol and URI

URI	`/api/rest/recording/contactBasicSearch`
Method	GET
Permissions	Search Contacts
Content Type	multipart/form-data

Request Fields

Name	Req?	Type	Description
ani	N	String	Calling number of the contact.
dnis	N	String	Called number of the contact. Agent's phone number.
firstName	N	String	Agent's first name.
lastName	N	String	Agent's last name.
line	N	String	The extension for the call (from the perspective of the agent who is recording the call). The accepted value is string with asterisk (*) or question mark (?) wildcards.

Name	Req?	Type	Description
			This parameter can appear zero or more times in a single query. When you provide multiple values for a parameter, the query combines these values with OR. OPTIONAL.
username	N	String	Agent's email address from the Person table.

Response Fields

The JSON output includes fields defined in the following table.

Field	Description
id	CCR database ID of the contact.
assocCallId	Associated contact ID of contact.
callDuration	Duration of the call.
recordingUrl	URL of the contact playback link.
isComplete	Indicates whether this is a completed call or the call is in progress. true = call is complete false = call is in progress
startTime	Timestamp the call started, in GMT.
agent	Detailed agent information.
$ref	URL reference to person resource
displayId	Display ID of the agent.
lastName	Last name of the agent
firstName	First name of the agent.
username	Email address of the agent.

If there are no contacts that meet the given parameters (for example, the provided agent has never taken a call, or is not in scope), the system returns a 404 Not Found response. For contacts not associated with an agent (for example, gateway recorded contacts), the "agent" attribute will be null.

JSON File Example

The following are examples of a formatted JSON response.


```
{
   "id" : 2,
   "assocCallId" : "00001007771411573215",
   "recordingUrl" : "https:\/\/1.1.1.1\/index.html#\/recordings?loadContact=2",
   "isComplete" : false,
   "startTime" : 1481520391000,
   "agent" : {
         "$ref" : "\/api\/rest\/recording\/person\/1234",
         "displayId" : "al dente",
         "lastName" : "dente",
         "firstName" : "al",
         "username" : "al.dente@your-domain.com"
   }
}
```

Contact Monitoring API

The Contact Monitoring API enables users to see the total number of contacts created within the past 15 minutes from the time the administrator calls this API.

All contacts are captured in the database and displayed with this API. However, the number of contacts displayed with this API may differ with the number of contacts shown in the contacts table. For example, depending on how you set the the Global Settings, Call Length parameter determines the number of contacts that are shown in the user interface. The number of contacts shown in the user interface is less than or equal to the total number of contacts shown with this API.

NOTE This API does not count root recordings until they are reconciled.

PROTOCOL AND URI

URI	`/api/rest/monitoring/contacts/lite`
Method	GET
Permissions	Administer Tenant permission
Content Type	application/JSON

RESPONSE FIELDS

The JSON output includes fields defined in the following table.

Field	Description
contacts	A call between an agent and a customer.

Field	Description
total	Number of contacts created in the past 15 minutes.
totalWithAudio	Number of contacts that were created in the past 15 minutes that have audio recordings uploaded.
totalWithVideo	Number of contacts that were created in the past 15 minutes that have screen recordings uploaded.

JSON RESPONSE EXAMPLE

The following is an example of a formatted JSON response.

```
{
   "contacts":{
         "total": "215",
         "totalWithAudio":175
         "totalWithVideo":85
   }
}
```

Evaluation Form API

Use this API to request QM evaluation form information from .

Protocol and URI

URL	`/api/rest/recording/evalform`
Method	GET
Content Type	multipart/form-data

Response Fields

Name	Description
id	Evaluation form ID.
name	Evaluation form name.
description	Description of the evaluation form .
creator	Evaluation form creator.
created	Time the evaluation form was created in UNIX epoch milliseconds.
manualWeighting	Indicates whether manual weighting applies to questions on the evaluation form.
createdTZ	Time zone in which the evaluation form was created.
updater	Updater indicator.
updated	Time indicator when the evaluation form was last updated, in UNIX epoch format.

Name	Description
updatedTz	Time zone in which the evaluation form was updated.
status	Status of the evaluation form.
defaultForm	Indicates whether the evaluation form shows default answer values.
approvalRequired	Indicates whether changes to the evaluation form must be approved.
bandRanges	Detail information regarding the evaluation form band ranges.
band	Evaluation form band number.
begin	Beginning of the evaluation form band.
end	Ending of the evaluation form band.
scoreType	Detail information the evaluation form score.
id	Score ID.
name	Name of the score type. **EXAMPLE** Percentage
sections	Detail information regarding the evaluation form sections.
id	Evaluation form section ID.
name	Evaluation form section name.
order	Order in which the section appears in the evaluation form.
weight	Evaluation form section weight factor.
startColor	Beginning color of the evaluation form section.
endColor	Ending color of the evaluation form section.
questions	Detail information regarding the evaluation form question.
id	Question ID.
text	Text of the question.

Name	Description
order	Order in which the question appears in the section.
weight	Weight applied to the question.
description	Description of the question.
options	Detail information regarding question options.
id	Option ID
type	Type of question. **EXAMPLE** KPI and Additive
label	Indicates whether the label for the question is displayed.
isDefault	Indicates whether this is a default question.
kpiPriority	Indicates whether this is a priority question for determining KPI score.
order	order in which the question appears.
points	Number of points for the question.

JSON File Example

The following is an example of a formatted JSON response file.

```
[
        {
                "id": number,
                "name": string,
                "description": string,
                "creator": string,
                "created": epoch milliseconds,
                "manualWeighting": Boolean,
                "createdTz": "America\/Chicago",
                "updater": "2",
                "updated": 1489634636917,
```

```
        "updatedTz": "America\/Chicago",
        "status": 0,
        "defaultForm": false,
        "approvalRequired": true,
        "bandRanges": [{
              "band": 1,
              "begin": 0,
              "end": 40
        }, {
              "band": 2,
              "begin": 40,
              "end": 60
        }, {
              "band": 3,
              "begin": 60,
              "end": 100
        }
  ],
  "scoreType": {
        "id": 1,
        "name": "percentage"
  },
        "sections": [{
                    "id": 1,
                    "name": "section1",
                    "order": 0,
                    "weight": 1.0,
                    "startColor": "#ffffff",
                    "endColor": "#000000",
                    "questions": [{
                                "id": 1,
                                "text": "how was the service",
```

```
                "order": 0,
                "weight": 0.5,
                "description": "",
                "options": [{
                            "id": 1,
                            "type": "KPI",
                            "label": "Y",
                            "isDefault": true,
                            "kpiPriority": 0,
                            "order": 0,
                            "points": 100
                      }, {
                            "id": 2,
                            "type": "ADDITIVE",
                            "label": "N",
                            "isDefault": false,
                            "kpiPriority": 0,
                            "order": 1,
                            "points": 0
                      }, {
                            "id": 3,
                            "type": "NOTAPPLICABLE",
                            "label": "N\/A",
                            "isDefault": false,
                            "kpiPriority": 0,
                            "order": 2,
                            "points": 0
                      }
                ]
          },
          ....
]
```

```
                        }
                ]
        },
        ...
]
```

Evaluation Form ID API

Use this API to request information regarding a specific QM evaluation form from .

Protocol and URI

URL	`/api/rest/recording/evalform/<id>`
Method	GET
Content Type	multipart/form-data

Response Fields

Name	Description
id	Evaluation form ID.
name	Evaluation form name.
description	Description of the evaluation form .
creator	Evaluation form creator.
created	Time the evaluation form was created in UNIX epoch milliseconds.
manualWeighting	Indicates whether manual weighting applies to questions on the evaluation form.
createdTZ	Time zone in which the evaluation form was created.
updater	Updater indicator.
updated	Time indicator when the evaluation form was last updated, in UNIX epoch format.

Name	Description
updatedTz	Time zone in which the evaluation form was updated.
status	Status of the evaluation form.
defaultForm	Indicates whether the evaluation form shows default answer values.
approvalRequired	Indicates whether changes to the evaluation form must be approved.
bandRanges	Detail information regarding the evaluation form band ranges.
band	Evaluation form band number.
begin	Beginning of the evaluation form band.
end	Ending of the evaluation form band.
scoreType	Detail information the evaluation form score.
id	Score ID.
name	Name of the score type. **EXAMPLE** Percentage
sections	Detail information regarding the evaluation form sections.
id	Evaluation form section ID.
name	Evaluation form section name.
order	Order in which the section appears in the evaluation form.
weight	Evaluation form section weight factor.
startColor	Beginning color of the evaluation form section.
endColor	Ending color of the evaluation form section.
questions	Detail information regarding the evaluation form question.
id	Question ID.
text	Text of the question.

Name	Description
order	Order in which the question appears in the section.
weight	Weight applied to the question.
description	Description of the question.
options	Detail information regarding question options.
id	Option ID
type	Type of question. **EXAMPLE** KPI and Additive
label	Indicates whether the label for the question is displayed.
isDefault	Indicates whether this is a default question.
kpiPriority	Indicates whether this is a priority question for determining KPI score.
order	order in which the question appears.
points	Number of points for the question.

JSON File Example

The following is an example of a formatted JSON response file.

```
[
        {
                "id": number,
                "name": string,
                "description": string,
                "creator": string,
                "created": epoch milliseconds,
                "manualWeighting": Boolean,
                "createdTz": "America\/Chicago",
                "updater": "2",
                "updated": 1489634636917,
```

```
        "updatedTz": "America\/Chicago",
        "status": 0,
        "defaultForm": false,
        "approvalRequired": true,
        "bandRanges": [{
              "band": 1,
              "begin": 0,
              "end": 40
        }, {
              "band": 2,
              "begin": 40,
              "end": 60
        }, {
              "band": 3,
              "begin": 60,
              "end": 100
        }
  ],
  "scoreType": {
        "id": 1,
        "name": "percentage"
  },
        "sections": [{
                    "id": 1,
                    "name": "section1",
                    "order": 0,
                    "weight": 1.0,
                    "startColor": "#ffffff",
                    "endColor": "#000000",
                    "questions": [{
                                "id": 1,
                                "text": "how was the service",
```

```
                "order": 0,
                "weight": 0.5,
                "description": "",
                "options": [{
                        "id": 1,
                        "type": "KPI",
                        "label": "Y",
                        "isDefault": true,
                        "kpiPriority": 0,
                        "order": 0,
                        "points": 100
                    }, {
                        "id": 2,
                        "type": "ADDITIVE",
                        "label": "N",
                        "isDefault": false,
                        "kpiPriority": 0,
                        "order": 1,
                        "points": 0
                    }, {
                        "id": 3,
                        "type": "NOTAPPLICABLE",
                        "label": "N\/A",
                        "isDefault": false,
                        "kpiPriority": 0,
                        "order": 2,
                        "points": 0
                    }
                ]
            },
            ....
]
```

```
                        }
                ]
        },
        ...
]
```

Export API

This API exposes REST-like endpoints for performing exports. It allows you to retrieve data from back-end models in a CSV format.

Protocol and URI

URI	`/api/rest/fileexport/<importType>` where <importType> is one of the predefined import types. **EXAMPLE** Examples of importTypes are `persons`, `groups`, and `teams`.
Method	GET
Permissions	Bulk Import
Content Type	multipart/form-data

Request Fields

Field	Req?	Type	Description
acdId	N	String	If set, returns agents associated with this ACD ID.
basicInfo	N	Boolean	When set to True, returns only basic agent information, including ID, first name, last name, and employee number.

By default (that is, with no parameters set) all agents are returned, regardless of activation status.

NOTE All variants enforce view rules. Only agents visible to the caller are returned.

Response Fields

The JSON output includes fields defined in the following tables.

The fields for persons are defined in the table below.

Field	Description
User ID	The user's ID in Webex WFO.
Email	The user's email address.
First Name	The user's first name.
Last Name	The user's last name.
Display Time Zone	The user's display time zone.
Login	The user's login ID.
Employee ID	The user's employee ID.
Team	The user's assigned team.
Is Reconcile Only	
Activated	The date the user was activated.
Deactivated	The date the user was deactivated.
Locale	The language used by the user.
Roles	The roles assigned to the user.
Full Scope	When set to TRUE, the user has scope over all groups, teams, and users. When set to FALSE, the user's scope is as configured in the Scope Teams and Scope Groups fields.
Scope Groups	The groups within the user's scope.
Scope Teams	The teams within the user's scope.
QM Views	The views assigned to the user in QM.
ACD ID	The user's ID within the ACD.
ACD Server ID	The ID of the ACD that is the source of data for the user.
Enable Scheduling	Indicates if the user can be scheduled. True/False.

Field	Description
Views	The views that the user is assigned in WFM.
Main View	The user's main view in WFM.
Skill Mappings	The skill mappings assigned to this user.
Company Start Date	The date the user started with the company.
Department Start Date	The date the user started with the department.
Rank	The user's rank within the contact center.
Company End Date	The date the user left the company.
Work Condition Profile	The work condition profile assigned to this user.
Schedule Release Profile	The schedule release profile assigned to this user.
Scheduling Time Zone	The time zone used to schedule the agent.
Max Staffing Group	The maximum staffing group to which the user is assigned.

The fields for groups are defined in the table below.

Field	Description
Group ID	The group's ID in the ACD.
Name	The name of the group.
Activated Date	The date the group was activated.
Deactivated Date	The date the group was deactivated.

The fields for teams are defined in the table below.

Field	Description
Team ID	The team's ID in the ACD.
Name	The name of the group.

Field	Description
Activated Date	The date the group was activated.
Deactivated Date	The date the group was deactivated.
Parent Group ID	The group to which the team belongs.
Productivity Compilation	(False/True) Indicates if the team's statistics are to be included when the Capture service compiles productivity statistics.

CSV File Output Example

The following is an example of the exported CSV file containing the requested data. The headers of the CSV file are localized to match the user's selected locale.

```
Team ID,Name,Activated Date,Deactivated Date
3,Exported Team 1,8/17/2016,12/31/2999
5,Exported Team 2,8/17/206,12/31/2999
```

Generic Text Import API

The Generic Text Import API provides the ability to import data from a text-based source.

Protocol and URI

URI	`/api/rest/cas/importtext`
Method	POST, PUT
Permissions	Administrator Tenant
Content Type	JSON

JSON Request Fields

The JSON file includes the fields defined in the following table.

NOTE The columns in the request can be listed in any order. Column headers are required for each column you choose to use.

Field	Sub-field	Req?	Description
records	receiver	N	A list of strings designating the recipients. Delimited by pipes.
records	sender	N	A string designating the sender. This does not need to be an email address.
records		Y	An array of records, one for each contact to be imported.
records	id	N	An identifier from the source associated with the contact. This is not the CCR ID. The contact will not be imported if the ID has already been used for a previous contact.
records	metadata	N	Any metadata associated with the contact. The names must match defined metadata fields.

Field	Sub-field	Req?	Description
textType		Y	Identifies the source of the text. The value should be one of the following: ■ email ■ chat ■ website ■ other
records	time	Y	Specifies the start time associated with the contact. The format is UNIX epoch time in milliseconds.
records	text	Y	The body of the contact that contains the conversation.
evalForm	id name	N	The evaluation form assigned to contacts if the contact is determined to be marked for quality. You can either provide an ID or name. If an ID is not provided, the API looks up the evaluation form by name.
records	username	Y	The login of the user associated with the contact.
records	subject	N	The subject associated with the contact.
records	references	N	Typically used in emails to track threaded discussions. Delimited by pipes.

JSON File Example

```
{
        "textType": "email|chat|twitter|website|other",
        "evalForm": {
              "id": 5,
              "name": "name of evalForm. Upload will work with only name or
  only id. Only one evalForm can be assigned."
        },
        "records": [
              {
```

```
                    "time": 1439993633000,
                    "username": "The agent’s username. Used to identify the
    agent this contact is attributed to",
                    "sender": "The sender of the email or text, usually an
    email address. It will take any string though",
                    "receiver": [
                          "receiver 1",
                          "receiver 2"
                    ],
                    "text": "This is the body of the text and the portion
    that will get searched for hits",
                    "subject": "This is the Subject of the Email or Text"
                    "metadata": {
                          "field1" : "value1",
                          "field2" : "value2"
                    },
                    "references": [
                          "reference 1",
                          "reference 2"
                    ],
                    "id" : "An identifier for the text; only pertains to
    emails"
              }
        ]
}
```

Generic Text Import with Multiple Files API

The Generic Text Import with Multiple Files API provides the ability to import data from a text-based source with contacts and contents in separate files. This API can be used either independently or by the Data Server service.

NOTE When you import text data using a Webex WFO API, the text should be UTF-8 encoded. This allows the inclusion of special characters that are not available with ASCII encoding.

Protocol and URI

URI	`/api/upload/textcontacts/`
Method	POST
Permissions	Bulk Import
Content Type	multipart/form-data

Request Fields

The CSV and JSON files include fields defined in the following table.

NOTE If you are using the Data Server to upload files, the name of the CSV or JSON file must be TEXTCONTACT.

Field	Req?	Type	Description
evalForm	N	String	Evaluation form name. The evaluation form assigned to a contact if the contact is marked for quality.
filename	Y	String	Name of the text file containing the conversation.
id	N	String	An identifier from the source associated with the contact. This is not the CCR ID. The contactis not

Field	Req?	Type	Description
			imported if the ID has already been used for a previous contact.
metadata	N	String	Metadata associated with the contact. Metadata names must match defined metadata fields.
receiver	N	String	List of recipients. Each recipient is delimited by a pipe symbol (|).
references	N	String	Text used in emails to track threaded discussions. Delimited by pipes (if the file is in CSV format) or in a JSON array (if the file is in JSON format).
sender	N	String	Sender name. This can be, but does not need to be, an email address.
subject	N	String	Subject associated with the contact.
textType	Y	String	Source of the text. Valid values: ■ chat ■ email ■ facebook ■ other ■ website
time	Y	Time	Start time associated with the contact. Format = Unix epoch time in milliseconds.
username	Y	String	Login of the user associated with the contact.

Supported Formats

A multipart upload consists of one or more contact data files and one or more content files.

File	Supported Formats
Contact data—contains	CSV, JSON

File	Supported Formats
contact information	
Content—contains text portions of a contact	TXT

You can also upload a ZIP file that contains files in all of the above formats. When unzipped, the contents of the file are treated as if they were included in a standard multipart request. See for more information.

IMPORTANT CSV and JSON files can only be used for contact data, and ZIP files can only be used to compress the files for a multipart upload. Content files cannot use CSV, JSON, or ZIP file formats.

Batch CSV File Example

```
textType,filename,evalForm,time,username,sender,receiver,subject,metadata,refere
   nces,id
   email,email1.txt,1,1439993633000,agent1@cisco.com,cust1@abc.com,agent1@cisco.
   com,my account,accNum=123|ssn=234,ref1|ref2,1
   email,email2.txt,evalForm1,1439993633000,agent1@cisco.com,cust2@abc.com,agent
   1@cisco.com|agent2@cisco.com,close my account,accNum=676,ref1,2
   chat,chat1.txt,1,1439993633000,agent1@cisco.com,cust3@abc.com,agent1@cisco.co
   m,billing question,accNum=900,,
```

ZIP File Example

You handle a ZIP differently than a CSV or JSON file, in that a ZIP file is a collection of files that are processed as if they were individual files within the multipart request. Remember these points when using a ZIP file:

- The name of the ZIP file is the key that needs to be referenced in the CSV/JSON.
- Any folder structure in the ZIP file is flattened and ignored.

For example, a multipart request looks this.

```
textcontacts.zip
          textcontacts.csv (contains 2 rows, for email1 and chat1)
          email1.txt
          chat1.txt
```

This multipart request is processed as if the files were all in the ZIP or all individually in the multipart request.

JSON File Example

JSON Body

```
[
        {
              "textType": "email",
              "time": 1439993633000,
              "username": "john.smith@acme.com",
              "sender": "mysender",
              "filename": "email1.txt",
              "subject": "This is the Subject of the Email or Text",
        },
        {
              "textType": "chat",
              "evalForm": {
                    "id": 5,
                    "name": "name of evalForm. Upload will work with only
  name or only id. Only one evalForm can be assigned."
              },
              "time": 1439993633000,
              "username": "john.smith@acme.com",
              "sender": "mysender",
              "filename": "chat1.txt",
              "subject": "This is the Subject of the Email or Text"
              "receiver": [
                    "receiver 1",
                    "receiver 2"
              ],
              "metadata": {
```

```
                "field1" : "value1",
                "field2" : "value2"
        },
        "references": [
                "reference 1",
                "reference 2"
        ],
    }
]
```

Content

email1.txt

```
This is the email body.
```

Import API

This API exposes REST-like endpoints for performing imports. It allows you to retrieve information about the back-end object models (that is, the back-end model fields and the types that are assignable to those fields) and import data from CSV files into those back-end models.

GET Import Protocol and URI

URI	`/api/rest/fileimport/{importType}` where importType is one of the pre-defined import types.`users` **EXAMPLE** `users`, `groups`, and `teams`.
Method	GET
Permissions	Bulk Import
Content Type	multipart/form-data

GET Import Request Fields

Field	Req?	Type	Description
acdId	N	String	If set, returns agents associated with this ACD ID.
basicInfo	N	Boolean	When set to True, only returns basic agent information (ID, first name, last name, employee number).

By default (that is, with no parameters set), all agents are returned, regardless of activation status.

NOTE All variants enforce view rules. Only agents visible to the caller are returned.

GET Import Response Fields

The JSON output includes fields defined in the following table.

Field	Description
fields	List of fields on the back-end model.
key	Database key identifying the field in the back-end.
label	Field label used in the user interface.
required	Specifies whether the field is required.
multivalued	Specifies whether the field can accept multiple values.
types	List of data types the field can accept. Examples of types include text, dates, and Boolean data.
code	Code for the data type.
description	Description of the data type.
format	Example of the data type.

GET Import JSON File Example

The following is an example of a formatted JSON response.

```
{
        "fields": [
              {
                    "key": "id",
                    "label": "Team ID",
                    "required": true,
                    "multivalued": false,
                    "types": [
                    {
                          "code": "Integer",
                          "description": "Number",
                          "format": "1, 2, 3"
                    }
              ]
        },
```

```
            {
                  "key": "name",
                  "label": "Name",
                  "required": true,
                  "multivalued": false,
                  "types": [
                  {
                        "code": "Text",
                        "description": "Text",
                        "format": "Example text"
                  }
      ]
},
{
            "key": "activatedDate",
            "label": "Activated Date",
            "required": true,
            "multivalued": false,
            "types": [
                  {
                        "code": "MM/DD/YYYY",
                        "description": "US Date",
                        "format": "07/04/2016"
                  },
                  {
                        "code": "DD/MM/YYYY",
                        "description": "European Date",
                        "format": "04/07/2016"
                  }
            ]
},
```

```
{
          "key": "productivityCompilation",
          "label": "Productivity Compilation",
          "required": true,
          "multivalued": false,
          "types": [
                {
                      "code": "TrueFalse",
                      "description": "true/false",
                      "format": "true"
                            }
                      ]
                }
          ]
}
```

POST Import Protocol and URI

URI	`/api/rest/fileimport/{importType}` where importType is one of the pre-defined import types. **EXAMPLE** `users`, `groups`, and `teams`.
Method	POST
Permissions	Bulk Import
Content Type	multipart/form-data

Import POST Response Fields

The JSON output includes fields defined in the following table.

Field	Description
tenantId	Tenant ID for multi-valued fields.

Field	Description
mapping	Describes the array of mapping objects for multi-valued fields.
column	CSV column from which data for mapping is retrieved.
field	Field of the back-end model to which the data will be mapped.
dataType	Type of data of the CSV column, which specifies for the back-end whether additional transformations are required for the data.

POST Import JSON File Example

The following is an example of a formatted JSON response.

JSON Format

```
{
        "tenantId":1,
        "mapping":[
              {
                    "column":"Team ID",
                    "field":"id",
                    "dataType": "Integer"
              },
              {
                    "column":"Name",
                    "field":"name",
                    "dataType": "Text"
              },
              {
                    "column":"Activated Date",
                    "field":"activatedDate",
                    "dataType": "MM/DD/YYYY"
              },
              {
```

```
                        "column":"Productivity Compilation",
                        "field":"productivityCompilation",
                        "dataType": "TrueFalse"
                }
        ]
}
```

POST Success Response Format

```
{
        "status": "SUCCESS",
        "responseText": "The file was successfully imported"
}
```

Import Preference APIs

Import preferences are user-defined data that are used as part of the CSV file import process. This data reduces the effort of importing files by allowing you to maintain your predefined preferences for CSV file imports.

This section describes the following import preference APIs:

Create Import Preference API

The Create Preferences API allows you to create a new import preference.

Create Import Preference Protocol and URI

URI	`/api/rest/importpreference`
Method	POST
Permissions	Bulk Import
Content Type	multipart/form-data

Create Import Preference Request Fields

Field	Description
name	Name of the new import preference.
type	Type of import preference.
mappings	List of field mappings for the import preference.
field	Field for the specified import preference.
column	Column heading for the field.
dataType	Type of data in the field.

Create Import Preference JSON File Example

The following is an example of a formatted JSON request.

```
{
	"name": "My Preference",
	"type": "persons",
	"mapping": [
		{
			"field": "personId",
			"column": "personId",
			"dataType": "Text"
		}
	]
}
```

Create Import Preference System Response

The system responds to a successful import preference deletion with a 200 OK Status.

Delete Import Preference API

The Delete Preference API allows you to delete an existing import preference.

Delete Import Preference Protocol and URI

URI	`/api/rest/importpreference/{id}` where id is the ID of the import preference you want to delete.
Method	DELETE
Permissions	Bulk Import
Content Type	multipart/form-data

Delete Import Preference System Response

The system responds to a successful import preference deletion with a 200 OK Status.

Import Preferences API

The Import Preferences API retrieves all the import preferences for the current tenant.

Import Preferences Protocol and URI

URI	`/api/rest/importpreference`
Method	GET
Permissions	Bulk Import
Content Type	multipart/form-data

Import Preferences Response Fields

The JSON output includes fields defined in the following table.

Field	Description
preferences	List of import preferences.
persons	List of user import preferences.
id	Import preference ID.
name	Import preference name.
type	Type of import preference.
mappings	List of field mappings for the import preference.
field	Field for the specified import preference.
column	Column heading for the field.
dataType	Type of data in the field.

JSON File Example

The following is an example of a formatted JSON response.

```
{
        "preferences": [
              {
```

```
            "persons": [
                  {
                        "id": 1,
                        "name": "My User Mapping",
                        "type": "persons",
                        "mappings": [
                              {
                                    "field": "personId",
                                    "column": "Person ID",
                                    "dataType": "Integer"
                              },
                              {
                                    "field": "email",
                                    "column": "Email",
                                    "dataType": "Text"
                              }
                        ]
                  },
                  {
                        "id": 2,
                        "name": "My Empty Mapping",
                        "type": "persons",
                        "mappings": []
                  }
            ]
      },
      {
            "groups": [
                  {
                        "id": 3,
                        "name": "My Group Mapping",
                        "type": "persons",
```

```
                                "mappings": []
                            }
                        ]
                    }
            ]
}
```

System Response

The system responds to a successful import preference deletion with a 200 OK Status.

Import Preferences by ID API

The Import Preferences by ID API retrieves an import preference by its ID, so you can view it.

Import Preferences by ID Protocol and URI

URI	`/api/rest/importpreference/{id}`
Method	GET
Permissions	Bulk Import
Content Type	multipart/form-data
Parameters	id = ID of the import preference you want to view.

Import Preferences by ID Response Fields

The JSON output includes fields defined in the following table.

Field	Description
id	Import preference ID.
name	Import preference name.
type	Type of import preference.
mappings	List of field mappings for the import preference.

Field	Description
field	Field for the specified import preference.
column	Column heading for the field.
dataType	Type of data in the field.

Import Preferences by ID JSON File Example

The following is an example of a formatted JSON response.

```
{
          "id": 1,
          "name": "My User Mapping",
          "type": "persons",
          "mappings": [
                {
                      "field": "personId",
                      "column": "Person ID",
                      "dataType": "Integer"
                },
                {
                      "field": "email",
                      "column": "Email",
                      "dataType": "Text"
                }
          ]
}
```

System Response

The system responds to a successful import preference deletion with a 200 OK Status.

Update Import Preference API

The Update Preferences API allows you to edit an existing import preference.

Update Import Preference Protocol and URI

URI	`/api/rest/importpreference/{id}` where id is the ID of the import preference you want to edit.
Method	PUT
Permissions	Bulk Import
Content Type	multipart/form-data

Update Import Preference Response Fields

Field	Description
name	Name of the import preference.
type	Type of import preference.
mappings	List of field mappings for the import preference.
field	Field for the specified import preference.
column	Column heading for the field.
dataType	Type of data in the field.

Update Import Preference JSON File Example

The following is an example of a formatted JSON request.

```
{
          "name": "Updated Name",
          "type": "persons",
          "mapping": [
                {
                      "field": "personId",
                      "column": "Person ID",
                      "dataType": "Text"
                }
          ]
```

```
}
```

Update Import Preference System Response

The system responds to a successful import preference deletion with a 200 OK Status.

Inclusion/Exclusion List API

Use the Inclusion/Exclusion List API to create, define, update, or delete list of extensions, phone numbers, or contact directions you either do or do not want to record. This API uses four methods:

- GET InclusionExclusion: Allows you to retrieve one or all Inclusion/Exclusion lists.
- PUT InclusionExclusion: Allows you to update information in the Inclusion/Exclusion list.
- POST InclusionExclusion: Allows you to create an Inclusion/Exclusion list.
- DELETE InclusionExclusion: Allows you to delete an Include/Exclude list.

GET Inclusion/Exclusion List API

GET Include/Exclude List allows you to retrieve one or all Include/Exclude lists.

GET Inclusion/Exclusion List API Protocol and URI

URI	`/api/rest/recording/inclusionExclusion` `/api/rest/recording/inclusionExclusion/<id>`
Method	GET
Permissions	Administer QM
Content Type	application/JSON
Parameters	<id> = Inclusion/Exclusion list ID

GET Inclusion/Exclusion List API Request Fields

Field	Req?	Type	Description
inclusionExclusionListId	Y	Integer	Inclusion/Exclusion List ID.
inclusionExclusionType	Y	Integer	Inclusion/Exclusion type:

Field	Req?	Type	Description
			1 = Inclusion 2 = Exclusion
inclusionExclusionPatternType	N	Integer	Pattern type of the Inclusion/Exclusion list" 1 = ANI (Calling number) 2 = DNIS (Called number) 3 = Extension 4 = Phone number 5 = Any
pattern	N	String	Pattern of the numbers in the Inclusion/Exclusion list. This allows you to specify a range of numbers to include or exclude. The pattern field allows two wildcard characters, and the plus (+) sign: ? = a single digit **EXAMPLE** 612822???? includes or excludes all numbers in the 612 area code with the 822 exchange. * = Any number of digits **EXAMPLE** 612* includes or excludes all numbers in the 612 area code. + = Country code indicator (only valid at the start of the string) **EXAMPLE** +44 includes or excludes contacts in the United Kingdom. **IMPORTANT** If you export an inclusion or exclusion list that includes an international calling code with the plus (+) sign, Excel will treat the number as a mathmatical

Field	Req?	Type	Description
			expression. To use these lists, convert the exported .CSV file to text (.TXT), open the file in Excel, and define this as a text formatted column.
inclusionExclusionCallDirection	N	Integer	ID for the direction of the call: 1 = Inbound 2 = Outbound 3 = Either

GET Service API Response Fields

The JSON output includes fields defined in the following table.

Field	Description
inclusionExclusionListId	Inclusion/Exclusion List ID.
inclusionExclusionType	Inclusion/Exclusion type: 1 = Inclusion 2 = Exclusion
inclusionExclusionPatternType	Pattern type of the Inclusion/Exclusion list" 1 = ANI (Calling number) 2 = DNIS (Called number) 3 = Extension 4 = Phone number 5 = Any
pattern	Pattern of the numbers in the Inclusion/Exclusion list. This allows you to specify a range of numbers to include or exclude. The pattern field allows two wildcard characters, and the plus (+) sign: ? = a single digit **EXAMPLE** 612822???? includes or excludes all numbers in the 612 area code with the 822 exchange.

Field	Description
	* = Any number of digits **EXAMPLE** 612* includes or excludes all numbers in the 612 area code. + = Country code indicator (only valid at the start of the string) **EXAMPLE** +44 includes or excludes contacts in the United Kingdom. **IMPORTANT** If you export an inclusion or exclusion list that includes an international calling code with the plus (+) sign, Excel will treat the number as a mathmatical expression. To use these lists, convert the exported .CSV file to text (.TXT), open the file in Excel, and define this as a text formatted column.
inclusionExclusionCallDirection	ID for the direction of the call: 1 = Inbound 2 = Outbound 3 = Either

GET Service API JSON File Example

```
{
   "inclusionExclusionListId" : <number>,
   "inclusionExclusionType" : <number>,
   "inclusionExclusionPatternType" : <number>,
   "pattern" : <string>,
   "inclusionExclusionCallDirection" : <number>,
}
```

PUT Include/Exclude List API

PUT Include/Exclude List allows you to update existing Inclusion/Exclusion lists.

PUT Include/Exclude List API Protocol and URI

URI	`/api/rest/recording/inclusionExclusion/<id>`
Method	PUT
Permissions	Administer QM
Content Type	application/JSON

PUT Include/Exclude List API Request Fields

Field	Req?	Type	Description
inclusionExclusionListId	Y	Integer	Inclusion/Exclusion List ID.
inclusionExclusionType	Y	Integer	Inclusion/Exclusion type: 1 = Inclusion 2 = Exclusion
inclusionExclusionPatternType	N	Integer	Pattern type of the Inclusion/Exclusion list: 1 = ANI (Calling number) 2 = DNIS (Called number) 3 = Extension 4 = Phone number 5 = Any
pattern	N	String	Pattern of the numbers in the Inclusion/Exclusion list. This allows you to specify a range of numbers to include or exclude. The pattern field allows two wildcard characters, and the plus (+) sign: ? = a single digit **EXAMPLE** 612822???? includes or excludes all numbers in the 612 area code with the 822 exchange. * = Any number of digits

Field	Req?	Type	Description
			EXAMPLE 612* includes or excludes all numbers in the 612 area code. + = Country code indicator (only valid at the start of the string) **EXAMPLE** +44 includes or excludes contacts in the United Kingdom. **IMPORTANT** If you export an inclusion or exclusion list that includes an international calling code with the plus (+) sign, Excel will treat the number as a mathematical expression. To use these lists, convert the exported .CSV file to text (.TXT), open the file in Excel, and define this as a text formatted column.
inclusionExclusionCallDirection	N	Integer	ID for the direction of the call: 1 = Inbound 2 = Outbound 3 = Either

PUT Include/Exclude List API Response Fields

The JSON output includes fields defined in the following table.

Field	Description
inclusionExclusionListId	Inclusion/Exclusion List ID.
inclusionExclusionType	Inclusion/Exclusion type: 1 = Inclusion 2 = Exclusion
inclusionExclusionPatternType	Pattern type of the Inclusion/Exclusion list"

Field	Description
	1 = ANI (Calling number) 2 = DNIS (Called number) 3 = Extension 4 = Phone number 5 = Any
pattern	Pattern of the numbers in the Inclusion/Exclusion list. This allows you to specify a range of numbers to include or exclude. The pattern field allows two wildcard characters, and the plus (+) sign: ? = a single digit **EXAMPLE** 612822???? includes or excludes all numbers in the 612 area code with the 822 exchange. * = Any number of digits **EXAMPLE** 612* includes or excludes all numbers in the 612 area code. + = Country code indicator (only valid at the start of the string) **EXAMPLE** +44 includes or excludes contacts in the United Kingdom. **IMPORTANT** If you export an inclusion or exclusion list that includes an international calling code with the plus (+) sign, Excel will treat the number as a mathmatical expression. To use these lists, convert the exported .CSV file to text (.TXT), open the file in Excel, and define this as a text formatted column.
inclusionExclusionCallDirection	ID for the direction of the call: 1 = Inbound 2 = Outbound 3 = Either

PUT Include/Exclude List API JSON File Example

{

```
    "inclusionExclusionListId" : <number>,
    "inclusionExclusionType" : <number>,
    "inclusionExclusionPatternType" : <number>,
    "pattern" : <string>,
    "inclusionExclusionCallDirection" : <number>,
}
```

POST Inclusion/Exclusion List API

POST Inclusion/Exclusion List allows you to create a list of contacts to include in or exclude from recording.

POST Inclusion/Exclusion List API Protocol and URI

URI	`/api/rest/recording/inclusionExclusion` `/api/rest/recording/inclusionExclusion/<id>`
Method	POST
Permissions	Administer QM
Content Type	application/JSON

POST Inclusion/Exclusion List API Request Fields

Field	Req?	Type	Description
inclusionExclusionListId	Y	Integer	Inclusion/Exclusion List ID.
inclusionExclusionType	Y	Integer	Inclusion/Exclusion type: 1 = Inclusion 2 = Exclusion
inclusionExclusionPatternType	N	Integer	Pattern type of the Inclusion/Exclusion list" 1 = ANI (Calling number) 2 = DNIS (Called number)

Field	Req?	Type	Description
			3 = Extension 4 = Phone number 5 = Any
pattern	N	String	Pattern of the numbers in the Inclusion/Exclusion list. This allows you to specify a range of numbers to include or exclude. The pattern field allows two wildcard characters, and the plus (+) sign: ? = a single digit **EXAMPLE** 612822???? includes or excludes all numbers in the 612 area code with the 822 exchange. * = Any number of digits **EXAMPLE** 612* includes or excludes all numbers in the 612 area code. + = Country code indicator (only valid at the start of the string) **EXAMPLE** +44 includes or excludes contacts in the United Kingdom. **IMPORTANT** If you export an inclusion or exclusion list that includes an international calling code with the plus (+) sign, Excel will treat the number as a mathmatical expression. To use these lists, convert the exported .CSV file to text (.TXT), open the file in Excel, and define this as a text formatted column.
inclusionExclusionCallDirection	N	Integer	ID for the direction of the call:

Field	Req?	Type	Description
			1 = Inbound 2 = Outbound 3 = Either

POST Inclusion/Exclusion List API Response Fields

The JSON output includes fields defined in the following table.

Field	Description
inclusionExclusionListId	Inclusion/Exclusion List ID.
inclusionExclusionType	Inclusion/Exclusion type: 1 = Inclusion 2 = Exclusion
inclusionExclusionPatternType	Pattern type of the inclusion/exclusion list: 1 = ANI (Calling number) 2 = DNIS (Called number) 3 = Extension 4 = Phone number 5 = Any
pattern	Pattern of the numbers in the Inclusion/Exclusion list. This allows you to specify a range of numbers to include or exclude. The pattern field allows two wildcard charaters: ? = a single digit **EXAMPLE** 612822???? includes or excludes all numbers in the 612 area code with the 822 exchange. * = Any number of digits **EXAMPLE** 612* includes or excludes all numbers in the 612 area code.
inclusionExclusionCallDirection	ID for the direction of the call: 1 = Inbound

Field	Description
	2 = Outbound 3 = Either

POST Inclusion/Exclusion List API JSON File Example

```
{
   "inclusionExclusionListId" : <number>,
   "inclusionExclusionType" : <number>,
   "inclusionExclusionPatternType" : <number>,
   "pattern" : <string>,
   "inclusionExclusionCallDirection" : <number>,
}
```

DELETE Include/Exclude List API

DELETE Include/Exclude List allows you to delete an Include/Exclude List.

DELETE Include/Exclude List API Protocol and URI

URI	`/api/rest/recording/inclusionExclusion/<id>`
Method	POST
Permissions	Administer QM
Content Type	application/JSON

Delete Include/Exclude List System Response

The system responds to a successful import preference deletion with a 200 OK Status.

Interaction Summary Read API

Use the Interaction Summary Read API to export interaction summaries.

Protocol and URI

URI	`/api/rest/cas/interactionsummary?ClientAppID="<string>"&ccrId=<Integer>`
Method	GET
Permissions	View Speech to Text Analytics
Parameters	ClientAppId = A string representing the application that called the API. It is assigned once an app is approved to invoke the API. ccrID = The ID for the CCR that the summary is associated with.

Response example

```
{
    "status": "success",
    "summary": "Jack called his mobile carrier to inquire about upgrading his data
plan as he was going through his data too quickly. The customer service agent
verified Jack's identity by confirming his name, account number, date of birth, and
phone number. The agent then looked up Jack's account and saw he was using a lot of
data. The agent mentioned a 60GB plan for $75 but Jack said he would need more. The
agent then offered a 120GB plan for $90 which also included unlimited calls within
North America. Jack asked if overseas calls would be included, and the agent said
different countries have different rates. Jack decided to upgrade to the 120GB plan
and the agent confirmed the change. Jack was satisfied and thanked the agent before
completing a customer satisfaction survey.",
    "Reason": "Success"
}
```

Jobs API

This API facilitates simple queries of the last “X” recording reconciliation jobs to provide details about in-flight job status as well as historical reconciliation job details.

Protocol and URI

URI	`/api/rest/reconciliation/jobs`
Method	GET
Permissions	Administer Tenant
Content Type	Not applicable

Request fields

Name	Description	Notes
limit	The number of jobs to return	Default = 1
tenantId	The tenant ID to query for reconciliation jobs	This field only applies to system administrator accounts. For tenant administrator users, the field defaults to the user’s tenant ID.

Response Fields

Field	Type	Description
id	Numeric	Job ID.
tenantId	Numeric	Tenant ID for job.
jobStartDate	Epoch	Job start date.
jobEndDate	Epoch	Job end date.

Field	Type	Description
jobStatus	String	Job status. Possible Values: ■ SCHEDULED ■ IN_QUEUE ■ IN_PROGRESS ■ ABORTED ■ FINISHED
minContactDate	Epoch	Minimum contact date to search for.
maxContactDate	Epoch	Maximum contact date to search for.
recordingsProcessed	Numeric	Number of recordings processed.
recordingsToProcess	Numeric	Number of recordings to process.
cdrMatches	Numeric	Number of matched CDRs.
rootResultingInCcr	Numeric	Number of root recordings that resulted in contact creation.
ccrsCreated	Numeric	Number of CCRs created.
extensionTableMatches	Numeric	Number of devices found to match contacts in this reconciliation job.
acdMatches	Numeric	Number of ACD matches.
averageSuccess	Numeric (milliseconds)	Average completion time for each successfully reconciled root recording.
successCount	Numeric	Number of successfully processed root recordings.
successTotalTime	Numeric (milliseconds)	Total time it took to process all root recordings found to have matching contacts.
isScheduledJob	True/False	Job was triggered by a schedule.
updatedDate	Epoch	Timestamp of the most recent update to the job.

Field	Type	Description
producerHostName	String	Hostname of machine that created the job.
consumerHostName	String	Hostname of machine that consumed the job.
contactReconciliationDuration	Numeric (seconds)	Duration of time taken to reconcile contacts.
realtimeEventDuration	Numeric (seconds)	Duration of realtime events.
nonRealtimeEventDuration	Numeric (seconds)	Duration of non-realtime events.
workflowDuration	Numeric (seconds)	Duration of workflow.
numberOfRealtimeEvents	Numeric	Number of real-time events processed by job.
numberOfNonRealtimeEvents	Numeric	Number of non real-time events processed by job.
properties	Key-value map of Strings	Map of properties supplied to job.

JSON File Examples

Request

The following is an example of a formatted JSON request.

```
curl https://<your-server>/api/rest/reconciliation/jobs?limit=1
```

Response

The following is an example of a formatted JSON response.

```
[
    {
        "id": 10250,
        "tenantId": 1,
```

```
        "jobStartDate": 1628177990210,
        "jobEndDate": null,
        "jobStatus": "FINISHED",
        "minContactDate": 1628177390197,
        "maxContactDate": 1628149190193,
        "recordingsProcessed": 0,
        "recordingsToProcess": 0,
        "cdrMatches": 0,
        "rootResultingInCcr": 0,
        "ccrsCreated": 0,
        "extensionTableMatches": 0,
        "acdMatches": 0,
        "averageSuccess": 0,
        "successCount": 0,
        "successTotalTime": 0,
        "isScheduledJob": false,
        "updatedDate": 1628177990310,
        "producerHostName": "rdpod75Rahm1@10.192.103.47",
        "consumerHostName": "rdpod75Rahm1@10.192.103.47",
        "contactReconciliationDuration": 0,
        "realtimeEventDuration": 0,
        "nonRealtimeEventDuration": 0,
        "workflowDuration": 0,
        "numberOfRealtimeEvents": 0,
        "numberOfNonRealtimeEvents": 0,
        "properties": {}
    }
]
```

Organizational Structure APIs

The Organizational Structure APIs are used to retrieve or create basic information regarding a tenant.

Assign Roles to a Person API

The Assign Roles to a Person API allows you retrieve or update a person's role.

Assign Roles to a Person API Protocol and URI

URI	`/api/rest/org/common/<group>`
Method	POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data

Assign Roles to a Person API Request Fields

Name	Req?	Description
name	Y	Name of the group
parentGroupId	Y	Unique ID of the parent group.
teams	N	list of team IDs associated with the team.

Assign Roles to a Person API JSON File Example

The following is an example of a formatted JSON response.

```
[
        {
              "id":4,
              "name":"tenantAdmin",
```

```
                "groups":,
                {
                "id":2,
                "name":"tenant CCS 1",
          },
          ...
]
```

Common Group API

This API creates a new group with the specified data.

Common Group API Protocol and URI

URI	/api/rest/org/common/group>
Method	POST
Permissions	ViewOrg UpdateOrg
Content Type	JSON

Common Group API Request Fields

Name	Req?	Description
name	Y	Name of the group
parentGroupId	Y	Unique ID of the parent group.
teams	N	list of team IDs associated with the team.

Organizational Structure Common API (Drill-down Information)

This API returns a JSON array containing information for each group, all the teams in each group, and all the agents in each team, for the scope of the user.

Common API Protocol and URI

URI	`/api/rest/org/common/common/permission/<permission>`
Method	GET
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	permission(s)

Common API Request Fields

Name	Req?	Description
permission	Y	Permission name for which you want to list group and team permission information.

Common API Response Fields

The JSON output includes fields defined in the following table.

Name	Description
groupId	Group ID for the teams in the array.
name	Name of the group.
displayId	Display ID of the group.
parentGroupId	Id of the specified group’s parent group.
parentGroupName	Name of the specified group’s parent group.
teams	Team information.
personId	Person ID of the user in the specified team.
firstName	First name of the user in the specified team.
lastName	Last name of the user in the specified team.

Name	Description
tenantId	Tenant ID of the user in the specified team.
email	Email address of the user in the specified team.
skillId	Skill ID of the user in the specified team.

Common API JSON File Example

The following is an example of a formatted JSON response.


```
[
        {
              "groupId":3,
              "name":"Tenant 1 Group 1",
              "displayId":3,
              "parentGroupId":2,
              "parentGroupName":"Tenant 1",
              "teams":
                    [
                          {
                          "groupId":4,
                          "name":"Tenant 1 Team 1"
                          "displayId":4,
                          "parentGroupId":3,
                          "parentGroupName":"Tenant 1 Group 1",
                          "agents":
                                [
                                      {
                                      "personId":7,
                                      "firstName":"al"
                                      "lastName":"dente",
                                      "groupId":4,
                                      "tenantId":2,
```

```
                                    "email":"al.dente",
                                    "skillId":"6"
                                    }
                                ]
                                [
                                    {
                                    "personId":8,
                                    "firstName":"max"
                                    "lastName":"stout",
                                    "groupId":4,
                                    "tenantId":2,
                                    "email":"max.stout",
                                    "skillId":"6"
                                    }
                                ]
                                ...
                            },
                            ...
                        }
                    }
                }
            }
]
```

Group API

Use this API to retrieve all groups to which the user has access, or to create a new group.

Group API Protocol and URI

For listing or creating all groups to which the user has access:

URI	/api/rest/org/common/group/permission/<permission>
Method	GET POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data

For listing or creating a specific group:

URI	`/api/rest/org/group/<id>`
Method	GET POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Perameter	The group ID to which the user has access.

Group API Request Fields–All Groups

The creates new groups to which the user has access.

Name	Req?	Description
id	Y	Group ID to of the group to be displayed.
name	Y	Name of the group.
activated	Y	Activation time in epoch milliseconds.
deactivated	N	Deactivation time in epoch milliseconds.
ParentGroupId	Y	Parent group name for the group.

Group API Response Fields

The JSON output includes fields defined in the following table.

Name	Description
id	Group ID to of the group to be displayed.
name	Name of the group.

Group API JSON File Example

The following is an example of a formatted JSON response.

```
[
          {
                "id":4,
                "name":"Acme Rockets",
          },
          {
                "id":5,
                "name":"Associated Widgets",
          },
          ...
]
```

Details for a single group:

```
[
   {
          {
                "id":4,
                "name":"Acme Rockets",
                "activated",1395205200000,
                "deavtivated",32503615200000
          }
          ...
   }
]
```

Group by ID (with Teams)

The Group by ID (with teams) API allows you to list information for the specified group (including its teams), update the specified group, or delete the team.

Group by ID (with Teams) Protocol and URI

URI	`/api/rest/org/common/group/<id>`
Method	GET POST DELETE
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	Group ID

Group by ID (with Teams) Request Fields

The POST method updates the specified group record.

Name	Req?	Description
name	Y	Name of the group to be updated.
parentGroupId	Y	Parent group name for the new group.

Group by ID (with Teams) Response Fields

The JSON output includes fields defined in the following table.

Name	Description
id	Group ID to display.
name	Name of the group to display.
activated	Time since group activation, in epoch milliseconds.

Name	Description
deactivated	Time since group deactivation, in epoch milliseconds.
parentGroupId	Parent group name for the group.
teams	Detailed information on the teams within the group.
name	Team name.
groupId	Group ID for the team.

Group by ID (with Teams) JSON File Example

The following is an example of a formatted JSON response.

```
[
        {
              "id":3,
              "name":"Tenant 1 Group 1",
              "activated":1388556000000,
              "deactivated":32503615200000,
              "parentGroupId":2,
              "teams":
              [
                    {
                          "name":"Tenant 1 Team 1"
                          "groupId":4
                    }
                    {
                          "name":"Tenant 1 Team 2"
                          "groupId":5
                    }

                    ...
              ]
        },
```

```
        ...
]
```

Group-Person API

The Group-Person API allows you to view, reassign, and add a person to a group.

Group-Person API Protocol and URI

URI	`/api/rest/org/group/<groupId>/person/`
Method	GET PUT POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	groupId - ID of the group to which the persons

Group-Person API Request Fields

Reassign a person to a group, and returns all users in that group after the reassign.

Name	Req?	Description
personId	Y	Person ID of the user in the group.

Group-Person API Response Fields

The JSON output includes fields defined in the following table.

Name	Description
firstName	First name of the person in the group.
lastName	Last name of the person in the group.
personId	Person ID of the user in the group.

Group-Person API JSON File Example

The following is an example of a formatted JSON response.

```
[
        {
              "firstName":"al",
              "lastName":"dente",
              "personId":2,
        },
        {
              "firstName":"max",
              "lastName":"stout",
              "personId":3,
        },
        ...
]
```

Group (with Scope) API

The Group (with scope) API allows you to view and create a group.

Group (with Scope) API Protocol and URI

URI	`/api/rest/org/common/group/permission/<permission>`
Method	GET POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	permission - Name of the group's scope.

Group (with Scope) API Request Fields

The POST method creates a group in the specified parent group.

Name	Req?	Description
name	Y	Name of the group.
parentGroupId	Y	ID of the parent group.

Group (with Scope) API Response Fields

The JSON output includes fields defined in the following table.

Name	Description
name	Name of the group.
id	ID of the group.

Group (with Scope) API JSON File Example

The following is an example of a formatted JSON response.

```
[
   {
          "name":"Tenant 1 Group 1",
          "groupId":1,
   },
   {
          "name":"Tenanr 1 Group 2",
          "groupId":2,
   }
   ...
]
```

Permissions API

Use the Permissions API to retrieve all available permissions for a group.

Permissions API Protocol and URI

For listing or creating all groups to which the user has access:

URI	/api/rest/org/permissions/
Method	GET
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data

Permissions API Response Fields

The JSON output includes fields defined in the following table.

Name	Description
id	ID of the specific permission.
name	Name of the permission.

Permissions API JSON File Example

The following is an example of a formatted JSON response.

```
[
        {
              "id":4,
              "name":"ExportRecording",
        },
        {
              "id":5,
              "name":"AlertOnApprovalRequired",
        },
        ...
]
```

Person API

The Person API displays or creates a new person (user) within Webex WFO.

Person API Protocol and URI

URI	`/api/rest/org/common/person`
Method	POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data

Person API Request Fields

The POST method creates a new person/agent. Roles are not required for the new person; they will default to default agent role. The groupId is not required; it will default to default team, and person detail data is only required if the user is to be scheduled. Views is not required; the default view will be used..

Name	Req?	Description
firstName	Y	First name of the new person.
lastName	Y	Last name of the new person.
email	Y	New person's email address.
timeZone	Y	New person's time zone.
acdId	Y	ACD of the new person.
enabledForScheduled	Y	Follows the checkbox shown on the user edit/create page. Used by WFM instead of agent detail to determine whether to schedule the new person.
agentDetail		Provides detail for the new person
skillMappings	N	Lists the user's skill mappings.
exceptions	N	Lists the user's exceptions.
views	N	Defines the user's view settings.
viewId	N	User's view ID.
mainView	N	Indicates whether the view ID specified is the user's main

Name	Req?	Description
		view. True = User's main view False = Not user's main view
workShifts	N	User's work shift settings.
workShiftId	N	User's work shift ID.
shiftWeekStart	N	Date the user's shift week starts.

Person API JSON File Example

The following is an example of a formatted JSON response.

```
{
   "firstName": "Al",
   "lastName": "Dente",
   "email": "al.dente@domain.com",
   "password": "123456789",
   "timeZone": 1,
   "acdId": "999",
   "enabledForScheduling":true,
   "agentDetail": {
         "skillMappings": [
               1,
               2
         ],
         "exceptions": [
               1,
               2
         ],
         "views": [
               {
                     "viewId": 1,
```

```
                        "mainView": true
                    }
                ],
            "workShifts": [
                    {
                            "workShiftId": 1,
                            "shiftWeekStart": "2014-08-01"
                    }
            ]
}
```

Person by ID API

The Person by ID API. allows you to view or update detailed information about a specific user. It can also be used to delete a specific user.

Person by ID API Protocol and URI

URI	`/api/rest/org/common/person/<personId>`
Method	GET PUT
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	personId

Person by ID API Request Fields (PUT Method)

The PUT method creates the specified user's detailed information.

IMPORTANT If the user you create using the PUT method is included in your ACD, the next time the ACD is synced, Webex WFO creates a duplicate person profile. You will need to use the Merge Person function to merge these person profiles.

Name	Req?	Description
id	Y	Person ID of the user.
acdId	Y	ID of the user's ACD.
acdServerId	Y	The ACD's server ID, as a string. This can be the ID configured on the ACD Configuration page in Webex WFO, or one of the following: ■ For systems that use the Unified CCE, this is requirted, and is the peripheral ID. For systems that use an Avaya CM/CMS, this is required, and is the CMS ACD ID.
firstName	Y	User's first name.
lastName	Y	User's last name.
email	Y	User's email address.
displayId	Y	User's name as displayed in Webex WFO.
activated	Y	Date person was activated.
deactivated	N	Date person was deactivated
groupId	Y	ID of the user's group.
timeZone	Y	Time zone of the user's contact center.
enabledForScheduling	N	If set to True, Webex WFO uses this field rather than Agent Detail to determine whether to schedule the user.
agentDetails	N	List of details for the user.
active	N	Indicates whether the user is active.
companyStartDate	N	Date the user started working for the company.

Name	Req?	Description
departmentStartDate	N	Date the user started working in the department.
terminationDate	N	Last date the user worked for the company.
agentRank	N	User’s rank in the department.
skillMappings	N	Lists the user’s skill mappings.
id	N	Skill mapping ID.
name	N	Skill mapping name.
workShifts	N	User’s work shift settings.
workShiftId	N	User’s work shift ID.
workShiftName	N	Name of the user’s work shift.
shiftWeekStart	N	Date the user’s shift week starts.
exceptions	N	Lists the user’s exceptions.
id	N	ID of the user’s exception.
name	N	Name of the user’s exception.
date	N	Date of the user’s exception.
start	N	Start date of the user’s exception.
end	N	End date of the user’s exception.
durationHours	N	Number of hours for the user’s exception.
entireDay	N	Indicates whether the exception is for an entire day: True = Entire day exception False = less than an entire day exception
paid	N	Indicates whether the exception is paid. True = Paid exception

Name	Req?	Description
		False = Unpaid exception
views	N	Defines the user's view settings.
viewId	N	User's view ID.
name	N	User's view name.
mainView	N	Indicates whether the view ID specified is the user's main view. True = User's main view False = Not user's main view

Person by ID API Response Fields

The JSON output includes fields defined in the following table.

Field	Description
firstName	User's first name.
lastName	User's last name.
email	User's email address.
timeZone	Time zone of the user's contact center.
acdId	User[s ACD ID.
groupId	User's groupID
enabledForScheduling	If set to True, Webex WFO uses this field rather than Agent Detail to determine whether to schedule the user.
AgentDetail	Lists detailed information for the user.
skillMappings	Lists the user's skill mappings.
exceptions	Lists the user's exceptions.
views	Defines the user's view settings.

Field	Description
viewId	Defines the user's view ID.
mainView	Sets whether the view ID specified is the user's main view.
workShifts	Defines the user's work shift settings.
workShiftId	Defines the user's work shift ID.
shiftWeekStart	Sets the date of the user's shift week.

Person by ID API JSON File Example

The following is an example of a formatted JSON response.


```
{
   "firstName": "Al",
   "lastName": "Dente",
   "email": "al.dente12333333335555@yourdomain.com",
   "timeZone": 1,
   "groupId": "6",
   "enabledForScheduling": true,
   "agentDetail": {
         "skillMappings": [
               1,
               2
         ],
         "exceptions": [
               1,
               2,
               4
         ],
         "views": [
               {
                     "viewId": 2,
                     "mainView": true
```

```
            }
        ],
        "workShifts": [
            {
                "workShiftId": 1,
                "shiftWeekStart": "2014-08-01"
                }
            ]
        }
    }
]
```

Person's Time Zone API

The Person's Time Zone API lists the specified person's configured time zone.

Person's Time Zone API Protocol and URI

URI	`/api/rest/org/person/<personId>/timezone`
Method	GET
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	personId

Person's Time Zone API Response Fields

The JSON output includes fields defined in the following table.

Field	Description
id	Time zone ID for the user.
name	Name of the user's time zone.

Person's Time Zone API JSON File Example

The following is an example of a formatted JSON response.

```
[
   {
          "id":"GMT-6:00",
          "name":"Central Time Zone"
   }
```

Roles API

The Roles API allows you to read all of the roles of a tenant.

Roles API Protocol and URI

URI	`/api/rest/org/role/by/tenant`
Method	GET
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	tenantId

Roles API Request Fields

The POST method creates the specified role for the tenant.

Name	Req?	Description
name	Y	Name of the new tenant role.
tenantId	Y	Tenant role ID associated with the new role.
permissions	Y	Permissions assigned to the new role.
id	Y	Permission ID for the new role.
name	Y	Name of the permission for the new role.

Roles API Response Fields

The JSON output includes fields defined in the following tables.

The following fields are returned when retrieving all roles for a tenant:

Field	Description
ID	Tenant role ID.
Name	Name associated with the tenant role ID.

The following fields are returned when retrieving a specific role for a tenant:

Field	Description
ID	Tenant role ID.
Name	Name associated with the tenant role ID.
Permissions	Permissions assigned to the role.
ID	Permission ID for the role.
Name	Name of the permission for the role.

Roles API JSON File Example

The following are examples of formatted JSON responses.

Listing all tenant roles:

```
[
   {
          "id":20
          "name":"SUPERVISOR"
   },
   {
          "id":21
          "name":"TenantAdmin"
   },
   {
```

```
        "id":19
        "name":"Agent"
    }
]
```

Listing permissions for a specific role:

```
[
    {
        "id":21
        "name":"TenantAdmin"
        "tenantId":39
        "permissions":
            [{
                "id":4
                "name":"UpdateOrg"
            }
            {
                "id":3
                "name":"ViewOrg"
            }
        }
    }
]
```

Team API

The Team API allows you to create new teams and assign them to a group.

Team API Protocol and URI

URI	`/api/rest/org/common/team`
Method	POST

Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data

Team API Request Fields

Name	Req?	Description
name	Y	Name of the new team.
parentGroupId	Y	ID of the group to which the new team is assigned.
users	N	IDs of the users assigned to the new team.

Team API JSON File Example

The following is an example of a formatted JSON response.

```
[
   {
          "name":"team one",
          "parentTeamId":1,
          "users":1,2,3,4,5
   }
]
```

Teams by ID (with Agents)

The Team by ID (with Agent) API produces a list of agents within a team, or updates a team's information.

Teams by ID (with Agents) Protocol and URI

URI	`/api/rest/org/common/team/<teamId>`
Method	GET PUT DELETE

Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	teamI

Teams by ID (with Agents) Request Fields

Name	Req?	Description
name	Y	Name of the team to be added.
parentGroupId	Y	Name of the parent group of the new team.

Teams by ID (with Agents) Response Fields

The JSON output includes fields defined in the following table.

Field	Description
ID	Tenant team ID.
Name	Tenant team name.
Activated	Team activation time in epoch milliseconds.
Deactivated	Team deactivation time in epoch milliseconds.
Parent Group ID	Tenant ID for the group to which the team belongs
Agents	Detail information about the agents in the team.
Person ID	ID of the agent.
Tenant ID	ID of the tenant.
ACD ID	ID of the tenant’s ACD.
First Name	Agent’s first name.
Last Name	Agent’s last name.

Field	Description
Email	Agent's email address.
Display ID	Tenant's display ID for the agent.
Group ID	Group ID for the agent.
Activated	Agent activation time in epoch milliseconds.
Deactivated	Agent deactivation time in epoch milliseconds.
Is Synchronized?	Defines whether the agent is synchronized from an ACD. True = Agent is synchronized with an ACD. False = Agent is not synchronized from an ACD.
Time Zone	Time zone in which the team is located.
AD Login	Active Directory login name.

Teams by ID (with Agents) JSON File Example

The following is an example of a formatted JSON response.


```
[
        {
                "id":8,
                "name":"Tenant 1 Team 1",
                "activated":1388556000000,
                "deactivated":3250361200000,
                "parentGroupId":21,
                "agents":
                [
                        {
                        "personId":2,
                        "tenantId":4,
                        "acdId":24,
                        "firstName":"al",
```

```
                        "lastName":"dente",
                        "email":"al.dente",
                        "displayId":"Al Dente",
                        "groupId":6,
                        "activated":1388556000000,
                        "deactivated":3250361200000,
                        "isSynchronized":true,
                        "timeZone":null,
                        "adLogin":true
                        },
                        {
                        "personId":5,
                        "tenantId":4,
                        "acdId":24,
                        "firstName":"max",
                        "lastName":"stout",
                        "email":"max.stout",
                        "displayId":"Max Stout",
                        "groupId":6,
                        "activated":1388556000000,
                        "deactivated":3250361200000,
                        "isSynchronized":true,
                        "timeZone":null,
                        "adLogin":true
                        },
                        ...
                ]
        },
        ...
]
```

Team (with Scope)

The Team (with Scope) API allows you to list an array of teams within your scope, or to create a team in the specified group.

Team (with Scope) Protocol and URI

URI	`/api/rest/org/common/team/permission/<permission>`
Method	GET POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data
Parameter	permission

Team (with Scope) Request Fields

Name	Req?	Description
name	Y	Name of the team to be created.
ParentGroupId	Y	Parent group to which the new team will belong.

Team (with Scope) Response Fields

The JSON output includes fields defined in the following table.

Name	Description
name	Name of the team to be created.
GroupId	Parent group to which the new team will belong.

Team (with Scope) JSON File Example

The following is an example of a formatted JSON response.

```
[
        {
```

```
                "name":"Tenant 1 Team 1",
                "groupId":2,
        },
        {
                "name":"Tenant 1 Team 2",
                "groupId":2,
        },
]
```

Tenant API

The Tenant API displays a list of all tenants in your scope, or creates a tenant.

Tenant API Protocol and URI

URI	`/api/rest/org/tenant`
Method	GET POST
Permissions	ViewOrg UpdateOrg
Content Type	multipart/form-data

Tenant API Request Fields

The GET method reads all roles of a specified person mapped to groups. The result is an array of roles, and each role has an array of groups.

Name	Req?	Description
name	Y	Name of the tenant.
activated	Y	Activation time in epoch milliseconds.
deactivated	N	Deactivation time in epoch milliseconds.

Tenant API Response Fields

The JSON output includes fields defined in the following table.

Name	Description
Name	Name of the tenant.
Activated	Activation time in epoch milliseconds.
Deactivated	Deactivation time in epoch milliseconds.

Tenant API JSON File Example

The following is an example of a formatted JSON response.

```
[
        {
              "name":"tenant 1",
              "activated":1395032400000,
              "deactivated":32503615200000,
        },
        {
              "name":"tenant 2",
              "activated":1395035860000,
              "deactivated":3350369990000,
        },
]
```

Recording Controls API

The Recording Controls API exists at the server level and within the Webex WFO application user interface. It provides a means for users to create an external application that interfaces with the Webex WFO.

NOTE The Recording Controls Authentication API allows users to integrate third-party applications into Recording Controls. When using this API, the destination for Recording Controls is a data server rather than the Webex WFO Web Service.

For information on using the Recording Controls Authentication API, refer to Logging In to the Recording Controls API.

The API does the following:

- Tags calls for recording and retention—Supports the timestamp parameter. The timestamp is used to find the active CCR for the call.
- Pauses and resumes a recording—Supports the timestamp parameter. Integrations that do not support the pause and resume command include Cisco Webex Contact Center and UJET.
- Attaches user-defined metadata to calls—Supports the timestamp parameter. The timestamp is used to find the active CCR for the call.
- Segments (divides) recordings—Does not support the timestamp parameter.
- Logs on to and off of phones (agent recording only—not supported for Gateway recording)—Does not support the timestamp parameter.
- Starts and stops screen recording—Does not support the timestamp parameter.

EXAMPLE The following is an example body, including a timestamp.

```
{"acdId":"2350111", "acdServerId" : "11", "timestamp":1681316746000}
```

Where 1681316746000 is the number of milliseconds since the "Epoch."

GMT: Wednesday, April 12, 2023 4:25:46 PM

Recording Controls API Protocol and URI

URI	`/api/rest/recordingcontrols/<command>`

Method	POST
Permissions	Recording Controls
Content Type	application/JSON

URI Request Parameters

The <command> portion of the URL specifies the recording control command you are issuing. The available commands are listed below.

Query Parameters

Field	Req?	Type	Description
acdId	N	String	The agent's ID in the ACD, as a string. **NOTE** The **acdServerId** parameter is required if you use the **acdId** parameter. **NOTE** If you use the actual Cisco-generated and stored **acdServerId**, the **acdId** and **acdServerId** are used to identify the user. **NOTE** The **acdId**, **acdServerId**, and **acdServerType** fields must be included within a metadata object in the body of the request. Other documented fields need to be passed via the URL
acdServerId	N	String	The server's ID for the ACD, as a string. This can be the ID configured on the ACD Configuration page in Webex WFO, or one of the following: ■ For systems that use the Unified CCE, this is required, and is the peripheral ID.

Field	Req?	Type	Description
acdServerType	N	String	The type of ACD configured. **NOTE** **acdServerType** is required when **acdServerId** is a value other than a Cisco-generated and stored value.
active_call_only	N	Boolean	Default is false. Metadata and tag commands can apply to either the active call or the previously completed call. If the active_call_only flag is set to **true**, a metadata or tag command will only apply if a call is currently in progress. If no call is in progress, the command will be ignored. DEFAULT=False
ccrId	N	String	The unique identifier for the call contact record. Enables you to add metadata after the completion of a call. **EXAMPLE** POST request: /api/rest/recordingcontrols/metadata?ccrId=13580 Content: JSON Request body: { "metadata" : [{"name":"ticket-id-key", "value" : "ABC123"}, {"name":"account-id-key", "value" : "321XYZ"}] }

Field	Req?	Type	Description

deviceName	N		Enables you to bypass the "Username" field. Issue a recording control request without knowing the username of the specific user if you have the deviceName of the phone that is associated to the user in the "Device Associations" page of Webex WFO. This grants you an additional way to link a recording control request to a users phone call if you only know the device name. This can be used as a POST method.
			
			EXAMPLE
			POST request:
			/api/rest/recordingcontrols/pause
			Request body:
			{
			"deviceName" : "SEP000CH0724412"
			}
			
			Response:
			{
			"description": "Command PAUSE sent successfully for Evelyn Bryant",
			"personId": 123,
			"timestamp": 1624657998751
			}

Field	Req?	Type	Description
personId	N	String	Specifies the user to whom the recording control API command should apply. If the personId parameter is not provided, Webex WFO uses the authenticated user issuing the command. If you provide a user identifier, it must be for a person in the authenticated user's scope. The user identifier is one of three formats. Processing determines which format is used based on parsing the contents. ■ Person ID—A unique identifier from WfoPerson.id ■ AD Login—A domain\username (requires "\"). ■ Email address—An email address (requires "@"). **NOTE** A backslash ("\") can only be added as a parameter. It cannot be embedded in the <personId> unless it is parametrized (<?personId=domain\username>) when using a Windows login value. **EXAMPLE** When using a personId, the request looks like this: `api/rest/recording controls/user/ <personId>/<command>`

Recording Controls Authentication API Protocol and URI

URI	`/api/rest/recordingcontrols/user/5/<metadata>`
Method	POST
Permissions	Recording Controls
Content Type	application/JSON

NOTE For information on metadata syntax, refer to Recording Controls API Request Commands.

Logging In to the Recording Controls API

Currently, the Recording Controls API supports Cisco or other devices configured to use recording controls. Use the following procedure to begin using the Recording Controls API:

1. Launch the Recording Controls service on a device. The server returns the login page.

   NOTE Cisco phones use XML to display all services.

2. Enter your login credentials. Webex WFO passes credentials to the Tomcat servlet on the data server, which makes an /api/rest/authorize POST request to the application server on your device's behalf.

   After the data server receives the `200 - Successful` response from application server, it stores the session cookies and sends the device to the main recording controls page.

3. From the main page, you can select the following commands:

   - Logout
   - Pause
   - Resume
   - Segment & Save
   - Segment & Delete
   - Metadata
   - Tag
   - Screen Start
   - Screen Stop
   - Call Status

   When you make a selection, the request is again passed through the data server to the application server. The data server adds the session cookies that are stored when you logged in.

   NOTE The timestamp parameter for pause and resume commands is supported.

IMPORTANT The **Segment and Save** command splits the contact, resulting in two contacts. The **Segment and Delete** command deletes the previous segment immediately, and creates a new contact. It does not send the deleted portion of the segmented call to the recycle bin.

Logging Out of the Recording Controls API

You can keep using recording controls until the session times out or until you select **Logout**. The procedure for logging out is the same as logging in.

Recording Controls API Request Commands

The request body is a JSON object. Some commands have required or optional additional parameters. These are specified in the request body. If no additional parameters are required, an empty JSON object should be sent ({}).

Command	Gateway Recording	Smart Desktop Recording	Signaling Server Recording	Description
Login			X	Log user into the device with the given extension. Associates the user with the device for hoteling. Supported parameters in the request body: extension (string) – extension of the phone to log in to Example request body: `{ "extension" : "1234" }`
Logout			X	Log user out of the device with the given extension. Removes the association of the user to the device for hoteling. Supported parameters in the request body: extension (string) – extension of the

Command	Gateway Recording	Smart Desktop Recording	Signaling Server Recording	Description
				phone to log out of Example request body: `{ "extension" : "1234" }`

Metadata	X	X	X	Attaches metadata to the active call.
				
				If there is no active call and active_call_only is false, this will attach metadata to the previous call. Supported parameters in the request body:
				
				metadata (array of objects)
				
				■ name (string) – metadata key
				■ value (string) – metadata value
				
				Example request body:
				
				`{`
				`"metadata" : [ {"name" :`
				`"custom-data-key", "value"`
				`: "my custom data"} ]`
				`}`

Pause	X	X	X	Pause both screen and audio recording for the active call. **NOTE** When Gateway recording is used, Pause events are inserted into audio recordings during the reconciliation processes. If the recording is already paused, this command is ignored.

Command	Gateway Recording	Smart Desktop Recording	Signaling Server Recording	Description
Resume	X	X	X	Resume both screen and audio recording for the active call. If the recording is not paused, this command is ignored.
Segment		X	X	Start a new segment contact for the active call. Supported parameter in the request body: **delete (string)**—whether to delete the previous segment ■ Values = true or false ■ Defaults to false (save segment) ■ If true, the previous segment in the call is deleted immediately. ■ If false, end of interaction workflow is processed on the previous segment like a normal call. Example request body: `{ "delete" : "true" }`
Start_Screen		X	X	Start a screen only recording. ■ If a regular call recording is in progress, a separate screen-only recording will be created. ■ If another screen-only recording has already been initiated, this command is ignored.

Command	Gateway Recording	Smart Desktop Recording	Signaling Server Recording	Description
Stop_Screen		X	X	End a screen-only recording. If no screen-only recording is in progress, this command is ignored.
Tag	X	X	X	Sets the reason for the active call to tagged. If there is no active call and active_call_only is false, this will change the reason of the previous call.
call_status			X	Returns a response that indicates if the CTI Signaling Service is aware of a call for the user. If the current call for the user has a call contact record recording in the database, the ID for that call contact record is also returned. Supported parameters in the request body: ■ ani (string) – Active call information only returns true if the current call's ANI matches. ■ dnis (string) – Active call information only returns true if the current call's DNIS matches. Example request body: `{ "ani" : "1234" }` This command returns two attributes in response to the request. ■ callActive (boolean) – True if the CTI Signaling Service is aware of

Command	Gateway Recording	Smart Desktop Recording	Signaling Server Recording	Description
				a call for the user. ■ contactId (integer) – Returns the Webex WFO contact ID for the call's Webex WFO contact if it exists. Example response attributes: `{ "callActive" : true, "contactId" : 56789 }`

Using Recording Controls from the Command Line

You can use the command line to send Recording Controls commands.

NOTE If you choose to use the command line, the commands and values must match the configured commands and values. If they don't match, the command will have no effect.

The following table lists the commands and their associated command line formats. These commands are not case-sensitive.

Command	Format
Metadata	Set the value of a defined metadata key to be associated with the current contact. The key cannot be blank. If the key's string value does not match a predefined metadata key in the system, it is ignored. The value is optional. (The second pipe character in the string must still exist.) If left blank, it is assumed that you want to set the key's value to a blank string or the numeric value of zero. The format of the value must match the format of the key that it is setting. The formats include string, number, and date. `dcc -c "metadata|<MetadataKey>|<MetadataValue>"`
Pause	Temporarily pause the capturing of desktop data, audio, and screen.

Command	Format
	Whatever media is being actively captured will be paused when this command is issued. The service will stop writing data to the capture files. In the case of the desktop analytics data, it continues to write data, but any keypress events will result in the particular key being pressed to be written as an asterisk in the capture file. **NOTE** When Gateway recording is used, Pause events are inserted into audio recordings during the reconciliation processes. This command can be used to pause screen-only recordings: `dcc -c pause`
Resume	Resume previously paused media capture. `dcc -c resume`
Segment Start	Start a new segment contact for the active call. `dcc -c segment`
Segment Stop	Stop a new segment contact for the active call. `dcc -c segment|<ani>|<dnis>|<associated>|<delete>` **EXAMPLE** For segment and delete: dcc -c segment||||true
Tag	Send an audit message with the TAG command, user name, and timestamp `.dcc -c tag`

Simplified Recording Controls API

When Simplified Recording Controls are set up, Webex WFO uses the data server's authentication to handle recording control API requests. As a result, users won't need to specify a username and password when issuing Recording Controls commands. The request is sent to the URL of the Webex WFO data server using the GET or POST method.

Protocol and URI

URI	GET: `/api/rest/recordingcontrols/<command>?acdId=<agent's ACD ID>&acdServerId=<agent's ACD server ID>` POST: `/api/rest/recordingcontrols/<command>`
Method	GET and POST
Permissions	Recording Controls
Content Type	application/JSON

GET Example

EXAMPLE

```
http://<Data Server IP>:8080/api/rest/recordingcontrols/<command>?acdId=<agent's ACD ID>&acdServerId=<agent's ACD server ID>
```

Where in the above example,

- `<Data Server IP>` — Enter the IP address of the data server.
- `<command>` — Enter a command such as pause, resume, or tag. The full list of commands are detailed in the table below.
- `<agent's ACD ID>` — Enter the ACD ID of the agent.

- `<agent's ACD server ID>` — Enter the server ID of the agent's ACD. The ACD Server ID can be found in Application Management > ACD Configuration

POST Example

EXAMPLE

```
http://<Data Server IP>:8080/api/rest/recordingcontrols/<command>
```

Where in the above example,

- `<Data Server IP>` — Enter the IP address of the data server.
- `<command>` — Enter a command such as pause, resume, or tag. The full list of commands are detailed in the table below.

Set up Simplified Recording Controls

1. If the Webex WFO Data Server Web Services is disabled, restart it and set its Startup type to **Automatic**.
2. Verify that tomcat server started up and deployed wars successfully. Look in the latest catalina.*.log file in `C:\Program Files\Calabrio ONE\Data Server\Tomcat\logs`.
3. Set the server port in the URL to 8080.

   EXAMPLE http://192.0.2.0:8080/api/rest/recordingcontrols/metadata
4. Verify that your system is up and running correctly, using a tool such as Poster.

   Use the following to make the GET request:

   ```
   GET http://192.0.2.0:8080/api/rest/system/tenant

   Content-Type: text/html;charset=utf-8
   ```

Use the following to make the POST request to test the metadata:

```
POST http://10.192.102.22:8080/api/rest/recordingcontrols/metadata

{
   "acdServerId": "43",
   "acdId": "16663",
   "metadata": [{
         "name":
```

```
            "text1",
            "value": "Test data"
            }
    ]
}
```

If this test indicates that your system is not functioning, check the latest localhost_access_log for a `localhost_access_log.*.txt`. This log will indicate whether a request was made to this server. If it does not, this is an indication that requests are not being sent to this server on port 8080, and you have an issue with the sender.

Command Descriptions

The following list details the available commands.

> **NOTE** Each command can use GET and POST methods. See GET Example for GET. POST may vary depending on the command. See the POST example in the description of each command below.

Command	Description

Metadata	Attaches metadata to the active call.
	
	If there is no active call and active_call_only is false, this will attach metadata to the previous call. Supported parameters in the request body:
	
	metadata (array of objects)
	
	■ name (string) – metadata key
	■ value (string) – metadata value
	
	Example POST request body:
	
	`{`
	`"acdServerId": "43",`
	`"acdId": "16663",`
	`"metadata": [{`
	`"metadata-key":`
	`"text1",`
	`"value": "Test data"`

Command	Description
	}] }
Pause	Pause both screen and audio recording for the active call. **NOTE** When Gateway recording is used, Pause events are inserted into audio recordings during the reconciliation processes. If the recording is already paused, this command is ignored. POST example: ▪ `{"acdServerId":"1","acdId":"123ABC"}`
Resume	Resume both screen and audio recording for the active call. If the recording is not paused, this command is ignored. POST example: ▪ `{"acdServerId":"1","acdId":"123ABC"}`
Segment	Start a new segment contact for the active call. POST example: `{"acdServerId":"1","acdId":"123ABC","delete":"true"}`
Start_Screen	Start a screen only recording. ▪ If a regular call recording is in progress, a separate screen-only recording will be created. ▪ If another screen-only recording has already been initiated, this command is ignored. POST example: ▪ `{"acdServerId":"1","acdId":"123ABC"}`
Stop_Screen	End a screen-only recording. If no screen-only recording is in progress,

Command	Description
	this command is ignored. POST example: ▪ `{"acdServerId":"1","acdId":"123ABC"}`
Tag	Sets the reason for the active call to tagged. If there is no active call and active_call_only is false, this will change the reason of the previous call. POST example: ▪ `{"acdServerId":"1","acdId":"123ABC"}`

Single Contact Export API

The Single Contact Export API allows you to download a contact from Webex WFO.

Protocol and URI

URI	`/api/rest/recording/media/export/contact/<id>?type=<format>`
Method	POST, GET
Permissions	Export Recordings
Content Type	application/json
Parameters	<id> = The contact ID of the contact that you want to export. If the <id> does not exist, you receive the following error message: "Requested contact does not exist!" <format> = The media format in which you want to export the contact. Audio/video formats: ■ WEBM ■ WEBM_VP8 ■ WEBM_VP9 Audio-only formats: ■ WAV ■ OPUS

Download a single contact

1. Submit a POST request to the URL. Wait for a period of time for Webex WFO to prepare the contact for download.
2. Submit a GET request with the same parameters to the same endpoint.

You may receive an error message like the one below detailing that the file does not exist or may have expired. If so, wait a little longer and try again.

```
{
"errorMessage": "The following file does not exist or may have expired:
   c309.wav",
"requestId": "31b466da-cbd2-42fe-bb87-99825e7572b1"
}
```

Example Response

The following is an example POST response:

```
{
     "encoding": "true"
}
```

Speech Hits API

Use this API to request a list of Analytics speech hits from .

Speech Hits API Protocol and URI

URI	`/api/rest/cas/contact`
Method	GET
Parameters	CCR ID
Permissions	View Audio Analytics
Content Type	multipart/form-data

Speech Hits API Response Fields

Name	Description
date	Date analytics was performed, in UNIX epoch milliseconds.
hits	Detail information regarding the speech hits.
phrase	Speech phrase.
stop	Stop point of the speech hit.
audit	Number of the speech audit.
confidence	Confidence probability of the speech hit.
start	Start point of the speech hit.
channel	Channel number of the speech hit.

Name	Description
category	Category of the speech hit.
path	Path of the speech hit.
firstName	Agent’s first name.
lastName	Agent’s last name.
contactId	ID number for the call.
team	Agent’s team name.
displayId	Agent’s display ID.
group	Agent’s group name.

Speech Hits API JSON File Example

The following is an example of a formatted JSON response file.

```
{
                "date": epoch milliseconds,
                "hits": [{
                      "phrase": string,
                      "stop": number,
                      "audit": number,
                      "confidence": number,
                      "start": number,
                      "channel": number,
                      "category": string
                },
                ...
          ],
          "path": "",
          "firstName": string,
          "lastName": string,
```

```
        "contactid": number,
        "team": string,
        "displayId": string,
        "group": string
}
```

Survey Configuration API

Use this API to request QM survey configuration information from .

Survey Configuration API Protocol and URI

URI	/api/rest/recording/survey
Method	GET
Permissions	Administer QM Surveys View Survey Results QM Survey Reports
Content Type	multipart/form-data

Survey Configuration API Response Fields

Name	Description
id	Survey ID.
name	Survey name.
description	Description of the survey.
creator	Survey creator.
created	Time the survey was created in UNIX epoch milliseconds.
totalPoints	Total number of points for the survey (one decimal place).
providerSurveyId	Survey ID for the provider.

Name	Description
surveyProviderConnectionId	Connection ID for the survey provider.
statusFK	Field key for the survey status.
metaDataFieldKey	Field key for the metadata.
lastModified	Date the survey was last modified.
properties	Detail regarding survey properties.
questions	Detail regarding survey questions.
id	Question ID number.
surveyId	Survey ID number.
label	Question label.
description	Question description.
providerQuestionId	Provider’s question ID.
type	Question type.
weight	Weight given to the question.
isKPI	Key performance indicator for the question.
lastModified	Date the question was last modified.
answers	Detailed information rgarding the question’s answers.
id	Answer ID number.
questionId	Question ID.
label	Question label.
description	Question description.
providerQuestionId	Provider’s question ID.
type	Question type.

Name	Description
weight	Weight given to the question.
value	Value of the answer.
lastModified	Time the values for the answer were last modified, in UNIC epoch time.

Survey Configuration API JSON File Example

The following is an example of a formatted JSON response file.

```
[
        {
              "id": number,
              "name": string,
              "description": string,
              "creator": string,
              "created": epoch,
              "totalPoints": 0.0,
              "providerSurveyId": string,
              "surveyProviderConnectionId": number,
              "statusFK": 2,
              "metaDataFieldKey": string,
              "lastModified": string,
              "properties": {},
              "questions": [{
                            "id": number,
                            "surveyId": number,
                            "label": string,
                            "description": string,
                            "providerQuestionId": string,
                            "type": string,
                            "weight": decimal,
                            "isKPI": Boolean,
```

```
                        "lastModified": epoch,
                        "answers": [{
                              "id": number,
                              "questionId": number,
                              "label": string,
                              "description": string,
                              "providerAnswerId": string,
                              "type": string,
                              "weight": deimal,
                              "value": string,
                              "lastModified": epoch
                        },
                        ...
                  ]
            },
            ...
            ]
      },
      ...
]
      {
            "id": number, // id
            "name": string, // name
            "description": string, // description
            "creator": string,
            "created": epoch,
            "totalPoints": 0.0,
            "providerSurveyId": string, // provider's id
            "surveyProviderConnectionId": number,
            "statusFK": 2,
            "metaDataFieldKey": string,
            "lastModified": string,
```

```
"properties": {},
"questions": [{
            "id": number,
            "surveyId": number,
            "label": string,
            "description": string,
            "providerQuestionId": string,
            "type": string,
            "weight": decimal,
            "isKPI": Boolean,
            "lastModified": epoch,
            "answers": [{
                        "id": number,
                        "questionId": number,
```

Survey ID API

Use this API to request information on a specific QM survey from .

Survey ID API Protocol and URI

URI	/api/rest/recording/survey/<id>
Method	GET
Parameter	<id> = Survey ID.
Permissions	Administer QM Surveys View Survey Results QM Survey Reports
Content Type	multipart/form-data

Survey ID API Response Fields

Name	Description
id	Survey ID.
name	Survey name.
description	Description of the survey.
creator	Survey creator.
created	Time the survey was created in UNIX epoch milliseconds.
totalPoints	Total number of points for the survey (one decimal place).

Name	Description
providerSurveyId	Survey ID for the provider.
surveyProviderConnectionId	Connection ID for the survey provider.
statusFK	Field key for the survey status.
metaDataFieldKey	Field key for the metadata.
lastModified	Date the survey was last modified.
properties	Detail regarding survey properties.
questions	Detail regarding survey questions.
id	Question ID number.
surveyId	Survey ID number.
label	Question label.
description	Question description.
providerQuestionId	Provider’s question ID.
type	Question type.
weight	Weight given to the question.
isKPI	Key performance indicator for the question.
lastModified	Date the question was last modified.
answers	Detailed information rgarding the question’s answers.
id	Answer ID number.
questionId	Question ID.
label	Question label.
description	Question description.
providerQuestionId	Provider’s question ID.

Name	Description
type	Question type.
weight	Weight given to the question.
value	Value of the answer.
lastModified	Time the values for the answer were last modified, in UNIC epoch time.

Survey ID API JSON File Example

The following is an example of a formatted JSON response file.

```
[
        {
              "id": number,
              "name": string,
              "description": string,
              "creator": string,
              "created": epoch,
              "totalPoints": 0.0,
              "providerSurveyId": string,
              "surveyProviderConnectionId": number,
              "statusFK": 2,
              "metaDataFieldKey": string,
              "lastModified": string,
              "properties": {},
              "questions": [{
                            "id": number,
                            "surveyId": number,
                            "label": string,
                            "description": string,
                            "providerQuestionId": string,
                            "type": string,
                            "weight": decimal,
```

```
                        "isKPI": Boolean,
                        "lastModified": epoch,
                        "answers": [{
                              "id": number,
                              "questionId": number,
                              "label": string,
                              "description": string,
                              "providerAnswerId": string,
                              "type": string,
                              "weight": deimal,
                              "value": string,
                              "lastModified": epoch
                        },
                        ...
                  ]
            },
            ...
            ]
      },
      ...
]
```

Text Hits API

Use this API to request a list of Analytics text hits from .

Text Hits API Protocol and URI

URI	`/api/rest/cas/textview`
Method	GET
Parameters	CCRID
Permissions	View Audio Analytics
Content Type	multipart/form-data

Text Hits API Response Fields

Name	Description
date	Date analytics was performed, in UNIX epoch milliseconds.
lastName	Agent’s last name.
receiver	Person to whom the email message was addressed.
emailBody	Text of the email message.
subject	Subject line of the email message.
groupId	ID number of the agent’s group.
team	Name of the agent’s team.
ccrId	Call completion record ID.

Name	Description
hits	Detail information regarding the text analytics hit.
position	Location of the hit within the message.
length	Length of the hit.
phrase	Phrase of the hit.
category	Category of the hit.
agentName	Agent's full name.
teamId	Agent's team ID number.
personID	Person ID for the person handling the message.
group	Agent's group name.

Text Hits API JSON File Example

The following is an example of a formatted JSON response file.

```
{
          "date": epoch milliseconds,
          "lastName": string,
          "receiver": [string],
          "emailBody": string,
          "subject": null,
          "team": string,
          "ccrId": number,
          "hits": [{
                       "position": number,
                       "length": number,
                       "phrase": string,
                       "category": string
                 },
                 ...
```

```
            ],
            "firstName": string,
            "sender": string,
            "agentname": string,
            "teamId": number,
            "personId": number,
            "group": string
}
```

Transcript Export API

The Transcript Export API allows you to retrieve the transcript of a single call using that call's call completion record ID. Every call found in a bulk contact export ZIP file includes a call completion record ID in the metadata. See "Create a Bulk Contact Export File" in the *Webex WFO User Guide* to set up recurring bulk exports or an ad hoc bulk export.

To retrieve the results, a configured user with Administer Speech to Text Analytics permission needs a cookie with the key: hazelcast.sessionId and a value corresponding to the authorized user. The session ID can be retrieved using the GET method for the Authorize API.

Transcript Export API Protocol and URI

URI	`/api/rest/cas/speechtextview`
Method	PUT
Permissions	Administer Speech to Text Analytics
Content Type	application/json

Transcript Export API Request Fields

Name	Req?	Description
ccrid	Y	The call completion record ID.
isRootRecording	Y	Indicates whether the recording is the original recording or the reconciled recording associated with an agent. For recordings associated with an agent, this field will always be False.

Transcript Export API CURL File Example

The following is an example of a formatted CURL request.

```
curl ‘https://<ip-address>/api/rest/cas/speechtextview’ -X PUT -H ‘Cookie:
   hazelcast.sessionId=HZE86954DF92F6472AB6298EBFB0CAE3E6’ -H ‘Content-Type:
   application/json; charset=UTF-8’ -H ‘Accept: */*’ --data-binary
‘{
   "ccrid":882488,
   "isRootRecording":"false"
}’
```