Cisco Meeting Server

Cisco Meeting Server Release 3.8

API Reference Guide

March 21, 2025

Contents

Change History 12

1 General Information 13
1.1 How to Use this Document 13
1.1.1 Interactive API Reference tool 13
1.2 Summary of 3.8 API additions and changes 15
1.2.1 Assigning a name to callProfiles API 17
1.2.2 Configuring screen resolution for content shared in web app meeting 17
1.2.3 Adding tags to coSpaces 18
1.2.4 Enabling surveys in Meeting Server 18
1.2.5 Support to send chat messages to a particular participant in a web app meeting 19

2 General Structure of Objects 21
2.1 Object Hierarchy 21

3 Accessing the API 26
3.1 Configuration Settings 26
3.2 Authentication 26
3.3 API access on the web interface 26
3.4 Third Party Tools to Use 27
3.4.1 Tips for using third-party API Tools 28

4 API Methods 30
4.1 URL format 30
4.2 GET Methods 31
4.2.1 Collections level 31
4.2.2 Using limit and offset at the Collections level 31
4.2.3 Individual object level 32
4.2.4 HTTP specifics 32
4.2.5 How GET methods are detailed in this document 33
4.3 POST Methods 33
4.3.1 HTTP specifics 33
4.3.2 How POST methods are detailed in this document 34
4.4 PUT Methods 34
4.4.1 HTTP specifics 34

4.4.2 How PUT methods are detailed in this document 34
4.5 DELETE Methods 34
4.6 Failure Reasons 35

5 Example Requests and Responses for Specific Methods 42
5.1 Retrieval of Current Active Calls 42
5.2 Instantiating a new call and connecting a participant to it 42

6 coSpace Related Methods 45
6.1 Retrieving coSpaces 45
6.2 Creating and Modifying a coSpace 47
6.2.1 Secondary coSpace URIs 51
6.2.2 Auto-generation of coSpace callId 52
6.3 Retrieving Detailed Information about a Single coSpace 52
6.3.1 Retrieving entry details for a specific coSpace 56
6.4 coSpace Member Methods 56
6.4.1 Retrieving the members of a coSpace 56
6.4.2 Adding and modifying a coSpace member 58
6.4.3 Retrieving Information on a coSpace member 60
6.5 Multiple coSpace Access Methods 60
6.5.1 General information 60
6.5.2 Retrieving coSpace access methods 61
6.5.3 Creating and modifying coSpace access methods 62
6.5.4 Retrieving information on an individual coSpace access method 65
6.6 Calling Out from a coSpace 65
6.7 Bulk creating, updating and deleting coSpaces 65
6.7.1 Creating /cospaceBulkParameterSets 65
6.7.2 Retrieving the parameter sets for creating coSpaces in bulk 67
6.7.3 Retrieving information on an individual /cospaceBulkParameterSet 67
6.7.4 Queueing the bulk sync operations 69
6.7.5 Retrieving the bulk sync operations 69
6.7.6 Retrieving a specific bulk sync operation 71
6.7.7 Examples 72
6.8 coSpace Diagnostics Methods 73
6.9 Using coSpace templates 73
6.9.1 Creating, modifying, retrieving, enumerating and deleting coSpace templates 73

6.10 Using Access Method templates 75
6.10.1 Creating, modifying, retrieving, enumerating and deleting coSpace template access method templates 75
6.11 Retrieving text based meeting entry information for a coSpace 78
6.12 Configuring metadata on a coSpace 80

7 Dial Plan Methods 82
7.1 Outgoing Dial Plan API Methods 82
7.1.1 Access to the outgoing dial plan 82
7.1.2 Retrieving outbound dial plan rules 83
7.1.3 Creating and modifying outbound dial plan rules 83
7.1.4 Retrieving information on an individual outbound dial plan rule 86
7.2 Dial Transforms 86
7.2.1 Retrieving dial transforms 87
7.2.2 Setting up and modifying dial transforms 88
7.2.3 Retrieving detailed information about an individual dial transform 89
7.3 Incoming Call Matching Dial Plan API Methods 89
7.3.1 Access to incoming domain matching rules 89
7.3.2 Retrieving incoming dial plan rules 90
7.3.3 Creating and modifying incoming dial plan rules 91
7.3.4 Retrieving information on an individual incoming dial plan rule 92
7.4 Incoming Call Forwarding Dial Plan API Methods 92
7.4.1 Access to incoming call forwarding rules 92
7.4.2 Retrieving incoming call forwarding dial plan rules 92
7.4.3 Creating and modifying incoming call forwarding dial plan rules 93
7.4.4 Retrieving information on an individual incoming call forwarding dial plan rule 94

8 Call Related Methods 95
8.1 Call Methods 95
8.1.1 Retrieving Information on Active Calls 95
8.1.2 Creating a New Call and Modifying an Active call 96
8.1.3 Retrieving Information on an Individual Active Call 100
8.1.4 Generating diagnostics for an individual call 105
8.1.5 Retrieve participants in a conference 106
8.1.6 Creating a new participant for a specified call 106
8.1.7 Set properties for all participants in a conference 116
8.1.8 Send closed captions using third party API tools 118

8.2 Call Profile Methods 118
8.2.1 Retrieving call profiles 118
8.2.2 Setting up and modifying call profiles 118
8.2.3 Retrieving detailed information about an individual call profile 123
8.3 Call Leg Methods 123
8.3.1 Retrieving Information on Active Call Legs 123
8.3.2 Adding and Modifying Call Legs 125
8.3.3 /callLegs/<call leg id>/generateKeyframe 134
8.3.4 Allowing Far End Camera Control (FECC) using the API 134
8.3.5 Taking snapshots of participants in a meeting 134
8.3.6 Retrieving Information on an Individual Call Leg 135
8.4 Call Leg Profile Methods 152
8.4.1 General information 152
8.4.2 Retrieving call leg profiles 152
8.4.3 Creating and modifying a call leg profile 161
8.4.4 Retrieving information on an individual call leg profile 170
8.4.5 Example call leg profile & access method use 170
8.4.6 /callLegProfiles/<call leg profile id>/usage object method 171
8.4.7 /callLegs/<call leg id>/callLegProfileTrace object method 171
8.5 Setting Individual Features for a Call Leg 174
8.6 Layout template Methods 174
8.6.1 Adding and Modifying Layout Templates 175
8.6.2 Retrieving (Enumerate) of Layout Templates 175
8.6.3 Retrieving Information on Layout Templates 175
8.6.4 Assigning a layout template description to Layout Templates 176
8.6.5 Retrieving a layout template description on Layout Templates 176
8.7 Call Branding Profile Methods 176
8.7.1 Retrieving Call Branding Profiles 176
8.7.2 Setting up and modifying call branding profiles 177
8.7.3 Retrieving detailed information about an individual call branding profile 177
8.8 DTMF Profile Methods 177
8.8.1 Retrieving DTMF Profiles 177
8.8.2 Setting up and modifying DTMF Profiles 179
8.8.3 Retrieving detailed information about an individual dtmfProfile 181
8.9 IVR Methods 181
8.9.1 Retrieving IVRs 181

8.9.2 Setting up and modifying IVRs ... 182
8.9.3 Retrieving detailed information about an individual IVR ... 182
8.10 IVR Branding Profile Methods ... 182
8.10.1 Retrieving IVR branding profiles ... 182
8.10.2 Setting up and modifying an IVR branding profile ... 183
8.10.3 Retrieving detailed information about an individual IVR branding profile ... 183
8.11 Participant Related Methods ... 183
8.11.1 Retrieving participants ... 184
8.11.2 Changing settings for a participant already in a conference ... 185
8.11.3 Retrieving detailed information on an individual participant ... 186
8.11.4 Retrieving a participant's call legs ... 190
8.11.5 Limiting a call's participants ... 190

9 User Related Methods ... 192
9.1 Retrieving Users ... 192
9.2 Retrieving Detailed Information on an Individual User ... 193
9.2.1 Retrieving a User's coSpace Associations ... 193
9.2.2 Applying coSpace templates to users ... 193
9.2.3 Retrieving users coSpace template information ... 194
9.2.4 Applying userCoSpaceTemplates with LDAP ... 194
9.2.5 Retrieving user provisioned coSpace information ... 195
9.2.6 Creating, modifying, and retrieving LDAP user provisioned coSpace sources ... 196
9.2.7 Creating, modifying, and retrieving LDAP user provisioned coSpace mappings ... 198
9.3 User Profile Methods ... 199
9.3.1 Retrieving user profiles ... 199
9.3.2 Setting up and modifying user profiles ... 200
9.3.3 Retrieving detailed information about an individual user profile ... 201

10 System Related Methods ... 202
10.1 Retrieving System Status ... 202
10.2 Retrieving System Alarm Status ... 203
10.3 Retrieving System Database Status ... 206
10.4 CDR Receiver URI Methods ... 207
10.4.1 Retrieving the CDR Receivers URI ... 207
10.4.2 Setting the CDR Receivers URI ... 208
10.5 Global Profile Methods ... 208

10.5.1 Retrieving the Global Profile 208
10.5.2 Setting the Global Profile 208
10.6 Licensing Methods 209
10.6.1 Retrieving license file information on a per Meeting Server instance 209
10.6.2 Retrieving cluster licensing information 218
10.7 TURN Server Methods 220
10.7.1 Retrieving Information on TURN Servers 221
10.7.2 Setting up and modifying TURN servers 221
10.7.3 Retrieving detailed information about an individual TURN server 222
10.7.4 Retrieving individual TURN Server status 223
10.8 Web Bridge Methods 224
10.8.1 Retrieving Information on Web Bridges 224
10.8.2 Setting Up and Modifying a Web Bridge 225
10.8.3 Retrieving detailed information about an individual Web Bridge 225
10.8.4 Updating the Web Bridge customization 226
10.8.5 Retrieving diagnostics on a Web Bridge (from 2.2) 226
10.9 Web Bridge Profile Methods 226
10.9.1 Retrieving web bridge profiles 226
10.9.2 Creating and modifying web bridge profiles 229
10.9.3 Finding out the web bridge profile currently in effect on a specified web bridge 231
10.9.4 Finding out the web bridge profile currently in effect at the top level (global) system level 232
10.9.5 Retrieving the web bridge profile currently in effect on an individual tenant 233
10.9.6 Creating, modifying, and retrieving Web Bridge addresses for a webBridgeProfile 235
10.9.7 Creating, modifying, and retrieving IVR numbers for a webBridgeProfile 236
10.10 Call Bridge Methods 237
10.10.1 Retrieving Information on Call Bridges 237
10.10.2 Setting Up and Modifying a Call Bridge 238
10.10.3 Retrieving detailed information about an individual Call Bridge 238
10.11 Call Bridge Group Methods 238
10.11.1 Retrieving Information on Call Bridge Groups 238
10.11.2 Setting Up and Modifying a Call Bridge Group 239
10.11.3 Retrieving detailed information about an individual Call Bridge Group 239
10.12 Call Bridge Cluster Methods 240

10.12.1 Retrieving the Call Bridge Cluster details 240
10.12.2 Setting Up and Modifying the Call Bridge Cluster 240
10.13 System Load Method 241
10.14 Compatibility Profile Methods 241
10.14.1 Retrieving compatibility profile operations 241
10.14.2 Setting up and modifying compatibility profile operations 242
10.15 Lobby Profile Methods 245
10.15.1 Retrieving lobby profiles operations 245
10.15.2 Setting up and modifying lobby profiles operations 246
10.16 System Diagnostics Methods 246
10.16.1 Retrieving system diagnostics 246
10.16.2 Retrieving an individual system diagnostic 247
10.16.3 Retrieving the content of an individual system diagnostic 247
10.17 System timedlogging methods 247
10.17.1 Activating a logging subsystem 247

11 LDAP Methods 249
11.1 LDAP Server Methods 250
11.1.1 Retrieving Information on LDAP Servers 250
11.1.2 Adding and modifying an LDAP Server 251
11.1.3 Retrieving detailed information about an individual LDAP Server 251
11.2 LDAP Mapping Methods 252
11.2.1 Adding and modifying an LDAP Mapping 252
11.2.2 Secondary LDAP Mapping parameter 253
11.2.3 Retrieving information on LDAP Mappings 253
11.2.4 Retrieving detailed Information about an individual LDAP Mapping 254
11.3 LDAP Source Methods 254
11.3.1 Retrieving Information on LDAP Sources 254
11.3.2 Adding and modifying an LDAP Source 255
11.3.3 Retrieving detailed information on a LDAP Source 256
11.4 LDAP Sync Methods 256
11.4.1 Retrieving scheduled LDAP sync methods 257
11.4.2 Initiating a new LDAP sync 258
11.4.3 Cancelling a scheduled LDAP sync 258
11.4.4 Retrieving information on a single LDAP sync method 258
11.5 External Directory Search Locations 259
11.5.1 Retrieving Information on external directory search locations 259

11.5.2 Adding and modifying external directory search locations 259
11.5.3 Retrieving detailed information on external directory search locations 260
11.5.4 Example of adding external directory search locations 260

12 Multi-tenancy 264
12.1 Tenants 265
12.1.1 Retrieving Tenants 265
12.1.2 Creating and Modifying a Tenant 266
12.1.3 Retrieving Detailed Information about an Individual Tenant 267
12.2 Tenant group operations 267
12.2.1 Retrieving Tenant Groups 267
12.2.2 Creating and Modifying a Tenant Group 268
12.2.3 Retrieving Detailed Information about an Individual Tenant Group 268

13 Query Methods 269
13.1 accessQuery Method 269
13.2 conversationIdQuery Method 269
13.3 uriUsageQuery Method 270

14 Dial-in security profile methods 271
14.1 General information 271
14.1.1 Using dial-in security profiles to implement minimum passcode length 272
14.2 Retrieving dial-in security profiles 272
14.3 Creating and modifying dial-in security profiles 273

15 Scheduler APIs 274
15.1 Configuring the Scheduler's HTTPS interface 274
15.2 Scheduler API message format 275
15.3 Content header format 275
15.4 Response codes 275
15.5 Custom Error Codes 276
15.6 Example request and response 276
15.7 Scheduler related methods 278
15.7.1 Retrieving service health status 278
15.7.2 Creating, modifying, and deleting meetings in a coSpace 279
15.7.3 Retrieving, modifying, or deleting individual meetings 282
15.7.4 Modifying or retrieving the list of participants in a meeting 284
15.7.5 Creating, modifying, and deleting meeting occurrences 285

15.7.6 Retrieving or modifying the participants in a meeting occurrence 287
15.7.7 Refreshing emails for a coSpace 288
15.7.8 Refreshing emails for single meeting and whole meeting series 289
15.7.9 Modifying and retrieving timed logging information 289

16 Using Profiles 290

Appendix A mall Change participant roles in a web app meeting 292
A.1 Available roles 292
A.2 Special roles 293
A.2.1 Space default role 293
A.2.2 Custom role 294
A.3 Assigning roles and coSpace membership 294
A.3.1 Members 294
A.3.2 Other participants 294

Appendix B Additional Call Leg Information 295
B.1 Call Leg Information 295
B.1.1 Call leg configuration 295
B.1.2 Call leg status 296

Appendix C Additional Multiparty Licensing Information 298
C.1 /system/licensing Information 298
C.2 /system/multipartyLicensing Information 299

Appendix D Installing and using RESTer with the API 300
D.1 Installing RESTer 300
D.1.1 Add/Confirm SSL Exception 301
D.1.2 Generate a Basic Auth header 302
D.1.3 GET command example 303
D.1.4 POST command example 304
D.1.5 PUT command example 308

Appendix E API methods presented by Web Bridge 3 309

Cisco Legal Information 310

Cisco Trademark 311

Figures:

Figure 1: Cisco Meeting Server documentation for release 3.8 15
Figure 2: Accessing the Call Bridge API via the Meeting Server web interface 27
Figure 3: Outline LDAP process 250
Figure 4: Outline multi-tenancy process 265
Figure 5: Value inheritance with profiles 291
Figure 6: Value inheritance with profiles when using tenants 291

Change History

Date	Change Summary
September 07, 2023	New version for Cisco Meeting Server 3.8. See *Summary of API additions* for an overview of all API additions and changes.
March 16, 2023	New version for Cisco Meeting Server 3.7.
October 13, 2022	Minor edits.
August 23, 2022	New version for Meeting Server 3.6.
July 21, 2022	Updated descriptions for URI and Secondary URI parameters.
April 20, 2022	New version for Cisco Meeting Server 3.5.
December 23, 2021	Updated Scope parameter under Cospace related methods.
December 15, 2021	New version for Cisco Meeting Server 3.4.
December 6, 2021	Updated parameter audioGainMode.
August 24, 2021	New version for Cisco Meeting Server 3.3.
July 19, 2021	Added the FAQ link in the Email Invitation API section.
May 6, 2021	Updated the descriptions of the parameters: qualityMain, qualityPresentation, remoteParty, directorySearchLocation, sipMultistream.
April 22, 2021	Minor edits.
April 21, 2021	New version for Cisco Meeting Server 3.2.
April 20, 2021	Added notes for the parameters sipPresentationChannelEnabled and loadBal-anceIndirectCalls.
March 16, 2021	Updated the document for short term credentials on the Meeting Server being a fully supported feature.
December 02, 2020	Added note to qualityMain and qualityPresentation parameters for callLegProfile regarding web app
November 30, 2020	New version for Cisco Meeting Server 3.1
November 04, 2020	Minor corrections.
September 04,2020	Minor corrections.
August 12, 2020	Minor corrections.
July 29, 2020	New version for Cisco Meeting Server 3.0,

1 General Information

The Cisco Meeting Server software can be hosted on specific servers based on Cisco Unified Computing Server (UCS) technology or on a specification-based VM server. Cisco Meeting Server is referred to as the Meeting Server throughout this document.

Note: Cisco Meeting Server software version 3.0 onwards does not support X-Series servers.

This document covers release 3.8 of the Application Program Interface for the Cisco Meeting Server.

Note: The Cisco Meeting Server software is referred to as the Meeting Server throughout the remainder of this guide.

Note: We strongly recommend that you try to keep API configuration changes, especially global profile changes that could affect multiple users/calls, to non-business hours within a maintenance window.

1.1 How to Use this Document

This guide falls into two parts:

- We suggest that you start by reading sections 2 to 5 inclusive and in order. These sections will help you understand the concepts behind the API, the way to use the API methods and some examples of usage.
- Then use the remaining sections (section 6 onwards) as reference material for the method(s) that you want to use.

  The guide covers the following different APIs:

  - Sections 6 to 14 cover the XML-based Call Bridge APIs.
  - Section 15 covers the Scheduler APIs, that use JSON content format.

This guide is part of the documentation set (shown in Figure 1) for the Meeting Server. These documents can be found on cisco.com.

1.1.1 Interactive API Reference tool

We recently introduced a new interactive API reference tool enabling you to see a high level view of the API objects and drill down to lower levels for the detail. There are also learning labs to help you get started, these will be added to over time. We encourage you to try out this tool; sometime in the future we will discontinue publishing the pdf version of the API Reference Guide.

https://developer.cisco.com/cisco-meeting-server/

Steps to use the tool:

1. Click **View the docs**

2. Select a category from the list in the left pane. For example: Call Related Methods.

3. Click on any method to see URI: GET/POST/PUT. Refer to the table of parameters and response elements with descriptions. For example: GET https://ciscocms.docs.apiary.io/api/v1/calls?

Note: If you are using a POST/PUT methods, the related 'Attributes' with descriptions appear on the right-hand pane when you select the method.

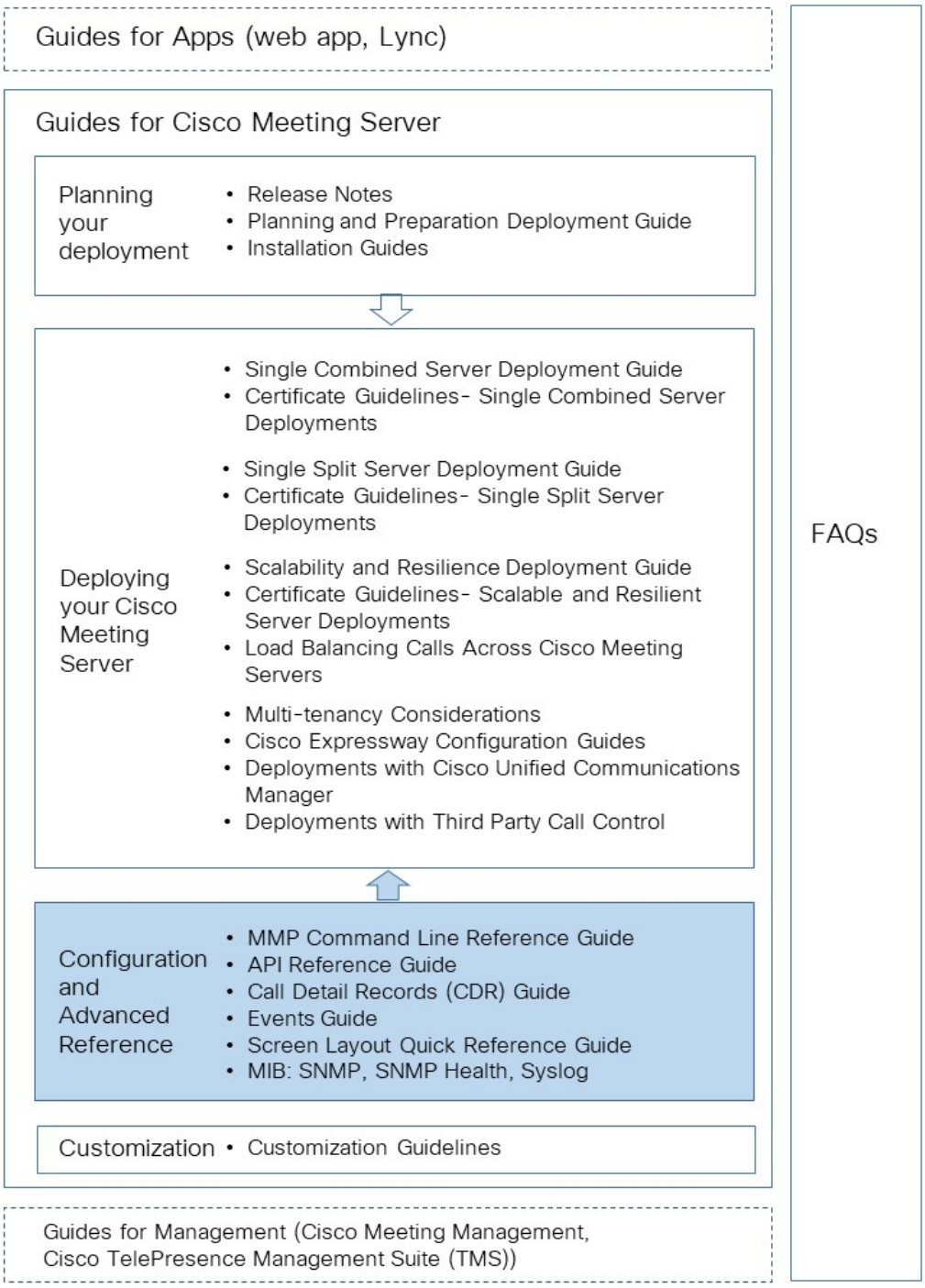
Learning labs

https://learninglabs.cisco.com/modules/cisco-meeting-server

The learning labs are intended as a starting point, covering a broad cross-section of what is possible with the Cisco Meeting Server API. Every learning lab is a step-by-step tutorial which takes you through the steps to complete the task from start to finish.

Example: The 'Setting up host and guest access with Cisco Meeting Server API' provides instructions to configure ways in which users can join meetings in a space with different options.

Figure 1: Cisco Meeting Server documentation for release 3.8



1.2 Summary of 3.8 API additions and changes

API functionality for Meeting Server 3.8 includes the following new API parameters.

New API parameter is added to search callProfile API by name

- **`name`** is introduced on:
  - POST to **`/callProfiles/`**
  - PUT to **`/callProfiles/<call profile id>/`**
  - GET on **`/callProfiles/`**
  - GET on **`/callProfiles/<call profile id>/`**

New API parameter is added to set the maximum resolution on the presenter's screen in web app meeting

- **`contentResolution`** parameter is introduced on:
  - POST, PUT and GET methods on **`/webBridgeProfiles/`**

New API parameter is added to add tags to coSpaces

- **`spaceTag`** parameter is introduced on:
  - POST to **`/coSpaces`**
  - PUT to **`/coSpaces/<coSpace ID>`**
  - GET on **`/coSpaces`**
  - GET on **`/coSpace/<coSpace ID>`**

New API parameter is added to enable the Surveys feature in web app:

- **`surveyAllowed`** is introduced on:
  - POST to **`/calls/`**
  - PUT to **`/calls/call id/`**
  - GET on **`/calls/call id/`**
  - POST to **`/callProfiles`**
  - PUT to **`/callProfiles/<call profile id>`**
  - GET on **`/callProfiles/<call profile id>`**
- **`surveyOpsAllowed`** is introduced on:
  - POST to **`/calls/<call Id>/callLegs`**
  - GET on **`/callLegs/<callLeg ID>`**
  - PUT to **`/callLegs/<callLeg ID>`**
  - POST to **`/calls/<call id>/participants`**
  - POST to **`/callLegProfiles`**

- GET on **/callLegProfiles/<call leg profile id>**
- PUT to **/callLegProfiles/<call leg profile id>**

API parameter enhanced to enable sending chat messages to a particular participant

- **chatAllowed**:
  - POST to **/calls**
  - PUT to **/calls/<call id>**
  - GET on **/calls/<call id>**
  - POST to **/callProfiles**
  - PUT to **/callProfiles/<call profile id>**
  - GET on **/callProfiles/<call profile id>**
- **chatContributionAllowed**:
  - POST to **/calls/<call id>/participants**
  - POST to **/calls/<call id>/callLegs**
  - GET on **/callLegs/<call leg id>**
  - PUT to **/callLegs/<call leg id>**
  - POST to **/callLegProfiles**
  - PUT to **/callLegProfiles/<call leg profile id>**
  - GET on **/callLegProfiles/<call leg profile id>**

1.2.1 Assigning a name to callProfiles API

A new API parameter **name** is introduced to search and retrieve **callProfiles** by its name. The parameter is supported on the following methods:

- POST to **/callProfiles/**
- PUT to **/callProfiles/<call profile id>/**
- GET on **/callProfiles/<call profile id>/**
- GET on **/callProfiles/**

Request parameters	Type/ Value	Description/ Notes
name	String	Name of the **callProfiles**.

1.2.2 Configuring screen resolution for content shared in web app meeting

A new API parameter **contentResolution** is introduced to set the maximum resolution of the presenter's screen, while sharing content in a web app meeting. This parameter is supported

on POST, PUT and GET methods on **/webBridgeProfiles/**.

Request parameters	Type/ Value	Description/ Notes
contentResolution	720p|1080p|**unrestricted**	The maximum resolution of the presenter's screen while sharing content in web app.

1.2.3 Adding tags to coSpaces

A new API parameter **spacetag** is introduced to add tags to coSpaces. This parameter is supported on the following methods:

- POST to **/coSpaces**
- PUT to **/coSpaces/<coSpace ID>**
- GET on **/coSpaces**
- GET on **/coSpace/<coSpace ID>**

Request parameters	Type/ Value	Description/ Notes
spaceTag	String	Name of the tag given to the particular **coSpace**. Tag for each space is unique, case-insensitive and may contain up to ten characters.

1.2.4 Enabling surveys in Meeting Server

A new API parameter **surveyAllowed** is introduced to enable/disable the Survey feature at call level. This parameter is supported on the following methods:

- POST to **/calls/**
- PUT to **/calls/call id/**
- GET on **/calls/call id/**
- POST to **/callProfiles**
- PUT to **/callProfiles/<call profile id>**
- GET on **/callProfiles/<call profile id>**

Request parameters	Type/ Value	Description/ Notes
surveyAllowed	true| **false**	True - Indicates that surveys are allowed in the meeting and participants can take survey. False - Indicates that surveys are not allowed in the meeting. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

Additionally, a new API parameter **surveyOpsAllowed** is introduced to enable/disable the Survey feature at the participant level. The parameter is supported on the following API methods:

- POST to **/calls/<call Id>/callLegs**
- GET on **/callLegs/<callLeg ID>**
- PUT to **/callLegs/<callLeg ID>**
- POST to **/calls/<call id>/participants**
- POST to **/callLegProfiles**
- GET on **/callLegProfiles/<call leg profile id>**
- PUT to **/callLegProfiles/<call leg profile id>**

Request parameters	Type/ Value	Description/ Notes
surveyOpsAllowed	true| **false**	True – Indicates that the participant can create/launch/delete/view results of surveys. False – Indicates that the participant cannot create/launch/delete/view results of surveys. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

1.2.5 Support to send chat messages to a particular participant in a web app meeting

API parameters **chatAllowed** and **chatContributionAllowed** are modified to support sending chat messages to a particular participant in a web app meeting. The parameters are supported on the following methods:

- **chatAllowed**:
  - POST to **/calls**
  - PUT to **/calls/<call id>**
  - GET on **/calls/<call id>**
  - POST to **/callProfiles**
  - PUT to **/callProfiles/<call profile id>**
  - GET on **/callProfiles/<call profile id>**

Request parameters	Type/ Value	Description/ Notes
chatAllowed	**true**| false	If the value is specified, it determines whether or not chat is allowed on this call/s using this call profile.

- **chatContributionAllowed**:
  - POST to **/calls/<call id>/callLegs**
  - GET on **/callLegs/<call leg id>**
  - PUT to **/callLegs/<call leg id>**
  - POST to **/calls/<call id>/participants**
  - POST to **/callLegProfiles**
  - PUT to **/callLegProfiles/<call leg profile id>**
  - GET on **/callLegProfiles/<call leg profile id>**

Request parameters	Type/ Value	Description/ Notes
chatContributionAllowed	**true**|false	If the value is specified, it determines whether or not this call leg/this participant/call legs using this **callLegProfile** are allowed to send messages on the chat.

2 General Structure of Objects

The Meeting Server‘s Application Programming Interface (API) is designed as a hierarchy of objects, similar to the directory structure. As an example, each configured coSpace exists as a node in this tree, and all of the users who are members of the coSpace exist as nodes “beneath” the coSpace object’s node. The API objects are accessed using a suitable REST client, see Section 3.4.

Note: Although the Cisco Meeting App and other Cisco Meeting Server guides refer to " spaces" rather than " coSpaces" , the API still uses /coSpace objects.

The Meeting Server has the potential to host a large number of active calls and coSpaces. To reduce the overhead of retrieving the entire collection of objects in a single response, responses typically return the first “N” matching entries and a count of the total number of objects of that type. To find an individual object’s active status, or to modify or delete it, use filters on the initial retrieval in order to identify the object in question. Refer to Section 4.2 for more information on filters and the GET command.

Note: Some nodes in the tree refer to sets of objects shared across the whole cluster (for example, the set of configured coSpaces or dial plan rules) whereas others (for example, the set of alarm conditions under " api/v1/system/alarms") are specific to the Meeting Server being accessed.

2.1 Object Hierarchy

In each case, the top level plural term sits above potentially many individual object nodes; these individual object nodes are identified by an <ID> which is a GUID, typically. For example, if there are 100 coSpaces configured on the Meeting Server, conceptually there would be 100 nodes directly beneath /coSpaces in the hierarchy.

The hierarchy of objects addressable via the Call Bridge API is:

```
/accessQuery

/callBrandingProfiles
/callBrandingProfiles/<call branding profile id>

/callBridges
/callBridges/<call bridge id>
/callBridgeGroups (2.1 onwards)
/callBridgeGroups/<call bridge group id> (2.1 onwards)

/calls
/calls/<call id>
/calls/<call id>/callLegs
```

```
/calls/<call id>/diagnostics
/calls/<call id>/participants
/calls/<call id>/participants/* (2.2 onwards)
/calls/<call id>/captions (3.4 onwards)

/callProfiles
/callProfiles/<call profile id>

/callLegs
/callLegs/<callLeg id>
/callLegs/<callLeg id>/callLegProfileTrace
/callLegs/<call leg id>/cameraControl (2.9 onwards)
/callLegs/<call leg id>/generateKeyframe (2.2 onwards)
/callLegs/<call leg id>/snapshot (3.5 onwards)

/callLegProfiles
/callLegProfiles/<call leg profile id>
/callLegProfiles/<call leg profile id>/usage

/clusterLicensing(3.0 onwards)
/clusterLicensing/raw (3.0 onwards)

/compatibilityProfiles (2.1 onwards)
/compatibilityProfiles/<compatibility profile id> (2.1 onwards)

/lobbyProfiles (3.7 onwards)
/lobbyProfiles/<lobby profile id> (3.7 onwards)

/conversationIdQuery (2.3 onwards)

/cospaceBulkParameterSets(2.0 onwards)
/cospaceBulkParameterSets/<coSpace bulk parameter set id> (2.0 onwards)
/cospaceBulkSyncs(2.0 onwards)
/cospaceBulkSyncs/<coSpace bulk sync id>(2.0 onwards)

/coSpaces
/coSpaces/<coSpace id>
/coSpaces/<coSpace id>/emailInvitation (3.2 onwards)
/coSpaces/<coSpace id>/accessMethods
/coSpaces/<coSpace id>/accessMethods/<access method id>
/coSpaces/<coSpace id>/accessMethods/<access method id>/emailInvitation (3.2
onwards)
/coSpaces/<coSpace id>/coSpaceUsers
/coSpaces/<coSpace id>/coSpaceUsers/<coSpaceUser id>
/coSpaces/<coSpace id>/diagnostics
/coSpaces/<coSpace id>/messages (3.0 removed)
/coSpaces/<coSpace id>/metadata (3.2 onwards)

/coSpaceTemplates (2.9 onwards)
/coSpaceTemplates/<coSpace template id> (2.9 onwards)
/coSpaceTemplates/<coSpace template id>/accessMethodTemplates (2.9 onwards)
/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method
```

```
template id> (2.9 onwards)

/dialInSecurityProfiles(3.0 onwards)
/dialInSecurityProfiles/<dial in security profile id> (3.0 onwards)

/dialTransforms
/dialTransforms/<dial transform id>

/directorySearchLocations
/directorySearchLocations/<directory search location id>

/dtmfProfiles
/dtmfProfiles/<dtmf profile id>

/forwardingDialPlanRules
/forwardingDialPlanRules/<forwarding dial plan rule id>

/inboundDialPlanRules
/inboundDialPlanRules/<inbound dial plan rule id>

/ivrs
/ivrs/<ivr id>
/ivrBrandingProfiles
/ivrBrandingProfiles/<ivr branding profile id>

/layoutTemplates
/layoutTemplates/<layout template id>
/layoutTemplates/<layout template id>/template (2.8 onwards)

/ldapMappings
/ldapMappings/<ldap mapping id>
/ldapServers
/ldapServers/<ldap server id>
/ldapSources
/ldapSources/<ldap source id>
/ldapSyncs
/ldapSyncs/<ldap sync id>
/ldapUserCoSpaceTemplateSources(2.9 onwards)
/ldapUserCoSpaceTemplateSources/<ldap user coSpace template source id>
 (2.9 onwards)
/ldapUserProvisionedCoSpaceMappings (3.1 onwards)
/ldapUserProvisionedCoSpaceMappings/<LDAP user provisioned coSpace mapping id>
(3.1 onwards)
/ldapUserProvisionedCoSpaceSources (3.1 onwards)
/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace mapping id>
(3.1 onwards)
/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace source id>
(3.1 onwards)

/outboundDialPlanRules
/outboundDialPlanRules/<outbound dial plan rule id>
```

```
/participants
/participants/<participant Id>
/participants/<participantId>/callLegs

/recorders (3.0 removed)
/recorders/<recorder id> (3.0 removed)
/recorders/<recorder id>/status (2.2 onwards) (3.0 removed)

/streamers (2.1 onwards) (3.0 removed)
/streamers/<streamer id> (2.1 onwards) (3.0 removed)
/streamers/<streamer id>/status (2.2 onwards) (3.0 removed)

/system/alarms
/system/cdrReceiver (deprecated)
/system/cdrReceivers
/system/cdrReceivers/<CDR receiver id>
/system/configuration/cluster
/system/configuration/xmpp (3.0 removed)
/system/database
/system/diagnostics
/system/diagnostics/<diagnostics id>
/system/diagnostics/<diagnostics id>/contents
/system/licensing (2.0 onwards)
/system/load (2.1 onwards)
/system/MPLicenseUsage (2.6 onwards)
/system/MPLicenseUsage/knownHosts  (2.6 onwards)
/system/multipartyLicensing (2.0 onwards)
/system/multipartyLicensing/activePersonalLicenses(2.0 onwards)
/system/profiles
/system/profiles/effectiveWebBridgeProfile (3.0 onwards)
/system/status
/system/timedLogging (3.2 onwards)


/tenantGroups
/tenantGroups/<tenant group id>

/tenants
/tenants/<tenant id>
/tenants/<tenant id>/effectiveWebBridgeProfile (3.0 onwards)

/turnServers
/turnServers/<turn server id>
/turnServers/<turn server id>/status

/uriUsageQuery (2.3 onwards)
/users
/users/<user id>
/users/<user id>/usercoSpaces
/users/<user id>/userCoSpaceTemplates (2.9 onwards)
/users/<user id>/userCoSpaceTemplates/<user coSpace Template id>(2.9 onwards)
```

```
/users/<user id>/userProvisionedCoSpaces (3.1 onwards)
/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id> (3.1
onwards)
/userProfiles
/userProfiles/<user profile id>

/webBridges
/webBridges/<web bridge id>
/webBridges/<web bridge id>/status (2.2 onwards)
/webBridges/<web bridge id>/updateCustomization

/webBridgeProfiles(3.0 onwards)
/webBridgeProfiles/<web bridge profile id> (3.0 onwards)
/webBridges/<web bridge id>/effectiveWebBridgeProfile
 (3.0 onwards)
/webBridgeProfiles/<web bridge profile id>/ivrNumbers (3.1 onwards)
/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id> (3.1
onwards)
/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses (3.1 onwards)
/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge
address id> (3.1 onwards)
```

3 Accessing the API

The API uses HTTPS as a transport mechanism.

Note: The time taken for API requests can vary depending upon factors, including but not limited to, the request type, number of outstanding requests, database size, server loading, latency between API client and the Call Bridge receiving API requests, and latency between the Call Bridge receiving the API request and the primary database. We recommend that when developing applications, you test API performance on a representative system.

Once you have carried out Configuration Settings and Authentication below you can then access and work with the API in one of the following ways:

- API access on the Meeting Server web interface: Meeting Server web interface provides an API explorer for Call Bridge APIs
- using third party tools: Third party tools can provide access to Call Bridge as well as Scheduler APIs.

3.1 Configuration Settings

To use the API, you need to connect via HTTPS via the same TCP ports as you would use to access the Web Admin Interface – typically port 443; that is, they use the same interface.

You also need to configure a username and password: you must provide these credentials in order to use the API. Set them using the MMP command **`user add <username> (admin|crypto|audit|appadmin|api)`**. This command prompts for the user's password; see the MMP Command Reference Guide for details.

3.2 Authentication

The API user supplies a shared secret username and password to the Meeting Server configured with the same username and password. The username and password are set in the MMP command line.

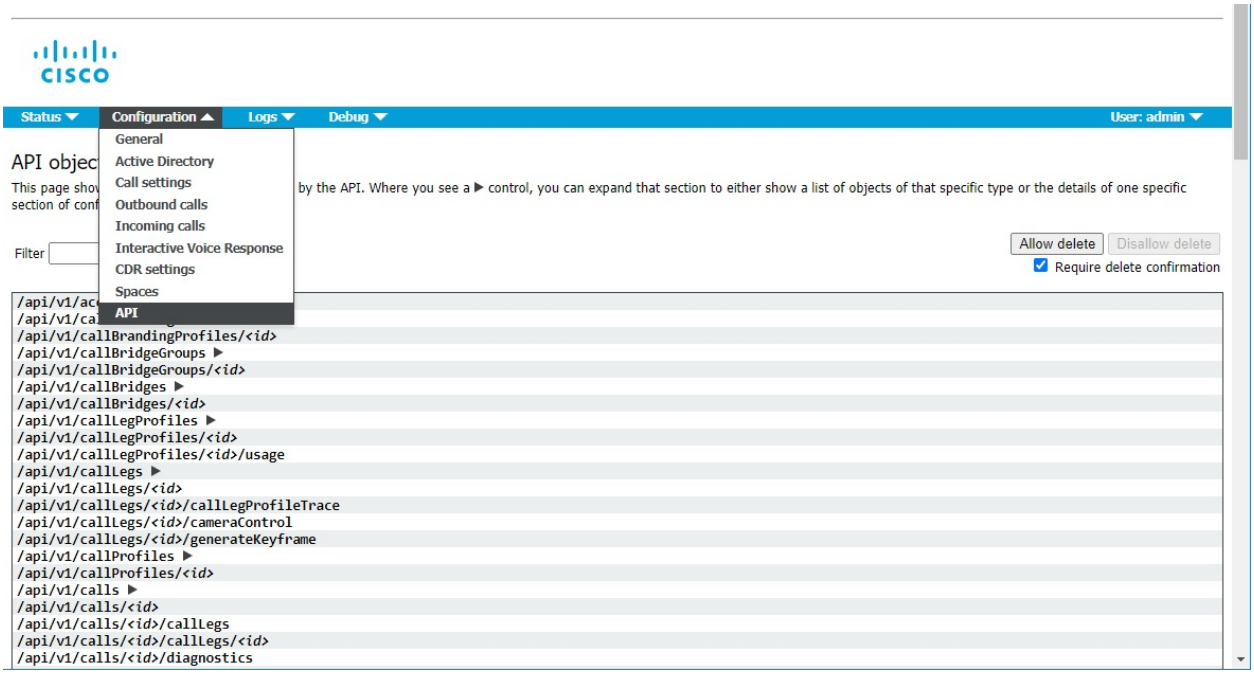
While the authentication credentials are sent in essentially plain text within the HTTP traffic, by using HTTPS the traffic itself cannot be read by an external party.

3.3 API access on the web interface

To simplify using the Call Bridge API without the need for third-party applications, version 2.9 introduced a user interface for the Call Bridge API that can be accessed via the **Configuration** tab of the Meeting Server web interface, as shown in Figure 2.

The Scheduler APIs introduced in version 3.3 are not supported via this interface. See Scheduler APIs.

Figure 2: Accessing the Call Bridge API via the Meeting Server web interface



Note: If you wish to delete any configured API objects, select **Allow delete** on the right-hand side of the screen. By default, deletion is disallowed and **Require delete confirmation** is checked to help prevent unintentional deletions.

3.4 Third Party Tools to Use

You can interact with the Meeting Server API using any software that can generate properly formatted HTTPS requests, such as: custom software, pre-built libraries, command line tools, or GUI based web/REST utilities. For RESTful APIs there are now ready-made tools available online to simplify interacting with REST APIs.

Examples of some popular third party tools that are known to work with Meeting Server include:

- Postman – This is a graphical utility for Windows, OSX, and Linux with many development tools and robust request/response tools. Install from https://www.getpostman.com/postman.
  Additionally, there is community content available to simplify Meeting Server API use with Postman available at: https://github.com/ciscocms/cms-postman-collection. This is a community supported pre-built Postman collection for Meeting Server API. It includes all

the Meeting Server commands and properties pre-populated to allow you to quickly get up and running using the Meeting Server API.

- RESTer – This is a lightweight browser-based REST tool for Google Chrome that supports advanced features such as variables, saving requests and histories. Install from Google Chrome Web Store (https://chrome.google.com/webstore/category/extensions). For information on installing and examples of using RESTer, see Appendix D.
- RESTED – This is an easy-to-use, lightweight browser-based REST tool for Mozilla Firefox that supports advanced features such as variables, saving requests and histories. Install from the **Add-ons...** option in Firefox.

If you prefer command line or cannot install third party software in your environment, popular tool examples include:

- curl – Command line web request utility available in most OSX and Linux installations. Can also be installed on Microsoft Windows.
- Powershell – Microsoft's command line scripting tool. Available by default on all modern Windows installations and easily scriptable for more complicated response parsing.

3.4.1 Tips for using third-party API Tools

- Make sure you use **`https://`** URLs and include the *port* your webadmin is listening on, not the webbridge.
- Most tools default to verifying SSL/TLS connections and will refuse to connect to Meeting Server if the certificate verification fails. If your URL does not match your certificate or is untrusted, locate the setting to ignore SSL checks in your tools or add an exception for your API URL. Browser-based clients usually follow the trusted/exception list of the browser they are running in.
- Make sure you have enabled authentication and **Basic Auth** in your requests.
- When using POST and PUT methods, make sure you use the request **Body** and **Content-Type: x-www-form-urlencoded**
- The Meeting Server API will ignore properties in PUT/POST actions that are misspelled and not return an HTTP error. Always check with a follow-up GET to confirm that your changes are implemented as expected.
- Remember that large result sets may require paging and using the offset option to see entries beyond the first few entry results.

- Most tools default to verifying SSL/TLS connections and will refuse to connect to Meeting Server if the certificate verification fails. If your URL does not match your certificate or is untrusted, locate the setting to ignore SSL checks in your tools or add an exception for

your API URL. Browser-based clients usually follow the trusted/exception list of the browser they are running in.

- Make sure you have enabled authentication and **Basic Auth** in your requests.
- When using POST and PUT methods, make sure you use the request **Body** and **Content-Type: x-www-form-urlencoded**
- The Meeting Server API will ignore properties in PUT/POST actions that are misspelled and not return an HTTP error. Always check with a follow-up GET to confirm that your changes are implemented as expected.
- Remember that large result sets may require paging and using the offset option to see entries beyond the first few entry results.

4 API Methods

There are four methods:

- GET – retrieves existing information
- POST – creates new objects in the hierarchy
- PUT – modifies an existing object
- DELETE – destroys an object in the tree

These methods are described in more detail below. However, it is important to firstly understand the URL format used to communicate these methods to the API.

4.1 URL format

To address or create individual objects, the URL format mirrors the conceptual hierarchy of objects, with some additional preceding tags to identify that the request is for the API. For example, to retrieve information on API object "/calls/dbfca0dd-dbe1-43bb-b101-beb9a7ef35f4" you need to issue:

```
GET /api/v1/calls/dbfca0dd-dbe1-43bb-b101-beb9a7ef35f4 HTTP/1.1
```

This string is structured as follows: at the top level, including "/api" means that the on-board HTTP server process can distinguish this HTTP method from a normal browser method, and including "v1" means the API handler knows that the request is being made by an object that understands version 1 of the API.

If an API method is successful, it gives a "200 OK" response from the Meeting Server. If an error occurs, the Meeting Server responds with a 4xx or 5xx HTTP status code.

A 503 ("Service Unavailable") status code is returned for API calls unable to be serviced due to a temporary "busy" condition on the Meeting Server – this can indicate that it would be useful to re-attempt the same method later.

A request supplying a <coSpace id> which does not correspond to a valid coSpace object gives "404 Not Found" response.

For 4xx and 5xx error cases, extended error information may be returned, for example:

```
<failureDetails>
<coSpaceDoesNotExist/>
</failureDetails>
```

In Call Bridge API, information is returned as "text/xml" body data, whereas this returned in JSON format in Scheduler APIs.

Generally, such a response consists of a "failureDetails" section and a list of errors. In the above example a method was attempted using a coSpace ID that did not match an active coSpace. Possible failure reasons are described in Section 4.6.

4.2 GET Methods

GET methods allow retrieval of information about existing API-accessible objects, and are used at two levels: Collections level and Individual object level.

4.2.1 Collections level

If the GET method is performed at the Collections level (the pluralized noun: “calls”, “coSpaces” and so on) then multiple matching child nodes will be retrieved. By design this is not guaranteed to be the entire list, but the total number of objects of that type present in the Meeting Server can be learned via this mechanism.

To retrieve just specific items, most GET methods at the Collections level allow the use of a filter expression. So the interface of a management tool initially presents the API user with the summed count of coSpaces (for example), basic details on the first “N” coSpaces (e.g. their names) and a filter box that the human user can use to search for the specific coSpace(s) of interest.

With no other additional parameters, a GET method at the Collections level will return items from the start of the Meeting Server's notional complete list. By comparing the number of items returned with the " total" value, it is straightforward to determine whether all elements have been returned (if the number of elements returned is equal to the " total" value).

4.2.2 Using limit and offset at the Collections level

The number of elements returned can be restricted to a limit chosen by the requestor, by including a " limit=<limitValue>" in the API request. This guarantees that no more than the specified " limitValue" number of elements will be returned – the Meeting Server will have its own limit in these cases too, and therefore the number of elements returned will be the lower of any supplied " limitValue" and the Meeting Server's own limit.

To retrieve elements other than the first " n" on the Meeting Server's notional list, it is also possible to supply an " offset=<offsetValue>" in the API request. This causes the Meeting Server to return elements that start at the specified position in its list, skipping the first " offsetValue" number of elements. If " offsetValue" is greater than the number of objects of that type, then no elements will be returned.

Note: The offset value should not be seen as a general mechanism for retrieving a large complete list. Sequential retrievals of one " page" of data followed by a second " page" will not necessarily be operating on the same complete list if any objects have been deleted or added in the interval between these methods.

The expectation is that, for each request and response, the requester will keep track of the offset and limit values used, and combine this knowledge with the number of elements returned in the response and the " total" indicated by the Meeting Server. If the " offsetValue" supplied by the requestor plus the number of elements returned is less than the " total" value indicated

in the response, the requestor then knows that there are more values present. The following table shows some examples:

Table 1: Examples of requestor offset/limits

Requestor offset	Requestor limit	XML Response	Meaning
<not supplied>	<not supplied>	<coSpaces total=" 7" > <cospace /> x 7 </coSpaces>	All coSpaces (0 – 6) present in response
<not supplied>	1	<coSpaces total=" 7" > <cospace /> x 1 </coSpaces>	First coSpace present in response
4	10	<coSpaces total=" 7" > <cospace /> x 3 </coSpaces>	coSpaces 4 – 6 present in response
0	100	<coSpaces total=" 53" > <cospace /> x 10 </coSpaces>	Meeting Server limits its response to its internal limit of 20 despite requestor allowing up to 100
20	10	<coSpaces total=" 53" > <cospace /> x 10 </coSpaces>	Meeting Server accepts requestor offset and limit, returning spaces 20..29

Note: The following GET limits apply:

- /api/v1/coSpaces – 20
- /api/v1/calls – 10

4.2.3 Individual object level

If the GET method is performed at the Individual object level, full information about just that one object will be returned. For example, after the unique ID of a coSpace has been learned via a (potentially filtered) GET of the "/coSpaces" node, a subsequent GET of the "/coSpaces/<coSpace id>" node would return expanded information about just that one coSpace.

4.2.4 HTTP specifics

GET methods contain the complete node location and any parameters specific to the retrieval being performed in the URI supplied by the API user. For example, to retrieve basic information

on the first “N” coSpaces, the URI would be:

/api/v1/coSpaces

whereas to list just those whose name includes “sales”, the GET would be performed on:

/api/v1/coSpaces?filter=sales

If a GET method has been successful and yields a “200 OK” response, the Meeting Server returns the retrieved information as “text/xml/json (for Scheduler APIs)” body data.

4.2.5 How GET methods are detailed in this document

For each GET method at the Collections level the following information is provided:

- The node it operates on
- A table of form parameters, such as filter, offset and limit mentioned above, some of which may be optional. Mandatory parameters are marked with an asterisk (*)
- A table showing the returned information

Both tables show the format of the parameter (e.g. ID or string) or the possible values (e.g. true|false)

For each GET method at the Individual level the following information is provided:

- The node it operates on
- A table showing the returned information

The form parameters are those for the Collections level, unless otherwise indicated.

4.3 POST Methods

POST methods create new objects; for example, to create a new configured coSpace or dial plan rule. Using a POST method to create a new call leg associated with a coSpace is the way to make a new outbound SIP connection.

4.3.1 HTTP specifics

Most POST methods require some parameters to be supplied: for example, creation of a coSpace requires the new coSpace’s name to be specified, and a new call leg can only be created if the remote party’s address is known. Such parameters must be supplied by the initiator of the POST method via the standard HTTP “x-www-form-urlencoded” format, as used by “<form>” elements in an HTML document.

If a POST method has been successful in adding a new object to the hierarchy, that object’s id, and its position within the hierarchy are returned in the “Location” field of the response.

4.3.2 How POST methods are detailed in this document

For each POST method the following information is provided:

- The node it operates on
- A table of form parameters, some of which may be optional. Mandatory parameters are marked with an asterisk (*)
- The format of each parameter (e.g. ID or string) or the possible values (e.g. true|false). If appropriate the default value of a parameter (the value used if you do not specify a parameter) is shown in bold e.g. **true**|false.

4.4 PUT Methods

PUT methods modify existing objects; for example, changing the name of a coSpace, muting a specific call leg or changing the layout.

In general, when using PUT in an object:

- omit a parameter to leave its value unchanged
- use a parameter with a new value to change to this value. Supply an empty value to unset a value. For example, to remove a tenant association from a coSpace, modify that coSpace with a parameter set including “tenant=”.

4.4.1 HTTP specifics

Parameters for a request must be supplied in “x-www-form-urlencoded” format.

4.4.2 How PUT methods are detailed in this document

Each PUT method is in the same section as the POST method for the same object e.g. creating and modifying a coSpace are dealt with together. Form parameters for modifying an object (PUT) are only noted if they differ from the POST method; for example, for callLegs.

4.5 DELETE Methods

A DELETE method removes an individual object from the hierarchy; for example, disconnecting a call leg or disassociating a user from a coSpace so that the user is no longer a member.

Therefore the DELETE method is typically performed at the Individual level e.g. DELETE on /api/v1/coSpace/<id>/accessMethods/<id>

The object’s ID is known either from a previous retrieval (GET) method at the Collections level or from the “Location” field in the response to a previous creation (PUT) method. (coSpace can be deleted at the Collections level.)

If the object is removed successfully, the Meeting Server sends a “200 OK” response.

Because of the relative simplicity of this method, it is not detailed elsewhere in this document – with the exception of deleting chat messages and deleting meetings and occurences for Scheduler APIs.

4.6 Failure Reasons

The following " failureDetails" codes can be returned by the API for any of the above methods, in response to a user error:

```
<failureDetails>
<tenantDoesNotExist />
</failureDetails>
```

Note: The error codes described in this section are specific to Call Bridge APIs. See Section 15.4 for Scheduler response codes.

Reason code	Description
accessMethodDoesNotExist	You tried to modify or remove an accessMethod using an ID that did not correspond to a valid access method. You tried to create a call leg or participant with an access method that does not correspond to the call's coSpace. (from version 3.2)
callBrandingProfileDoesNotExist	You tried to modify or remove a call branding profile using an ID that did not correspond to a valid call branding profile.
callBridgeDoesNotExist	You tried to modify or remove a configured clustered Call Bridge using an ID that did not correspond to a valid clustered Call Bridge.
callBridgeGroupDoesNotExist	You tried to modify, remove or use a Call Bridge group using an ID that did not correspond to a valid Call Bridge group (from version 2.1).
callBridgeGroupUnavailable	You tried to create a participant on a Call Bridge Group that is unavailable or could not accept the call (from version 2.2).
callBridgeUnavailable	You tried to create a participant on a Call Bridge that is unavailable or could not accept the call (from version 2.2).
callDoesNotExist	You tried to perform a method on a call object using an ID that did not correspond to a currently active call.
callRecordingCannotBeModified	You tried to start/stop recording a call that cannot be modified.

Reason code	Description
callStreamingCannotBeModified	You tried to start/stop streaming a call that cannot be modified (from version 2.1).
callLegCannotBeDeleted	You tried to delete a call leg that can't be deleted.
callLegDoesNotExist	You tried to perform a method on a call leg object using an ID that did not correspond to a currently active call leg.
callLegProfileDoesNotExist	You tried to modify or remove a callLegProfile using an ID that did not correspond to a valid call leg profile.
callProfileDoesNotExist	You tried to modify or remove a callProfile using an ID that is not valid.
cdrReceiverDoesNotExist	You tried to modify or remove a CDR receiver using an ID that did not correspond to a valid CDR receiver.
coSpaceAccessMethodTemplateDoesNotExist	You tried to modify, remove or retrieve a coSpace access method template using an ID that did not correspond to a valid coSpace access method template on the system (from version 2.9).
coSpaceCallDoesNotExist	You tried to create a call leg or participant (with an access method specified) on a call that is not associated with a coSpace. (from version 3.2)
coSpaceDoesNotExist	You tried to modify or remove a coSpace using an ID that did not correspond to a valid coSpace on the system
coSpaceTemplateDoesNotExist	You tried to modify, remove or retrieve a coSpace template using an ID that did not correspond to a valid coSpace template on the system (from version 2.9).
coSpaceUserDoesNotExist	You tried to modify or remove a coSpace user using an ID that did not correspond to a valid coSpace user.
databaseNotReady	You tried a method (e.g. initiation of an LDAP sync method) before the database was ready
dialInSecurityProfileDoesNotExist	You tried to modify, remove, or retrieve a dial-in security profile using an ID that did not correspond to a valid dial-in security profile. (3.0 onwards)
directorySearchLocationDoesNotExist	You tried to reference, modify or remove a directory search location using an ID that did not correspond to a valid directory search location.
dtmfProfileDoesNotExist	You tried to reference, modify or remove a DTMF profile using an ID that did not correspond to a valid DTMF profile.

Reason code	Description
duplicateCallBridgeName	You tried to create or modify a clustered Call Bridge to use a name that would clash with an existing configured clustered Call Bridge.
duplicateCoSpaceId	You tried to create or modify a coSpace call ID to use a call ID that clashed with one used by another coSpace.
duplicateCoSpaceUri	You tried to create or modify a coSpace to use a URI that clashed with one that corresponds to another coSpace. (Two coSpaces can't share the same URI, because the Meeting Server must be able to uniquely resolve an incoming call to a coSpace URI).
duplicateCoSpaceIdPasscode	You tried to modify a coSpace, or create or modify a coSpace access method, using a call ID/passcode combination that clashed with another call ID/passcode that is already used by that coSpace or one of its access methods.
duplicateCoSpaceUriPasscode	You tried to modify a coSpace, or create or modify a coSpace access method, using a URI/passcode combination that clashed with URI/passcode combination that is already used by that coSpace or one of its access methods.
duplicateCoSpaceSecret	You tried to modify a coSpace, or create or modify a coSpace access method, using a secret that clashed with one that is already used by that coSpace or one of its access methods.
duplicateUserCoSpaceTemplate	You tried to assign the same coSpace template to a user for a second time (from version 2.9).
forwardingDialPlanRuleDoesNotExist	You tried to modify or remove an forwarding dial plan rule using an ID that did not correspond to a valid forwarding dial plan rule.
inboundDialPlanRuleDoesNotExist	You tried to modify or remove an inbound dial plan rule using an ID that did not correspond to a valid inbound dial plan rule.
inboundDialPlanRuleUriConflict	You tried to make modifications to an inbound dial plan rule which would have caused a URI conflict. For example, this can happen if you try to add a rule which matches multiple tenants and more than one tenant has a coSpace with the same URI.

Reason code	Description
invalidOperation	You tried an operation which isn't supported; for example, you attempted to POST to /api/v1/system/profiles or issue a DELETE for a configured user generated from an LDAP sync.
invalidValue	You entered an empty string or an invalid character as a language parameter.
invalidVersion	You attempted an operation with an invalid API version.
ivrBrandingProfileDoesNotExist	You tried to modify or remove an IVR branding profile object using an ID that did not correspond to a valid IVR branding profile on the system.
ivrDoesNotExist	You tried to modify or remove an IVR object using an ID that did not correspond to a valid IVR on the system.
ivrNumberDoesNotExist	You tried to modify, remove, or retrieve an IVR number using an ID that did not correspond to a valid IVR number (from version 3.1).
ivrUriConflict	You tried to make modifications to an IVR object which would have caused a URI conflict.
layoutTemplateDoesNotExist	You tried to modify, remove or retrieve a layout template using an ID that did not correspond to a valid layout template on the system (from version 2.8).
layoutTemplateDescriptionTooLong	You tried to set a layout template description that exceeds the allowed size (from version 2.8).
ldapMappingDoesNotExist	You tried to modify or remove an LDAP mapping using an ID that did not correspond to a valid LDAP mapping.
ldapServerDoesNotExist	You tried to modify or remove an LDAP server using an ID that did not correspond to a valid LDAP server.
ldapSourceDoesNotExist	You tried to modify or remove an LDAP source using an ID that did not correspond to a valid LDAP source.
ldapSyncCannotBeCancelled	You tried to cancel an LDAP synchronization that has either started or completed – only LDAP synchronization methods that have not started yet can be canceled.
ldapSyncDoesNotExist	You tried to query or cancel an LDAP synchronization with an ID that did not correspond to a valid LDAP synchronization.
ldapUserCoSpaceTemplateSourceDoesNotExist	You tried to remove or retrieve using an ID that did not correspond to an existing LDAP user coSpace template source entry (from version 2.9).

Reason code	Description
ldapUserProvisionedCoSpaceMappingDoesNotExist	You tried to modify or remove an LDAP user provisioned coSpace mapping using an ID that did not correspond to a valid LDAP user provisioned coSpace mapping (from version 3.1).
ldapUserProvisionedCoSpaceSourceDoesNotExist	You tried to modify or remove an LDAP user provisioned coSpace source using an ID that did not correspond to a valid LDAP user provisioned coSpace source (from version 3.1).
loadBalancingDisabled	You tried to create a participant in a Call Bridge Group with load balancing for outgoing calls disabled (from version 2.2).
maxNumberOfWebBridgeAddressesReached	You tried to add a new web bridge address for a web bridge profile that already had the maximum number of allowed entries defined. Please remove one to be able to add another one (from version 3.1).
maxNumberOfIvrNumbersReached	You tried to add a new IVR number for a web bridge profile that already had the maximum number of allowed entries defined. Please remove one to be able to add another one (from version 3.1).
messageDoesNotExist	You tried to remove a coSpace message using an ID that did not correspond to a valid coSpace message.
outboundDialPlanRuleDoesNotExist	You tried to modify or remove an outbound dial plan rule using an ID that did not correspond to a valid outbound dial plan rule
parameterError	One or more parameters in a request were found to be invalid. Supporting parameter and error values give more detail about the failure.
participantCannotBeDeleted	You tried to delete a participant that could not be deleted, for instance a remotely-hosted participant.
participantCannotBeModified	You tried to modify a participant that could not be modified, for instance a participant hosted on a remote deployment.
participantLimitReached	You tried to add a new participant beyond the maximum number allowed for the call.
passcodeTooShort	You tried to set a passcode to a coSpace or a coSpace access method but its length is not compliant with the minimum allowed passcode length as specified in the effective dial-in security profile. (3.0 onwards)

Reason code	Description
recorderDoesNotExist	You tried to modify or remove a recorder using an ID that did not correspond to a valid recorder.
recordingNotAllowedByLicensing	You tried to start recording without having the correct license. (Previously recordingLimitReached) (3.0 onwards)
retryAfter	You tried to retrieve the text based meeting entry information when the server was busy or was fetching externally hosted template. Retry later or retry after recommended **`retryAfter`** period in seconds.
streamerDoesNotExist	You tried to modify or remove a streamer using an ID that did not correspond to a valid streamer (from version 2.1).
streamingNotAllowedByLicensing	You tried to start streaming without having the correct license. (Previously streamingLimitReached) (3.0 onwards)
tenantDoesNotExist	You tried to modify or remove a tenant using an ID that did not correspond to a valid tenant
tenantGroupCoSpaceIdConflict	Your request to remove or use a tenant group would have resulted in a coSpace ID conflict.
tenantGroupDoesNotExist	You tried to modify, remove or use a tenant group that does not exist.
tenantParticipantLimitReached	You tried to add a new participant beyond the maximum number allowed for the owning tenant.
tooManyCdrReceivers	You tried to add a new CDR receiver when the maximum number were already present. Currently, up to 2 CDR receivers are supported.
tooManyLdapSyncs	A method to create a new LDAP synchronization method failed. Try again later. Note that the ldapSyncs table is limited to 10 entries. If the error persists, try deleting older completed ldapSyncs.
unrecognisedObject	There are elements in the URI you are accessing that are not recognized; for example, you tried to perform a GET on /api/v1/system/profile rather than (the correct) /api/v1/system/profiles
userCoSpaceTemplateDeletionProhibited	You tried to withdraw an auto-generated coSpace template from a user and this is not allowed (from version 2.9).

Reason code	Description
userCoSpaceTemplateDoesNotExist	You tried to modify, remove or retrieve a user coSpace template using an ID that did not correspond to a valid user coSpace template for that user (from version 2.9).
userDoesNotExist	You tried to modify or remove a user using an ID that did not correspond to a valid user.
userProfileDoesNotExist	You tried to modify a user profile using an ID that did not correspond to a valid user profile.
valueTooLong	You entered too long language parameter.
userProvisionedCoSpaceDoesNotExist	You tried to instantiate, modify, remove or retrieve a user provisioned coSpace using an ID that did not correspond to a valid user provisioned coSpace for that user (from version 3.1).
webBridgeAddressDoesNotExist	You tried to modify, remove, or retrieve a web bridge address using an ID that did not correspond to a valid web bridge address (from version 3.1).
webBridgeProfileDoesNotExist	You tried to modify, remove, or retrieve a web bridge profile using an ID that did not correspond to a valid web bridge profile. (3.0 onwards)

5 Example Requests and Responses for Specific Methods

5.1 Retrieval of Current Active Calls

As described in Section 4.2, retrieval methods using GET involve no body content posted by the retriever. If the request is valid, the Meeting Server returns XML response data.

Request:

```
GET /api/v1/calls HTTP/1.1\r\n
Host: test.example.com\r\n
User-Agent: API console\r\n
Connection: keep-alive\r\n
Authorization: Basic Ym9iOmJ1aWxkZXI=\r\n
\r\n
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 187
Connection: close
\r\n
<?xml version="1.0"?>
<calls total="1">
<call id="527089d6-6581-4331-8417-971c05c9e274">
<name>Sales coSpace</name>
<coSpace>2dcf2b7a-3410-4066-b638-46273698d469</coSpace>
</call>
</calls>
```

5.2 Instantiating a new call and connecting a participant to it

As described above, any parameters needed for the creation method (in this case, the address of the remote party), need to be supplied by the issuer as form data. If the request is successful, details about the new object are returned by the Meeting Server in the “Location” header field.

1. Use a filtered coSpace enumeration to find " Development Team" :

   Request:

   ```
   GET /api/v1/coSpaces?filter=Development%20Team HTTP/1.1\r\n
   Host: 127.0.0.1\r\n
   \r\n
   ```

Response:

```
HTTP/1.1 200 OK\r\n
Content-Type: text/xml\r\n
Content-Length: 197\r\n
Connection: close\r\n
\r\n
<?xml version="1.0"?>
<coSpaces total="1">
<coSpace id="581caae0-420a-43df-9a9e-f690c70e12d3"><name>Development
Team</name><autoGenerated>false</autoGenerated><uri>dev_
team</uri></coSpace>
</coSpaces>
```

2. Matching coSpace 581caae0-420a-43df-9a9e-f690c70e12d3 found in enumeration response, now use it to instantiate a call from the coSpace:

   Request:

   ```
   POST /api/v1/calls HTTP/1.1\r\n
   Host: 127.0.0.1\r\n
   Content-Type: www-formurl-encoded\r\n
   Content-Length: 44\r\n
   \r\n
   coSpace=581caae0-420a-43df-9a9e-f690c70e12d3
   ```

   Response:

   ```
   HTTP/1.1 200 OK\r\n
   Location: /api/v1/calls/8867d8f1-0918-4653-b41e-7341200e277a\r\n
   Connection: close\r\n
   \r\n
   ```

3. Call " /api/v1/calls/8867d8f1-0918-4653-b41e-7341200e277a" is successfully instantiated. Now create a participant as a call out from the newly-instantiated call:

   Request:

   ```
   POST /api/v1/calls/8867d8f1-0918-4653-b41e-7341200e277a/participants
   HTTP/1.1\r\n
   Host: 127.0.0.1\r\n
   Content-Type: www-formurl-encoded\r\n
   Content-Length: 33\r\n
   \r\n
   remoteParty=username1@example.com
   ```

   Response:

   ```
   HTTP/1.1 200 OK\r\n
   Location: /api/v1/participants/2671a77d-4bd5-4bf2-8ed6-f14afd80c2ac\r\n
   Connection: close\r\n
   \r\n
   ```

4. Participant " /api/v1/participants/2671a77d-4bd5-4bf2-8ed6-f14afd80c2ac" is now successfully created.

6 coSpace Related Methods

Note: Although the web app and other Cisco Meeting Server guides refer to " spaces" rather than " coSpaces" , the API still uses /coSpace objects. The Web Admin interface has been changed to refer to " spaces" .

This chapter details the API methods related to management of coSpaces. The chapter covers:

- retrieving coSpaces
- creating and modifying a coSpace
- retrieving detailed information about a single coSpace
- retrieving the members of a coSpace
- adding and modifying a coSpace member
- retrieving coSpace access methods
- creating and modifying coSpace access methods
- calling out from a coSpace
- bulk creating, updating and deleting coSpaces
- coSpace diagnostics
- using coSpace templates
- using access method templates
- retrieving text based meeting entry information for a coSpace
- configuring metadata on a coSpace

6.1 Retrieving coSpaces

GET method on the “/coSpaces” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve coSpaces other than the first “page" in the notional list (see Section 4.2.2).
filter	String	Supply “filter=<string>” in the URI to return just those coSpaces that match the filter
tenantFilter	ID	Supply tenantFilter=<tenant id> to return just those coSpaces associated with that tenant
callLegProfileFilter	ID	Supply callLegProfileFilter=<call leg profile id> to return just those coSpaces using that call leg profile

Parameters	Type/Value	Description/Notes
spaceTag	String	Name of the tag given to the particular **coSpace**. Tag for each space is unique, case-insensitive and may contain up to ten characters.

The response includes the total count of the number of coSpaces present which match the filter if provided, irrespective of the number returned within the response. (With no filter, this value is the total number of configured coSpaces.)

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <coSpaces total="N"> tag with potentially multiple <coSpace> elements within it. <coSpace> elements follow the general form on the left.
coSpace id	ID	The "ID" value returned in the opening tag is a unique identifier for the coSpace, and can be used for future modify / delete / query methods on that coSpace.
name	String	The human-readable name that will be shown on clients' UI for this coSpace. It can a have a maximum length of 200 characters.
uri	String	The URI that a SIP system would use to dial in to this coSpace. The URI can have a maximum length of 200 alpha numeric characters with the following criteria: • '.', '-', '_' are the only special characters that can be used. • URI cannot start or end with '.' and '-'; cannot be used together; cannot be used multiple times in a row.
secondaryUri	String	The secondary URI for this coSpace – this provides the same functionality as the "uri" parameter, but allows more than one URI to be configured for a coSpace. The secondaryURI can have a maximum length of 200 alpha numeric characters with the following criteria: • '.', '-', '_' are the only special characters that can be used. • URI cannot start or end with '.' and '-'; cannot be used together; cannot be used multiple times in a row.
callId	Number	The numeric ID that a user would enter at the IVR (or via a web client) to connect to this coSpace. It can have a maximum length of 200 digits.
tenant	ID	If provided, associates the specified call leg profile with this tenant
autoGenerated	true|false	Whether this coSpace has been added automatically or manually ▪ true - this coSpace has been added automatically as part of an LDAP sync operation. therefore it is not possible, to remove it except by modifying the parameters of the sync operation ▪ false - this coSpace has been added either via an API method or by using web app; it can be modified or removed via the API

6.2 Creating and Modifying a coSpace

- Creating: POST method to the "/coSpaces" node. If the coSpace was created successfully, a "200 OK" response is received, and the "Location" header contains the ID for the new coSpace
- Modifying: PUT method on a "/coSpaces/<coSpace ID>" node

Parameters	Type/Value	Description/Notes
userProvisionedCoSpace	ID	Instantiate a new coSpace from the user provisioned cospace. When this parameter is present all the other parameters are ignored. (from version 3.1)
name	String	The human-readable name that will be shown on clients' UI for this coSpace It can a have a maximum length of 200 characters.
uri	String	The URI that a SIP system would use to dial in to this coSpace. The URI can have a maximum length of 200 alpha numeric characters with the following criteria: • '.', '-', '_' are the only special characters that can be used. • URI cannot start or end with '.' and '-'; cannot be used together; cannot be used multiple times in a row.
secondaryUri	String	The secondary URI for this coSpace – this provides the same functionality as the "uri" parameter, but allows more than one URI to be configured for a coSpace. The secondaryURI can have a maximum length of 200 alpha numeric characters with the following criteria: • '.', '-', '_' are the only special characters that can be used. • URI cannot start or end with '.' and '-'; cannot be used together; cannot be used multiple times in a row.
callId	Number	The numeric ID that a user would enter at the IVR (or via a web client) to connect to this coSpace. This parameter can have a maximum length of 200 digits.

Parameters	Type/Value	Description/Notes
cdrTag	String	Up to 100 characters of free form text to identify this coSpace in a CDR; when a " callStart" CDR is generated for a call associated with this coSpace, this tag will be written (as " cdrTag") to the callStart CDR. See the Cisco Meeting Server CDR Reference for details. The cdrTag can be modified in a PUT method.
passcode	Numeric	The security code for this coSpace with a maximum limit of 63 digits.
defaultLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	The default layout to be used for new call legs in this coSpace. See Default layout options for the difference in naming between the API and the Web Admin interface. If left blank, then the setting at the /callLegProfile associated with this coSpace is used.
tenant	ID	If provided, associates the specified tenant with this coSpace
callLegProfile	ID	If provided, associates the specified call leg profile with this coSpace
callProfile	ID	If provided, associates the specified call profile with this coSpace
callBrandingProfile	ID	If provided, associates the specified call branding profile with this coSpace
lobbyProfile	ID	If provided, associates the specified lobby profile with this coSpace.
dialInSecurityProfile	ID	If provided, associates the specified dial-in security profile with this coSpace (3.0 onwards)
requireCallId	true|false	If this value is supplied as true, and no callId is currently specified for the coSpace, a new auto-generated call Id will be assigned

Parameters	Type/Value	Description/Notes
secret	String	If provided, sets the security string for this coSpace. If absent, a security string is chosen automatically if the coSpace has a callId value. This is the security value associated with the coSpace that needs to be supplied with the callId for guest access to the coSpace.
regenerateSecret	true|false	• If provided as true - a new security value is generated for this coSpace and the former value is no longer valid (for instance, any hyperlinks including it will cease to work) • If provided as false - do not generate a new secret value for this coSpace; this has no effect This parameter is only valid for the modify (PUT) case.
nonMemberAccess	**true**|false	Controls whether non-members of the coSpace are able to have access to the coSpace. If not provided, behaviour defaults to true. (From version 2.0).
ownerJid	String	Indicates the coSpace is owned by the user with the specified JID. (From version 2.0).
streamUrl	URL	Indicates where the coSpace is streamed to, if streaming is initiated. (From version 2.1).
ownerAdGuid	ID	If provided, the coSpace will be owned by the user with the given AD GUID. (From version 2.1).
meetingScheduler	String	Name of person (not necessarily a user) who scheduled the creation of this coSpace, which if set is propagated to any call objects as the “ownerName” field. (From version 2.2).
panePlacementHighestImportance	Number	If panePlacementHighestImportance is provided, pane placement will be activated for this coSpace - the active range of importance values will be from " highest importance" down to 1 (inclusive). (From version 2.4)

Parameters	Type/Value	Description/Notes
panePlacementSelfPaneMode	**skip**|self|blank |<unset>	If pane placement is activated, this defines the layout behavior for this coSpace when viewing an endpoint's own layout pane. (From version 2.7) skip - same as the pre-2.7 version behavior, do not include a pane in the layout for viewing the system's own importance level (default). self - shows the viewing endpoint's video back to themselves blank - leaves a blank pane to indicate to the viewer of the endpoint where other participants will see them. <unset> - follows this order of precedence: use the value set for panePlacementHighestImportance on /calls, if panePlacementHighestImportance on /calls is unset, then use the value set for panePlacementHighestImportance on /coSpace (if the call is to a space), if panePlacementHighestImportance on /coSpace is also unset, then it reverts to the skip behavior defined above. By default, panePlacementSelfPaneMode is set to <unset>.
defaultAccessMethod	ID | " "	Associates the specified access method as the default access method to be used for dial outs.

Parameters	Type/Value	Description/Notes
panePlacementActiveSpeakerMode	allowself | suppressself | none	• allowself – displays the participant in the first pane if they are the active speaker, additional to displaying the participant in the pane set by pane placement. The same view is displayed for all participants. • suppressself – displays the active speaker in the first pane for all other participants other than the active speaker. For the active speaker, the previous speaker is displayed in the first pane. • none – the feature is disabled.
spaceTag	String	Name of the tag given to the particular **coSpace**. Tag for each space is unique, case-insensitive and may contain up to ten characters.

Note: You can also use this PUT to modify the values of a coSpace created in a web app. For example, the coSpace will have been created with the cdrTag of the user who created it but you can change that value with an API call. (This is unlike the cdrTag of an automatically generated coSpace, which cannot be updated with an API call.)

Default layout options

The naming of the defaultLayout options varies between the API and the Web Admin Interface **Configuration > coSpaces** page. The "mapping" is shown in the table below.

API	Web Admin Interface
allEqual	all equal
speakerOnly	full screen
telepresence	overlay (the loudest speaker is in a large pane and a number of the previous speakers are in small panes which overlay the bottom of the loudest speaker's pane.)
stacked	stack (the loudest speaker is in a large pane and a number of the previous speakers are in small panes below the loudest speaker's pane.)

6.2.1 Secondary coSpace URIs

Per coSpace, there is an optional secondaryUri parameter as shown above. This allows flexibility; for example, numeric dialing in addition to a name.

- When creating or modifying a coSpace (see the previous section) you can supply a secondaryUri parameter in addition to the form parameters in the table above e.g. uri

- The secondary URI will be checked for validity and uniqueness in the same way as the uri, and if valid, establishes a new URI by which the coSpace can be reached
- When retrieving information on an individual coSpace (see below) the secondaryUri value will be returned, if it is defined for this coSpace
- The secondaryUri can be created automatically during an LDAP sync if the new LDAP mapping parameter is used. See coSpaceSecondaryUriMapping

6.2.2 Auto-generation of coSpace callId

A new auto-generated Call Id is assigned if " requireCallId=true" is set via a create (POST) or modify (PUT) method on the coSpace, and no callId is currently specified for the coSpace.

6.3 Retrieving Detailed Information about a Single coSpace

GET method performed on a "/coSpaces/<coSpace ID>" node. If the coSpace ID supplied is valid, a "200 OK" response is received, containing a single " <coSpace id=<ID>" object with data as described above for the creating and modifying case.

Parameters	Type/Value	Description/Notes
name	String	The human-readable name that will be shown on clients' UI for this coSpace with a maximum length of 200 characters.
uri	String	The URI that a SIP system would use to dial in to this coSpace. The URI can have a maximum length of 200 alpha numeric characters with the following criteria: • '.', '-', '_' are the only special characters that can be used. • URI cannot start or end with '.' and '-'; cannot be used together; cannot be used multiple times in a row.
secondaryUri	String	The secondary URI for this coSpace – this provides the same functionality as the "uri" parameter, but allows more than one URI to be configured for a coSpace. The secondaryURI can have a maximum length of 200 alpha numeric characters with the following criteria: • '.', '-', '_' are the only special characters that can be used. • URI cannot start or end with '.' and '-'; cannot be used together; cannot be used multiple times in a row.

Parameters	Type/Value	Description/Notes
callId	Number	The numeric ID that a user would enter at the IVR (or via a web client) to connect to this coSpace. This parameter can have a maximum limit of 200 digits.
cdrTag	String	Up to 100 characters of free form text to identify this coSpace in a CDR; when a " callStart" CDR is generated for a call associated with this coSpace, this tag will be written (as " cdrTag") to the callStart CDR. See the Cisco Meeting Server CDR Reference for details. The cdrTag can be modified in a PUT method.
passcode	Numeric	The security code for this coSpace with a maximum limit of 63 digits.
defaultLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	The default layout to be used for new call legs in this coSpace. See Default layout options for the difference in naming between the API and the Web Admin interface
tenant	ID	If provided, associates the specified tenant with this coSpace
callLegProfile	ID	If provided, associates the specified call leg profile with this coSpace
callProfile	ID	If provided, associates the specified call profile with this coSpace
lobbyProfile	ID	If provided, associates the specified lobby profile with this coSpace.
callBrandingProfile	ID	If provided, associates the specified call branding profile with this coSpace

Parameters	Type/Value	Description/Notes
autoGenerated	true| false	This shows whether the coSpace was added automatically as part of an LDAP sync operation ■ true - this coSpace has been added automatically as part of an LDAP sync operation. therefore it is not possible, to remove it except by modifying the parameters of the sync operation ■ false - this coSpace has been added either via an API method or by using web app; it can be modified or removed via the API
nonMemberAccess	**true**| false	Whether non-members of the coSpace are able to have access to the coSpace, if not provided, behaviour defaults to true. (From version 2.0)
numAccessMethods	Number	If additional access methods have been defined for this coSpace, this returns a count of the number of additional access methods for the coSpace.
secret	String	If provided, sets the security string for this coSpace. If absent, a security string is chosen automatically if the coSpace has a callId value. This is the security value associated with the coSpace that needs to be supplied with the callId for guest access to the coSpace.
ownerId	ID	Indicates the coSpace is owned by the user with the specified GUID. (From version 2.0).
ownerJid	String	Indicates the coSpace is owned by the user with the specified JID. (From version 2.0).
streamUrl	URL	Indicates where the coSpace is streamed to, if streaming is initiated. (From version 2.1).
meetingScheduler	String	Name of person (not necessarily a user) who scheduled the creation of this coSpace, which if set is propagated to any call objects as the “ownerName” field. (From version 2.2).
panePlacementHighestImportance	Number	If provided, pane placement will be activated for this specific coSpace - the active range of importance values will be from " highest importance" down to 1(inclusive). (From version 2.4)

Parameters	Type/Value	Description/Notes
panePlacementSelfPaneMode	**skip**|self|blank |<unset>	Indicates if self pane mode is activated for this coSpace. (From version 2.7) skip - same as the pre-2.7 version behavior, on a per-viewer basis the screen layout skips the self pane and displays the pane of the next important participant (default). blank - leaves a blank pane instead of displaying the important participant, so the important participant still sees the other participants in the same pane position as all other viewers. <unset> - follows this order of precedence: use the value set for panePlacementHighestImportance on /calls, if panePlacementHighestImportance on /calls is unset, then use the value set for panePlacementHighestImportance on /coSpace (if the call is to a space), if panePlacementHighestImportance on /coSpace is also unset, then it reverts to the skip behavior defined above. By default, panePlacementSelfPaneMode is set to <unset>.
defaultAccessMethod	ID | " "	The default access method used for dial outs.

Parameters	Type/Value	Description/Notes
panePlacementActiveSpeakerMode	allowself | suppressself | none	Indicates if the active speaker will be displayed for this coSpace. • allowself – displays the participant in the first pane if they are the active speaker, additional to displaying the participant in the pane set by pane placement. The same view is displayed for all participants. • suppressself – displays the active speaker in the first pane for all other participants other than the active speaker. For the active speaker, the previous speaker is displayed in the first pane. • none – the feature is disabled.
spaceTag	String	Name of the tag given to the particular **coSpace**. Tag for each space is unique, case-insensitive and may contain up to ten characters.

6.3.1 Retrieving entry details for a specific coSpace

From 2.1, a “meetingEntryDetail” node is added to allow retrieval of entry details for a specific coSpace meeting. Perform a GET on /coSpaces/<coSpace id>/meeting EntryDetail.

Response values are uri and callId.

6.4 coSpace Member Methods

6.4.1 Retrieving the members of a coSpace

GET method on a “/coSpaces/<coSpace ID>/coSpaceUsers” node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those coSpace users that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve coSpaces other than the first " page" in the notional list (see Section 4.2.2).
callLegProfileFilter	ID	Supply callLegProfileFilter=<ID> to return just members using that call leg profile

The response includes the total count of coSpace users configured for the queried coSpace which match the filter, irrespective of the number returned within the response. (With no filter, this value is the total number of users associated with the coSpace.)

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <coSpaceUsers total="N"> tag with potentially multiple <coSpaceUser> elements within it. <coSpaceUser> elements follow the general form on the left.
coSpaceUser id	ID	<coSpaceUser> elements have their own ID and also contain an ID for the user.
userJid	String	The XMPP ID of the user.
userId	ID	Identifies the user with no relationship to any coSpace association, and may or may not be the same as the ID of the "coSpaceUser" object.
autoGenerated	true|false	Whether this coSpace user has been added automatically or manually ■ true - this coSpaceUser has been added automatically as part of an LDAP sync operation. Therefore, it is not possible to remove it except by modifying the parameters of the sync operation ■ false - this coSpaceUser has been added either via an API method or by using web app; it can be modified or removed via the API
callLegProfile	ID	If provided, associates the specified call leg profile with this coSpace user
canDestroy	true|**false**	Whether this user is allowed to delete the coSpace.
canAddRemoveMember	true|**false**	Whether this user is allowed to add or remove other members of the coSpace.
canChangeName	true|**false**	Whether this user is allowed to change the name of the coSpace.
canChangeUri	true|**false**	Whether this user is allowed to change the URI of the coSpace
canChangeCallId	true|**false**>	Whether this user is allowed to change the Call ID of the coSpace
canChangePasscode	true|**false**	Whether this user is allowed to change the passcode of the coSpace
canRemoveSelf	true|**false**	Whether this user is allowed to remove himself from the coSpace
canChangeNonMemberAccessAllowed	**true**|false	(Deprecated in 3.0) Whether this user is allowed to change the "non-member access allowed setting" of the coSpace.

6.4.2 Adding and modifying a coSpace member

- Adding: POST method to a "/coSpaces/<coSpace ID>/coSpaceUsers" node
- Modifying: PUT method performed on a "/coSpaces/<coSpace ID>/coSpaceUsers/<coSpaceUser ID>" node. The parameters that you can modify are listed below, with the exception of "userJid"

Parameters	Type/Value	Description/Notes
userJid *	String	JID of the user to be added as a member
callLegProfile	ID	If provided, associates the specified call leg profile with this coSpace user
lobbyProfile	ID	If provided, associate the specified lobby profile with the coSpace user.
		All these parameters are optional and have a default setting of false if not provided in a POST operation—also see the note on permissions below.
canDestroy	true|**false**	Whether this user is allowed to delete the coSpace.
canAddRemoveMember	true|**false**	Whether this user is allowed to add or remove other members of the coSpace.
canChangeName	true|**false**	Whether this user is allowed to change the name of the coSpace.
canChangeUri	true|**false**	Whether this user is allowed to change the URI of the coSpace
canChangeCallId	true|**false**	Whether this user is allowed to change the Call ID of the coSpace
canChangePasscode	true|**false**	Whether this user is allowed to change the passcode of the coSpace
canRemoveSelf	true|**false**	Whether this user is allowed to remove himself from the coSpace
canChangeNonMemberAccessAllowed	**true**|false	(Deprecated in 3.0) Whether this user is allowed to change the "non-member access allowed setting" of the coSpace.
canChangeScope	true | **false**	Whether this user is allowed to change the scope of access methods on the coSpace. If this parameter is not supplied in a create (POST) operation, it defaults to **`false`**.

If the member was added successfully, a "200 OK" response is received, and the "Location" header in the response contains the new user ID.

coSpace Permissions

Members with canAddRemoveMember set to true can add other users as members of the coSpace from a web app. New members have identical permissions to the member who added them, except in one case: when the original member also has canRemoveSelf set to false.

Members who cannot remove themselves from the coSpace (as controlled by canRemoveSelf) should not be able to create a second member in order to delete their own membership. Therefore any member created from web app by another member in this situation will have **`canAddRemoveMember`** set to false and **`canRemoveSelf`** set to true (see the table below). All other permissions are copied from the original member.

Original user permission		Created user permissions		Notes
canAddRemoveMember	canRemoveSelf	canAddRemoveMember	canRemoveSelf	
False	NA	NA	NA	They can't add another user
True	True	True	True	All permissions copied
True	False	False	True	All other permissions copied.

Using the API provides more flexibility: it is possible to create coSpaces with members who cannot remove themselves, but who can be removed by another member. Members can always be removed via the API.

Auto-generated members (created by an LDAP sync) have auto-generated permissions because it makes no sense to allow them to make changes that will be overwritten by the next LDAP sync. Therefore, for these users the following parameters are always set to false: canDestroy, canChangeName, canChangeUri, canChangeCallId and canRemoveSelf. The other "can" parameters are set to True. Note that changing any of these settings for an auto-generated member via the API will only have a temporary effect and will be overwritten at the next LDAP sync: you can discover whether a member is auto-generated – see the next section.

For a summary of default settings for the permissions, see the table below:

Permission	coSpace created by:		
	web app	LDAP sync	API
canDestroy	true	false	false
canAddRemoveMember	true	true	false
canChangeName	true	false	false
canChangeUri	true	false	false

Permission	coSpace created by:		
	web app	LDAP sync	API
canChangeCallId	true	false	false
canChangePasscode	true	true	false
canRemoveSelf	true	false	false
(Deprecated in 3.0) canChangeNonMemberAccessAllowed	true	false	true

6.4.3 Retrieving Information on a coSpace member

GET method performed on a "/coSpaces/<coSpace ID>/coSpaceUsers/<coSpaceUser ID>" node. If the retrieval is valid, a "200 OK" response is received, with containing a single <coSpaceUser id=<ID>> object with data as described above for the creating and modifying case. In addition

- the autoGenerated value shows whether the coSpace member was added to the coSpace automatically as part of an LDAP sync operation
- canChangeScope shows whether this user is allowed to change the scope of access methods on the coSpace

6.5 Multiple coSpace Access Methods

6.5.1 General information

There are two related tables of objects:

- Access method per-coSpace, " /coSpaces/<cospace ID>/accessMethods [/<accessMethod ID>]"
- Call leg profile, " /callLegProfiles/<callLegProfile ID>" .

Access method per coSpace

Access methods define combinations of URI , passcode , callId and secret that can be used to access a coSpace

Optionally, Access methods can have an associated callLegProfile; any call leg joining via such an Access method has that call leg profile applied to it. If the Access method has no call leg profile but the coSpace does, then so does coSpace's call

Note: When you send an email invitation from a web app to one or more people to join a coSpace or active call, only one set of URI, passcode, callId, secret information is included. If

the scope field for an access method is set to public then this information is used. If no access methods have a public scope then the call information from the coSpace's own configuration is included.

Call leg profile

A call leg profile can be associated with a coSpace object, making it the default call leg profile for all call legs in that coSpace (for instance, those that connect via its configured URI and secondaryUri). The effect of the coSpace call leg profile can still be overridden by more specific overrides imposed via call leg profiles configured for additional coSpace access methods. See the Call Leg Profile Methods section.

6.5.2 Retrieving coSpace access methods

GET method on the " /coSpaces/<coSpace id>/accessMethods/" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those coSpace access methods that match the filter
offset limit	Number number	An " offset" and " limit" can be supplied to retrieve coSpaces other than the first " page" in the notional list (see Section 4.2.2).
callLegProfileFilter	ID	Supply callLegProfileFilter=<ID> to return just accessMethods for coSpaces using that call leg profile

Response elements	Type/Value	Description/Notes
		Response is a collection of " <accessMethod id=<access method id>>" objects contained within an " <accessMethods>" object <accessMethod> elements follow the general form on the left.
accessMethod id	ID	
uri	String	The URI to be used for dialing in via this access method. It can have a maximum length of 200 characters.
callId	ID (Number)	The " call ID" to be used for connecting via this access method (using the IVR or Web Bridge login). callId can have a maximum length of 200 digits.
passcode	String	A passcode required for this access method. It can have a maximum length of 200 characters.
callLegProfile	ID	The ID of a call leg profile to apply to calls in via this access method

6.5.3 Creating and modifying coSpace access methods

- Creating: POST method to the " /coSpaces/<coSpace id>/accessMethods" node
- Modifying: PUT method on a " /coSpaces/<coSpace id>/accessMethods/<access method id>" node

Parameters	Type/Value	Description/Notes
uri	String (URI user part)	The URI to be used for dialing in via this access method. (The URI " user part" is the part before any '@' character in a full URI.). It can have a maximum length of 200 characters.
callId	ID (Number)	The " call ID" to be used for connecting via this access method (using the IVR or Web Bridge login). It can have a maximum length of 200 digits.
passcode	String	A passcode required for this access method with a maximum length of 200 digits.
name	String	The name associated with this access method (from version 2.9). It can a have a maximum length of 200 characters.
callLegProfile	ID	The ID of a call leg profile to apply to calls in via this access method
lobbyProfile	ID	The ID of the lobbyProfile to be applied to this access-method

Parameters	Type/Value	Description/Notes
secret	String	If provided, sets the security string for this coSpace access method. If absent, a security string is chosen automatically if the coSpace access method has a callId value. This is the security value associated with the coSpace access method that needs to be supplied with the callId for guest access to the coSpace via this access method.
regenerateSecret	true|false	• true – a new security value is generated for this coSpace access method and the former value is no longer valid (for instance, any hyperlinks including it will cease to work) • false – do not generate a new secret value for this coSpace access method; this has no effect This parameter is only valid for the modify (PUT) case

Parameters	Type/Value	Description/Notes
scope	public|private| member|directory	The visibility of this coSpace access method to users of web app who are members of the coSpace: ■ public – details of this coSpace access method are visible to members of the coSpace and all participants in the meeting ■ private – details of this coSpace access method is visible only to the owner of the space in web app, or visible to admin users using the Call Bridge API. No video domain will be appended to the video address. ■ member – details of this coSpace access method are visible to members of the coSpace (from version 3.1) ■ directory – details of this coSpace access method can be found through search (from version 3.1) [Note: in 3.1 there is no search, so behavior is the same as **`public`**] Note: if you set the scope to **`public`** then the web app can no longer edit the coSpace details. In addition, the uri shown under the name is that from the access method.
importance	Number	The importance value assigned to all participants joining via this access method. Maximum value is 2,147,483,647. (From version 2.4)

Parameters	Type/Value	Description/Notes
dialInSecurityProfile	ID	If provided, associates the specified dial in security profile with this coSpace access method (3.0 onwards)

If the coSpace access method is created successfully, a “200 OK” response will be received, and the “Location” header in the response will contain the new coSpace access method ID.

6.5.4 Retrieving information on an individual coSpace access method

GET method on a " /coSpaces/<coSpace id>/accessMethods/<access method id>" node.

If the access method ID supplied is valid, a " 200 OK" response and a single <accessMethod id=access method id> object will be returned with data in the previous section.

6.6 Calling Out from a coSpace

Adding a remote party to a coSpace requires that this coSpace has an active call from which connections can be made. Essentially this makes an initial call out from a coSpace a combination of two other API methods:

1. Creation of a new call.
2. Adding a new outgoing call leg to a call.

These methods are described in the Section 8.

6.7 Bulk creating, updating and deleting coSpaces

6.7.1 Creating /cospaceBulkParameterSets

- Creating: POST method to the " /cospaceBulkParameterSets" node. Creates a new parameter set, see table below. Returns location of new parameter set /cospaceBulkParameterSets/<bulk parameter set guid>
- Modifying: PUT method to the " /cospaceBulkParameterSets" node. Updates the parameters within this parameter set, but needs to be synchronized for it to take effect.

Parameter	Type	Description
startIndex	Number	Index that coSpace mappings start from (inclusive)
endIndex	Number	Index that coSpace mappings end at (inclusive)

Parameter	Type	Description
coSpaceUriMapping	String	If specified, this is the mapping that describes what URIs will be used for the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set, coSpace will not have a dialable URI. Syntax: uri-mapping = [uri-component] [“$index$”] [uri-component] Where: uri-component = *(uri-character / escaped-character) uri-character = *(unescaped-character EXCLUDING ‘@’) unescaped-character = any character EXCLUDING ‘$’ and ‘\’ escaped-character = “\\” / “\$” ; producing ‘\’ and ‘$’ respectively. These need to be unique so if an index is not used there will be clashes, unless the field is just left completely blank .
coSpaceNameMapping	String	If specified, this is the mapping that describes what names will be used for the cospaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. Syntax: name-mapping = [name-component] [“$index$”] [name-component] Where: name-component = *(unescaped-character / escaped-character) unescaped-character = any character EXCLUDING ‘$’ and ‘\’ escaped-character = “\\” / “\$” ; producing ‘\’ and ‘$’ respectively. These are not required to be unique.
coSpaceCallIdMapping	String	If specified, this is the mapping that describes what call IDs will be used for the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not have a callId. Syntax: id-mapping = [id-component] [“$index$”] [id-component] Where: id-component = *(unescaped-character / escaped-character) unescaped-character = any character EXCLUDING ‘$’ and ‘\’ escaped-character = “\\” / “\$” ; producing ‘\’ and ‘$’ respectively These need to be unique so if index is not used there will be clashes, unless the field is just left completely blank. Secrets will be auto-generated if CallIdMapping is set.
tenant	ID	If specified this is the tenant to be associated with the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not be associated with a tenant.

Parameter	Type	Description
callProfile	ID	If specified this is the call profile to be associated with the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not be associated with a call profile.
callBrandingProfile	ID	If specified this is the call branding profile to be associated with the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not be associated with a call branding profile.
nonMemberAccess	**true**| false	Whether non-members will be able to access the bulk created coSpaces. If this parameter is not supplied in a create (POST) operation, it defaults to " true" and non members can access the coSpace.

6.7.2 Retrieving the parameter sets for creating coSpaces in bulk

GET method on " /cospaceBulkParameterSets " node.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <cospaceBulkParameterSets total="N"> tag with potentially multiple <cospaceBulkParameterSet> elements within it. <cospaceBulkParameterSet> elements follow the general form on the left.
startIndex	Number	Index that coSpace mappings start from (inclusive)
endIndex	Number	Index that coSpace mappings end at (inclusive)

6.7.3 Retrieving information on an individual /cospaceBulkParameterSet

GET method on " /cospaceBulkParameterSets/<coSpace bulk parameter set id>" node

Parameter	Type	Description
		Response is structured as a top-level <cospaceBulkParameterSets total="N"> tag with potentially multiple <cospaceBulkParameterSet> elements within it. Each <cospaceBulkParameterSet> element may include the following elements.
startIndex	Number	Index that coSpace mappings start from (inclusive)
endIndex	Number	Index that coSpace mappings end at (inclusive)

Parameter	Type	Description
coSpaceUriMapping	String	If specified, this is the mapping that describes what URIs will be used for the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set, coSpace will not have a dialable URI. Syntax: uri-mapping = [uri-component] ["$index$"] [uri-component] Where: uri-component = *(uri-character / escaped-character) uri-character = *(unescaped-character EXCLUDING '@') unescaped-character = any character EXCLUDING '$' and '\' escaped-character = "\\" / "\$" ; producing '\' and '$' respectively. These need to be unique so if an index is not used there will be clashes, unless the field is just left completely blank .
coSpaceNameMapping	String	If specified, this is the mapping that describes what names will be used for the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. Syntax: name-mapping = [name-component] ["$index$"] [name-component] Where: name-component = *(unescaped-character / escaped-character) unescaped-character = any character EXCLUDING '$' and '\' escaped-character = "\\" / "\$" ; producing '\' and '$' respectively. These are not required to be unique.
coSpaceCallIdMapping	String	If specified, this is the mapping that describes what call IDs will be used for the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not have a callId Syntax: id-mapping = [id-component] ["$index$"] [id-component] Where: id-component = *(unescaped-character / escaped-character) unescaped-character = any character EXCLUDING '$' and '\' escaped-character = "\\" / "\$" ; producing '\' and '$' respectively These need to be unique so if index is not used there will be clashes, unless the field is just left completely blank. Secrets will be auto-generated if CallIdMapping is set.
tenant	ID	If specified this is the tenant to be associated with the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not be associated with a tenant.

Parameter	Type	Description
callProfile	ID	If specified this is the call profile to be associated with the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not be associated with a call profile.
callBrandingProfile	ID	If specified this is the call branding profile to be associated with the coSpaces created with a /cospaceBulkSync using this cospaceBulkParameterSet. If not set then the coSpace will not be associated with a call branding profile.
nonMemberAccess	**true**| false	Whether non-members will be able to access the bulk created coSpaces. If this parameter is not supplied in a create (POST) operation, it defaults to " true" and non members can access the coSpace.

6.7.4 Queueing the bulk sync operations

- Creating: POST method to the " /cospaceBulkSyncs" node. Queues the bulk sync operations for execution as soon as possible. Returns location /cospaceBulkSync/<bulk sync guid>

  Note: Bulk Sync will iterate between startIndex and endIndex (inclusive at both end) and expand and insert the mapping parts .

- Modifying: PUT method to the " /cospaceBulkSyncs" node not supported.

Parameter	Type/Value	Description
cospaceBulkParameterSet	ID	Parameter set GUID that is going to be synchronised
removeAll	true| **false**	If supplied, determines whether the sync will remove all entries that were created using the parameter set. Used only if you need to remove all spaces that were created previously. If set to true then no spaces will be created. If set to false, or omitted, then all spaces previously created using this parameter set will be removed and new spaces based on the new mappings will be created. If this parameter is not supplied in a create (POST) operation, it defaults to " false"

6.7.5 Retrieving the bulk sync operations

GET method on " /cospaceBulkSyncs " node.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <cospaceBulkSyncs total="N"> tag with potentially multiple <cospaceBulkSync> elements within it. <cospaceBulkSync> elements follow the general form on the left.
cospaceBulkParameterSet	ID	Parameter set that was used for this bulk sync
status	pending| running| complete| failedCoSpaceUriConflict| failedCallIdConflict| failedIndexRangeInvalid| failedIndexRangeTooGreat| failedNoSuchParameterSet| failed	Status of the sync operation: **pending** - the sync operation is in a queue waiting to execute **running** - the sync operation is currently running **complete** - the sync operation has successfully completed **failedCoSpaceUriConflict** - the sync failed because it would involve creating a URI that conflicts with one that already exists **failedCallIdConflict** - the sync failed because it would involve creating a call ID that conflicts with one that already exists **failedIndexRangeInvalid** - the sync failed because the " startIndex" was greater than the " endIndex" **failedIndexRangeTooGreat** - the sync failed because the difference between " endIndex" and " startIndex" was too large **failedNoSuchParameterSet** - the " cospaceBulkParameterSet" refered to in the sync command did not exist **failed** - the sync operation failed
removeAll	true| **false**	If supplied, determines whether the sync will remove all entries that were created using the parameter set. Used only to remove all spaces that were created previously. If set to true then no spaces will be created. If set to false, or omitted, then all spaces previously created using this parameter set will be removed and new spaces based on the new mappings will be created. If this parameter is not supplied in a create (POST) operation, it defaults to " false"

6.7.6 Retrieving a specific bulk sync operation

GET method on " /cospaceBulkSyncs/<coSpace bulk sync id>" node.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <cospaceBulkSyncs total="N"> tag with potentially multiple <cospaceBulkSync> elements within it. <cospaceBulkSync> elements follow the general form on the left.
cospaceBulkParameterSet	ID	Parameter set that was used for this bulk sync
status	pending| running| complete| failedCoSpaceUriConflict| failedCallIdConflict| failedIndexRangeInvalid| failedIndexRangeTooGreat| failedNoSuchParameterSet| failed	Status of the sync operation: **pending** - the sync operation is in a queue waiting to execute **running** - the sync operation is currently running **complete** - the sync operation has successfully completed **failedCoSpaceUriConflict** - the sync failed because it would involve creating a URI that conflicts with one that already exists **failedCallIdConflict** - the sync failed because it would involve creating a call ID that conflicts with one that already exists **failedIndexRangeInvalid** - the sync failed because the " startIndex" was greater than the " endIndex" **failedIndexRangeTooGreat** - the sync failed because the difference between " endIndex" and " startIndex" was too large **failedNoSuchParameterSet** - the " cospaceBulkParameterSet" refered to in the sync command did not exist **failed** - the sync operation failed

Response elements	Type/Value	Description/Notes
removeAll	true| **false**	If supplied, determines whether the sync will remove all entries that were created using the parameter set. Used only to remove all spaces that were created previously. If set to true then no spaces will be created. If set to false, or omitted, then all spaces previously created using this parameter set will be removed and new spaces based on the new mappings will be created. If this parameter is not supplied in a create (POST) operation, it defaults to " false"

6.7.7 Examples

Creating coSpaces in bulk

1. Create a cospaceBulkParameterSet with parameters:

   startIndex=1000
   endIndex=1999
   coSpaceUriMapping=space.$index$
   coSpaceNameMapping=Space $index$
   coSpaceCallIdMapping=811$index$

2. Create a cospaceBulkSync with parameters:

   cospaceBulkParameterSet=<GUID from above>

This will create 1000 spaces starting with

"Space 1000" space.1000@domain.com , callID=8111000

and ending in

"Space 1999" space.1999@domain.com , callID=8111999

To update the range:

1. PUT new range to cospaceBulkParameterSets/<GUID from above>
2. Create a cospaceBulkSync with parameters:

   cospaceBulkParameterSet=<GUID from above>

This deletes all the previous spaces and creates a new set. This whole operation will succeed or fail. In failure the transaction will be rolled back and the spaces that previously existed will still be there.

To delete a range:

1. Create a cospaceBulkSync with parameters:

   cospaceBulkParameterSet=<GUID from above>&removeAll=true

This removes all spaces that were created using this parameter set. They will get removed even if they have been renamed, or edited in any other way.

6.8 coSpace Diagnostics Methods

A POST to " /coSpaces/<coSpace id>/diagnostics" triggers the generation of call diagnostics for the specified coSpace.

6.9 Using coSpace templates

6.9.1 Creating, modifying, retrieving, enumerating and deleting coSpace templates

From 2.9, API node **`/coSpaceTemplates`** is used to implement coSpace templates with the following request parameters:

Parameters	Type/Value	Description/Notes
name	String	the human-readable name associated with this coSpace template with a maximum length of 200 characters.
description	String	a longer description of the coSpace template to give users an explanation of why they might want to use this template
callProfile	ID	if provided, associates the specified call profile with this coSpaceTemplate
callLegProfile	ID	if provided, associates the specified call leg profile with this coSpaceTemplate
dialInSecurityProfile	ID	if provided, associates the specified dial-in security profile with this coSpaceTemplate (3.0 onwards)

This API node **`/coSpaceTemplates`** supports the following operations:

- POST to **`/coSpaceTemplates`**
- PUT to **`/coSpaceTemplates/<coSpace template id>`**
- DELETE on **`/coSpaceTemplates/<coSpace template id>`**
- GET on **`/coSpaceTemplates/<coSpace template id>`**, gives the following responses:

Response values	Type/Value	Description/Notes
name	String	the human-readable name associated with this coSpace template with a maximum length of 200 characters.
description	String	a longer description of the coSpace template to give users an explanation of why they might want to use this template
callProfile	ID	if provided, the specified call profile associated with this coSpaceTemplate
callLegProfile	ID	if provided, the specified call leg profile associated with this coSpaceTemplate
lobbyProfile	ID	If specified, associates the specified lobby profile with this coSpace template.
dialInSecurityProfile	ID	if provided, the specified dial-in security profile associated with this coSpaceTemplate (3.0 onwards)
numAccessMethodTemplates	Number	The number of access method templates associated with this coSpace template
defaultAccessMethodTemplate	ID | " "	If specified, associates the access method template as the default one for the coSpace template. When a coSpace is instantiated from the coSpace template, the instantiated default access method template becomes the default access method for the coSpace.

- Enumerate GET on **`/coSpaceTemplates`**, gives the following responses:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve coSpace templates other than the first page in the notional list
limit		
filter	String	supply filter=<string> to return just those coSpace templates that match the filter

The response is structured as a top-level <coSpaceTemplates total=" N" > tag with potentially multiple <coSpaceTemplate> elements within it.

Each <coSpaceTemplate> tag may include the following elements:

Response elements	Type/Value	Description/Notes
name	String	the human-readable name associated with this coSpace template with a maximum length of 200 characters.

Response elements	Type/Value	Description/Notes
callProfile	ID	if provided, the specified call profile associated with this coSpaceTemplate
callLegProfile	ID	if provided, the specified call leg profile associated with this coSpaceTemplate
lobbyProfile	ID	If specified, associates the specified lobby profile with this coSpace template.
dialInSecurityProfile	ID	if provided, the specified dial-in security profile associated with this coSpaceTemplate (3.0 onwards)
numAccessMethodTemplates	Number	The number of access method templates associated with this coSpace template
defaultAccessMethodTemplate	ID | " "	If specified, associates the access method template as the default one for the coSpace template. When a coSpace is instantiated from the coSpace template, the instantiated default access method template becomes the default access method for the coSpace.

6.10 Using Access Method templates

6.10.1 Creating, modifying, retrieving, enumerating and deleting coSpace template access method templates

2.9 introduced the API node **`/coSpaceTemplates/<coSpace template ID>/accessMethodTemplates`** with the following request parameters:

Parameters	Type/Value	Description/Notes
name	String	the human-readable name associated with this access method template with a maximum length of 200 characters.
uriGenerator	String	the expression to be used to generate URI values for this access method template; the allowed set of characters are 'a' to 'z', 'A' to 'Z', '0' to '9', '.', '-', '_' and '$'; if non empty it must contain exactly one '$' character.
callLegProfile	ID	if provided, associates the specified call leg profile with this accessMethodTemplate
lobbyProfile	ID	If specified, associates the specified lobby profile with this accessMethod template.
generateUniqueCallId	true | **false**	whether to generate a unique numeric ID for this access method which overrides the global one for the cospace if this parameter is not supplied in a create (POST) operation, it defaults to " false"

Parameters	Type/Value	Description/Notes
dialInSecurityProfile	ID	if provided, associates the specified dial-in security profile with this accessMethodTemplate (3.0 onwards)

The API node **`/coSpaceTemplates/<coSpace template ID>/accessMethodTemplates`** supports the following operations:

- POST to **`/coSpaceTemplates/<coSpace template ID>/accessMethodTemplates`**
- PUT to **`/coSpaceTemplates/<coSpace template ID>/accessMethodTemplates/<access method template ID>`**
- DELETE on **`/coSpaceTemplates/<coSpace template ID>/accessMethodTemplates/<access method template ID>`**
- GET on **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method template id>`**, gives the following responses:

Response values	Type/Value	Description/Notes
name	String	the human-readable name associated with this access method template with a maximum length of 200 characters.
uriGenerator	String	the expression to be used to generate URI values for this access method template; the allowed set of characters are 'a' to 'z', 'A' to 'Z', '0' to '9', '.', '-', '_' and '$'; if non empty it must contain exactly one '$' character.
callLegProfile	ID	if provided, the specified call leg profile associated with this accessMethodTemplate
generateUniqueCallId	true | **false**	whether to generate a unique numeric ID for this access method which overrides the global one for the cospace
dialInSecurityProfile	ID	if provided, the specified dial-in security profile associated with this accessMethodTemplate(3.0 onwards)

Response values	Type/Value	Description/Notes
scope	public|private| member|directory	The visibility of this coSpace access method to users of web app who are members of the coSpace: ■ public – details of this coSpace access method are visible to members of the coSpace and all participants in the meeting ■ private – details of this coSpace access method is visible only to the owner of the space in web app, or visible to admin users using the Call Bridge API. No video domain will be appended to the video address. ■ member – details of this coSpace access method are visible to members of the coSpace (from version 3.1) ■ directory – details of this coSpace access method can be found through search (from version 3.1) [Note: in 3.1 there is no search, so behavior is the same as **`public`**] Note: if you set the scope to **`public`** then the web app can no longer edit the coSpace details. In addition, the uri shown under the name is that from the access method.

- Enumerate GET on **`/coSpaceTemplates/<coSpace template ID>/accessMethodTemplates`**, gives the following responses:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve coSpace access method templates other than the first page in the notional list
limit		
filter	String	supply filter=<string> to return just those coSpace access method templates whose name matches the filter
callLegProfileFilter	String	supply callLegProfileFilter=<string> to return just those coSpace access method templates that use the specified call leg profile

Response is structured as a top-level <accessMethodTemplates total=" N" > tag with potentially multiple <accessMethodTemplate> elements within it.

Each <accessMethodTemplate> tag may include the following elements:

Response elements	Type/Value	Description/Notes
name	String	the human-readable name associated with this access method template with a maximum length of 200 characters.

-

Response elements	Type/Value	Description/Notes
uriGenerator	String	the expression to be used to generate URI values for this access method template; the allowed set of characters are 'a' to 'z', 'A' to 'Z', '0' to '9', '.', '-', '_' and '$'; if non empty it must contain exactly one '$' character.
callLegProfile	ID	if provided, associates the specified call leg profile with this access method template
generateUniqueCallId	true | false	whether to generate a unique numeric ID for this access method which overrides the global one for the cospace if this parameter is not supplied in a create (POST) operation, it defaults to " false"
dialInSecurityProfile	ID	if provided, the specified dial-in security profile associated with this access method template(3.0 onwards)
scope	public| private| member| directory	Visibility of access methods using this template for users that are members of the coSpace. (3.2 onwards) • public – details of this coSpace access method are visible to members of the coSpace and all participants in the meeting • private – details of this coSpace access method is visible only to the owner of the space in web app, or visible to admin users using the Call Bridge API. No video domain will be appended to the video address. • member – details of this coSpace access method are visible to members of the coSpace • directory – details of this coSpace access method can be found through search If this parameter is not supplied in a POST operation, it defaults to **`private`**. Any spaces created with Private scope will not have a domain in the video address.

6.11 Retrieving text based meeting entry information for a coSpace

The Email invitation API is introduced to retrieve text based meeting entry information suitable for distributing, typically via e-mail.

- GET on **`/api/v1/coSpaces/<coSpace id>/accessMethods/<access method id>/emailInvitation`**

- GET on **`/api/v1/coSpaces/<coSpace id>/emailInvitation`**

URI Parameters	Type/Value	Description/Notes
language (optional)	String	In the form of a language tag " xx" or " xx_XX" (xx language code and XX region code) or any other string between 1 and 32 characters (allowed characters: 'a'-'z', 'A'-'Z', '0'-'9', and '_'). **Note:** Refer to Cisco Meeting Server Customization Guidelines for the list of supported languages and for details on customizing the email invite.
organizer (optional)	String	If provided, includes the organizer details in the email invitation text. The organizer details could be name or email address of the organizer/host as included in the API.

Response Elements	Type/ Value	Description/Notes
invitation	String	Email invitation text.
language	String	Language tag of email invitation. If no language is specified, then it defaults to en_US. If the specified language is invalid, then a " 400 - Bad Request" response is returned.

Failure responses

Response Elements	Type/ Value	Description / Notes
`<failureDetails> <parameterError parameter=\"language\" error=\"invalidValue\" /></failureDetails>`	400 - Bad Request	You entered an empty string or an invalid character as a language parameter.
`<failureDetails> <parameterError parameter=\"language\" error=\"valueTooLong\" /></failureDetails>`	400 - Bad Request	You entered a long language parameter.

Response Elements	Type/ Value	Description / Notes
`<failureDetails> <retryAfter=1 /></failureDetails>`	503 - Service Unavailable	You tried to retrieve the text based meeting entry information when the server was busy or was fetching externally hosted template. Retry later or retry after recommended `retryAfter` period in seconds.

Note: The email invitation text will include the webbridgeaddress or the IVR numbers, only if they are configured in the webbridgeprofile and mapped to tenant level or system level.

6.12 Configuring metadata on a coSpace

Metadata is a text string which can be configured on the coSpace, which allows management applications such as Cisco Meeting Management to store metadata on a coSpace. The metadata comprises of a list of participants that is retrieved while using the blast dial feature on Meeting Management. This is supported from version 3.2 onwards with the API node `/coSpaces/<coSpace id>/metadata` on the following methods:

- PUT to `/coSpaces/<coSpace id>/metadata`
- GET on `/coSpaces/<coSpace id>/metadata`

The following parameters are retrieved for the blast dial feature:

Parameters	Type/Value	Description/Notes
blastDial	String	List of Blast Dial objects

Parameters	Type/Value	Description/Notes
attemptLimit	Numeric	The maximum number of times the system will attempt to dial out a contact if they fail to connect to the call.
attemptDelay	Numeric	The minimum duration the system waits before retrying to dial out a contact. The default is 180 seconds. .

<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>participants</td><td>String</td><td>List of participant objects
<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>address</td><td>string</td><td>The participant's email address.</td></tr>
<tr><td>name</td><td>string</td><td>The participant's name</td></tr>
<tr><td>audioPrompt</td><td>on|off</td><td>This parameter is set at the participant level. On – Blast dial audio prompt will be played, and user has to enter DTMF key to join the conference. Off – Blast dial audio prompt will not be played, and user doesn't have to enter any DTMF key to join the conference.</td></tr>
<tr><td>audioPromptGlobal</td><td>on|off</td><td>This parameter is set at the global setting on Meeting Management. On – The audio prompt is enabled for all the participants at global level. Off – The audio prompt is disabled for all the participants at global level. Enabling or disabling the audio prompt at global level overrides the setting at the participant level.</td></tr>
</table></td></tr>
</table>

The **`coSpaceMetaDataConfigured`** response element on GET on **`/calls/<call id>`** returns true if there is metadata configured on **`/cospaces/<cospace id>/metadata`** and false otherwise.

Note: Some Meeting Management features such as blast dial require metadata to be stored on the coSpace. Changing the metadata can cause these features to fail.

7 Dial Plan Methods

This chapter details the API methods related to configuring dial plans for outbound calls, inbound calls and call forwarding. The chapter covers:

- retrieving outbound dial plan rules
- creating and modifying outbound dial plan rules
- retrieving information on an individual outbound dial plan rule
- transforming outbound dial plans
- retrieving dial plan rules for incoming calls
- creating and modifying dial plan rules for incoming calls
- retrieving information on the dial plan rule for an individual incoming call
- retrieving dial plan rules for forwarding incoming calls
- creating and modifying dial plan rules for forwarding incoming calls
- retrieving information on the dial plan rule to forward an individual incoming call

7.1 Outgoing Dial Plan API Methods

7.1.1 Access to the outgoing dial plan

Typically, the configuration of which trunks / proxies to use for outbound calls is based on the domain of the (SIP) destination being called, which is specified in the outgoing dial plan. The outgoing dial plan sits in the API object tree under the " /outboundDialPlanRules" node, use the POST method to create the outgoing dial plan or set it up via the Web Admin Interface (see note below).

If you are deploying Call Bridge clustering, use the API parameter scope to choose whether to apply each outbound dial plan rule to every Call Bridge in the cluster, or just to a particular Call Bridge so the Call Bridge can be trunked to its local Call Control solution (if appropriate).

Note: The API parameter callRouting specifies the mechanism for traversal of outgoing SIP/Lync calls, use this parameter to set up firewall traversal for SIP and Lync devices. This is still a beta feature.

Note: On the Web Admin Interface, the table of outbound rules is configured through the **Configuration > Dial plan** page. All rules added via the Web Admin Interface are global and applied to every Call Bridge in the cluster. You cannot use the Web Admin interface to specify the call routing for outbound SIP/Lync calls using a specific Call Bridge or Call Bridge group.

7.1.2 Retrieving outbound dial plan rules

GET method on the " /outboundDialPlanRules/" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those outbound dial plan rules that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve coSpaces other than the first " page" in the notional list (see Section 4.2.2).
tenantFilter	ID	If supplied, this filter only returns those outbound dial plan rules associated with the specified tenant.

Response elements	Type/Value	Description/Notes
		Response is a collection of " <outboundDialPlanRule id=<ID>>" objects contained within an " <outboundDialPlanRules>" object <outboundDialPlanRule> elements follow the general form on the left.
outboundDialPlanRule id	ID	
domain	String	The domain to match in order to apply the dial plan rule; either a complete value (e.g. " example.com") or a “wildcarded” one (e.g. " *.com")
priority	Number	A numeric value which determines the order in which dial plan rules (including rules with wild-carded domains) will be applied. Rules with higher priority values are applied first
tenant	ID	If a tenant is specified, this rule will only be used to make outbound call legs from calls associated with that tenant; otherwise, this rule may be used from any call.

7.1.3 Creating and modifying outbound dial plan rules

- Creating: POST method to the " /outboundDialPlanRules" node. If the outgoing dial plan rule is created successfully, a “200 OK” response will be received, and the “Location” header in the response will contain the new outgoing dial plan rule ID
- Modifying: PUT method on an " /outboundDialPlanRules/<outbound dial plan rule ID>" node

Parameters	Type/Value	Description/Notes
domain *	String	The domain to match in order to apply the dial plan rule; either a complete value (e.g. " example.com") or a “wildcarded” one (e.g. " *.com")

Parameters	Type/Value	Description/Notes
priority	Number	A numeric value which determines the order in which dial plan rules (including rules with wildcarded domains) will be applied. Rules with higher priority values are applied first. If a rule is matched, but the call cannot be made, then other lower priority rules may be tried depending on the failureAction parameter for the rule.
localContactDomain	String	Used when forming an explicit contact domain to be used: if you leave this field blank then the localContactDomain is derived from the local IP address. If you are using Lync, we suggest that you set localContactDomain. If you are not using Lync, we recommend that localContactDomain is not set to avoid unexpected issues with the SIP call flow.
localFromDomain	String	Used when forming the calling party for outgoing calls using this dial plan rule
sipProxy	String	The address (IP address or hostname) of the proxy device through which to make the call. If not set, it is a direct call.
trunkType	**sip**| lync| avaya	Used to set up rules to route calls out to third party SIP control devices such as CiscoExpressway, Avaya Manager or Lync servers. If set to lync or avaya then outgoing calls that use this rule will be made as Lync or Avaya calls with some specialized behavior. sip means that calls using this rule will be standard SIP calls. A common use of the Meeting Server is with an Avaya PBX; these calls will be audio-only. However, the Meeting Server does not impose this restriction on interoperability with Avaya products (some of which support video also): therefore a call of type of 'avaya' does not imply that the call is audio-only.
failureAction	**stop**| continue	Whether or not to try the next outbound dial plan rule if the current one did not result in a connected call. If a rule has a failureAction of stop, then no further rules are used.

Parameters	Type/Value	Description/Notes
sipControlEncryption	**auto**| encrypted| unencrypted	Whether to enforce use of encrypted control traffic on calls made via this rule: • encrypted: allow only encrypted SIP control traffic (TLS connections) • unencrypted: use only unencrypted traffic (TCP or UDP) • auto: attempt to use encrypted control connections first, but allow fall back to unencrypted control traffic in the event of failure. Note: Ensure all " Lync" outbound dialing rules are explicitly set to **Encrypted** mode to prevent the Call Bridge attempting to use unencrypted TCP for these connections in the event of the TLS connection attempt failing.
scope	**global**| callBridge| callBridgeGroup	The entities for which this outbound dial plan rule is valid: • global – all Call Bridges are able to use this outbound dial plan rule to reach a matching domain. • callBridge – this outbound dial plan rule is only valid for a single nominated Call Bridge – whose ID is given in callBridge parameter. • callBridgeGroup – this outbound dial plan rule is only valid for a single nominated Call Bridge Group – whose ID is given in the callBridgeGroup parameter. (From version 2.2). If this parameter is not supplied in a create (POST) operation it defaults to “global”.
callBridge	ID	If the rule has a scope of callBridge (see above), this is the id of the Call Bridge for which the rule is valid
callBridgeGroup	ID	If the rule has a scope of callBridgeGroup (see above), this is the id of the Call Bridge Group for which the rule is valid (from version 2.2).
tenant	ID	If a tenant is specified, this rule will only be used to make outbound call legs from calls associated with that tenant; otherwise, this rule may be used from any call.

Parameters	Type/Value	Description/Notes
callRouting (beta feature)	**default**| traversal	This is the media routing that should be used for SIP calls originating from this rule: • default – calls using this rule will use normal, direct, media routing • traversal – media for calls using this rule will flow via a TURN server if this parameter is not supplied in a create (POST) operation, it defaults to “default” .

7.1.4 Retrieving information on an individual outbound dial plan rule

GET method on a " /outboundDialPlanRules/<outbound dial plan rule ID>" node. If the outbound dial plan rule ID supplied is valid, a " 200 OK" response and a single " <outboundDialPlanRule id=<ID>>" object will be returned with data as per the previous section.

7.2 Dial Transforms

Dial Transforms are applied to outgoing calls prior to the Outbound rules taking effect. When dial transforms are applied, the outbound dial plan rules are applied to the transformed number. Dial Transforms only affect Outbound calls, they do NOT affect gateway calls.

There are three stages to the transform:

- A “type” is applied, which defines the type of preprocessing to apply to the transform.
  - Raw: produces one component – $1
  - Strip: removes dots, dashes, spaces and produces one component – $1
  - Phone: use to transform to an international phone number – produces two components $1county code and $2number

  Note: A phone URI is recognized as a purely numeric string (optionally prefixed by a ‘+’) when it begins with a valid international dial code (e.g. 44 for UK or 1 for US) followed by the correct number of digits for a phone number for that region.
- The components are matched using regular expressions to see if the rule is valid
- An output string is created from the components according to the defined transform

Examples

Example	Type	Match	Transform
For US numbers, use 'vcs1' directly	Phone	($1/01/)	$2@vcs1
For UK numbers, add a prefix and use 'vcs2'	Phone	($1/44/)	90044$2@vcs2
For UK numbers starting with a 7, add '90044' as a prefix, add '123@mobilevcs' as a suffix	Phone	($1/44/)($2/^7/)	90044$2{}123@mobilevcs
For unrecognized all-digit strings, use '@vcs3' as a suffix	Strip	($1/(\d){6,}/)	$1@vcs3
Replace + with 00	Strip	($1/\+(\d)+/)	$1{/\+/00/}
Replace an alphanumeric regex e.g. (.*)@example.com and replace with \1.endpoint@vc.example.com	Raw	($1/(.*) @example.com/)	$1{/@example.com$/ .endpoint@vc.example.com/}

In a Call Bridge cluster you do not need to configure the dial transforms separately on each Call Bridge. The dial transforms for the cluster are those defined on the Call Bridge host server that is co-located with the first coSpace database in the database cluster.

Note: Although the same dial transforms are applied to all Call Bridge in the cluster, the outbound dial plan rules can be configured per-Call Bridge.

7.2.1 Retrieving dial transforms

GET method performed on the “/dialTransforms” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
filter	String	Enter a filter to retrieve only those dial transforms that match the strong.

Response elements	Type/Value	Description/Notes
Type	raw| strip| phone	Response is structured as a top-level <dialTransforms total=" N" > tag with potentially multiple <dialTransform> elements within it. The type of pre-processing to apply to this transform Raw: produces one component - $1 Strip: removes dots, dashes, spaces and produces one component - $1 Phone: An international phone number - produces two components $1county code and $2number

7.2.2 Setting up and modifying dial transforms

- Creating: POST method to the “/dialTransforms" node
- Modifying: PUT to " /dialTransforms/<dialTransform id>"

Parameters	Type/Value	Description/Notes
type	raw| strip| phone	The type of preprocessing to apply to this transform. Default is " raw" if parameter not supplied in a create (POST) operation. Raw: produces one component - $1 Strip: removes dots, dashes, spaces and produces one component - $1 Phone: An international phone number - produces two components $1county code and $2number
match	String	If provided, the regular expression describing whether this rule will be applied. An empty string means “match all”. This is a logically AND'd combination of regular expressions, each applied to a component of the pre-processed expression. The format is ($<componentnum_1>/<regex_1>/)($<componentnum_2>/<regex_2>/)($<componentnum_3>/<regex_3>/)... For example • `($2/abc/)`: Component 2 must contain 'abc' • `($1/^0/)($1/9$/)`: Component 1 must start with a 0 and end with a 9 • `($1/^44$/)($2/^7/)`: Component 1 must be '44' and component 2 must start with a 7

Parameters	Type/Value	Description/Notes
transform	String	The replacement transform to be applied. This allows references to the pre-processed components, as well as one or more regular expression substitutions encased in curly braces with the following special strings replaced as described. $<componentnum> : Replace with component $<componentnum>{}: Replace with component $<componentnum>{/<matchregex1>/<replaceregex1/}{/<matchregex2>/<replaceregex2/}{/<matchregex3>/<replaceregex3/}...: Replace with component, with all instances of matchregex1 replaced by replaceregex1, and subsequently matchregex2 replaced by replaceregex2, etc. Capture groups are supported. Examples are: • `abc` : Replace everything with 'abc' • `$1$2@t.com` : Component 1 followed by component 2 followed by " @t.com • `$1{}123@t.com` : Component 1 followed by " 123@t.com" • `$1{/999/123/}@t.com` : Component 1 with all instances of '999' replaced by '123', followed by " @t.com" • `$1{/\D//}{/^9//}@example.com` : Component 1 with all non-digits removed and leading 9 removed, followed by " @example.com"
priority	Number	The priority this transform rule should have. Rules with higher priorities are applied first
action	**accept**| acceptPhone| deny	The action to take if this rule matches. Defaults to " accept" if parameter not supplied in a create (POST) operation.

7.2.3 Retrieving detailed information about an individual dial transform

GET method performed on a “/dialTransform/<dialTransform id>" node. If the dial transform id supplied is valid, a “200 OK” response is received, with XML content matching section " Retrieving dial transforms" on page 87.

7.3 Incoming Call Matching Dial Plan API Methods

7.3.1 Access to incoming domain matching rules

When an incoming SIP call is routed to the Meeting Server, the Call Bridge looks through the configured inbound dial plan rules first and tries to match the " domain" part of the destination

URI " <user>@<domain>" against the rules. Use the POST method on API object **`/inboundDialPlanRules`** to create a new inbound dial plan rule to match against incoming SIP calls, or set it up via the Web Admin Interface (see note below).

Note: On the Web Admin Interface, the table of inbound rules is configured through the **Configuration > Incoming calls** page.

7.3.2 Retrieving incoming dial plan rules

GET method on the " /inboundDialPlanRules" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those incoming dial plan rules that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve incoming dial plan rules other than the first " page" in the notional list (see Section 4.2.2).
tenantFilter	ID	Supply tenantFilter to return only those inbound dial plan rules associated with the specified tenant

Response elements	Type/Value	Description/Notes
		Response is a collection of " <inboundDialPlanRule id=<ID>>" objects contained within an " <inboundDialPlanRules>" object <inboundDialPlanRule> elements follow the general form on the left.
inboundDialPlanRule id	ID	
domain	Text	The domain to match in order to apply the dial plan rule. Must be a complete value (e.g. " example.com")
priority	Number	Determines the priority of the inbound dial plan rule where multiple rules are applicable
resolveTocoSpaces	**true**|false	If set to true, calls to this domain are matched against coSpace URIs (if a match is then found, the incoming call leg becomes a participant in the coSpace).
resolveTolvrs	**true**|false	If set to true, calls to this domain are matched against configured IVR URIs (if a match is then found, the incoming call leg connects to that IVR).

Response elements	Type/Value	Description/Notes
resolveToLyncConferences	true|**false**	If set to true, calls to this domain are resolved to a Lync conference URL; if the resolution is successful, the incoming call leg becomes a participant in the Lync conference. If this parameter is not supplied in a create (POST) operation, it defaults to “false”.
resolveToLyncSimplejoin	true|**false**	If set to true, calls to this domain are resolved by an HTTPS lookup to the given URL. If the resolution is successful, the incoming call leg becomes a participant in the Lync conference. (From version 2.2).
tenant	ID	If specified, calls to this inbound domain are only matched against coSpace URIs for the specified tenant

7.3.3 Creating and modifying incoming dial plan rules

- Creating: POST method to the " /inboundDialPlanRules" node. If the incoming dial plan rule is created successfully, a “200 OK” response will be received, and the “Location” header in the response will contain the new incoming dial plan rule ID
- Modifying: PUT method on an " /inboundDialPlanRules/<inbound dial plan rule ID>" node

Parameters	Type/Value	Description/Notes
domain *	String	The domain to match in order to apply the dial plan rule. Must be a complete value (e.g. " example.com")
priority	numeric	inbound dial plan rules' configured domain values are always exactly matched against incoming calls. For the purposes of generating full URIs to advertise for incoming calls (especially cases where multiple rules are applicable) you can also set a numeric priority value – higher values will be preferred
resolveTocoSpaces	**true**|false	If set to true, calls to this domain will be matched against coSpace URIs (if a match is then found, the incoming call leg becomes a participant in the coSpace).
resolveTolvrs	**true**|false	If set to true, calls to this domain will be matched against configured IVR URIs (if a match is then found, the incoming call leg connects to that IVR).
resolveToLyncConferences	true|**false**	If set to true, calls to this domain will be resolved to a Lync conference URL; if the resolution is successful, the incoming call leg becomes a participant in the Lync conference. If this parameter is not supplied in a create (POST) operation, it defaults to “false”.

Parameters	Type/Value	Description/Notes
resolveToLyncSimplejoin	true|**false**	If set to true, calls to this domain will be resolved by an HTTPS lookup to the given URL. If the resolution is successful, the incoming call leg becomes a participant in the Lync conference. If this parameter is not supplied in a create (POST) operation, it defaults to “false”. (From version 2.2).
tenant	ID	If specified, calls to this inbound domain will only be matched against coSpace URIs for the specified tenant

7.3.4 Retrieving information on an individual incoming dial plan rule

GET method on a " /inboundDialPlanRules/<inbound dial plan rule ID>" node. If the incoming dial plan rule ID supplied is valid, a " 200 OK" response and a single " <inboundDialPlanRule id=<ID>>" object will be returned with data as per the previous section.

7.4 Incoming Call Forwarding Dial Plan API Methods

7.4.1 Access to incoming call forwarding rules

If the " domain" part of the destination URI of an incoming SIP call fails to match any of the inbound dial plan rules, the call will be handled according to the rules in the call forwarding dial plan rules. The rules decide whether to reject the call outright or to forward the call in bridge mode.

Call forwarding rules can overlap, and include wildcards. You order rules using the Priority value; higher numbered rules are tried first. By defining rules, you decide whether to forward the call or not. It might be appropriate to “catch” certain calls and reject them.

For calls that will be forwarded, you can rewrite the destination domain, a new call is created to the specified domain.

The call forwarding dial plan sits in the API object tree under a " /forwardingDialPlanRules" node. Use the POST method to create the forwarding rules or set them up via the Web Admin Interface (see note below)

Note: On the Web Admin Interface, the Incoming Call Forwarding rules are configured through the Call Forwarding section of the **Configuration > Incoming calls** page.

7.4.2 Retrieving incoming call forwarding dial plan rules

GET method on the " /forwardingDialPlanRules/" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those incoming call forwarding rules that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve coSpaces other than the first " page" in the notional list (see Section 4.2.2).
tenantFilter	ID	If supplied, this filter restricts the results returned to those forwarding dial plan rules that are associated with the specified tenant.

Response elements	Type/Value	Description/Notes
		Response is a collection of " <forwardingDialPlanRule id=<ID>>" objects contained within an " <forwardingDialPlanRules>" object <forwardingDialPlanRule> elements follow the general form on the left.
forwardingDialPlanRule id	ID	The matchPattern and priority are described in the next section
matchPattern	Text	The domain to match in order to apply the dial plan rule. Must be a complete domain name (e.g. " example.com") or a “wildcarded” one (e.g. *.com)
priority	Number	Numeric value used when determining the order in which to apply forwarding dial plan rules; higher values will be applied first
tenant	ID	The tenant associated with the forwardingDualPlanRule

7.4.3 Creating and modifying incoming call forwarding dial plan rules

- Creating: POST method to the " /forwardingDialPlanRules" node. If the forwarding dial plan rule is created successfully, a “200 OK” response will be received, and the “Location” header in the response will contain the new forwarding dial plan rule ID
- Modifying a forwarding dial plan rule is a PUT method on a " /forwardingDialPlanRules/<forwarding dial plan rule ID>" node

Parameters	Type/Value	Description/Notes
matchPattern	String	The domain to match in order to apply the dial plan rule. Must be a complete domain name (e.g. " example.com") or a “wildcarded” one (e.g. exa*.com). Wildcards are permitted in any part of a domain matching pattern, but do not use “matchPattern=*” as a match all, otherwise you will create call loops.
destinationDomain	String	Calls that are forwarded with this rule will have their destination domain rewritten to be this value

Parameters	Type/Value	Description/Notes
action	**forward** |reject	If set to " forward" causes matching call legs to become point-to-point calls with a new destination. " reject" causes the incoming call leg to be rejected
callerIdMode	**regenerate**| preserve	When forwarding an incoming call to a new destination address, whether to preserve the original calling party's ID or to generate a new one. If this parameter is not supplied in a create (POST) operation, it defaults to " regenerate"
priority	Number	Numeric value used when determining the order in which to apply forwarding dial plan rules; higher values will be applied first
tenant	ID	If a tenant is specified, calls using this rule will be associated with the specified tenant.
uriParameters	**discard**| forward	When forwarding an incoming call to a new destination address, this parameter determines whether to discard any additional parameters that are present in the destination URI of the incoming call, or to forward them on to the destination URI of the outbound call. If this parameter is not supplied in a create (POST) operation, it defaults to " discard". This parameter is present from version 2.0 onwards

7.4.4 Retrieving information on an individual incoming call forwarding dial plan rule

GET method on a " /forwardingDialPlanRules/<forwarding dial plan rule ID>" node. If the forwarding dial plan rule ID supplied is valid, a " 200 OK" response and a single " <forwardingDialPlanRule id=<ID>>" object will be returned with data as per the previous section.

8 Call Related Methods

This chapter details the API methods for:

- active calls
- call profiles
- call legs
- call leg profiles
- call branding profiles
- layout templates
- dtmf profiles
- ivr methods
- ivr branding profiles
- participants

Note: Call objects created with the API remain active after a call is disconnected. To prevent the system becoming overloaded, we recommend that you use the DELETE method to delete the call objects when the call is finished.

8.1 Call Methods

8.1.1 Retrieving Information on Active Calls

GET method performed on the " /calls" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
coSpaceFilter	ID	Supply an ID to return just those calls that match the filter
tenantFilter	ID	Supply an ID to return just those calls that belong to the specified tenant

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <calls total="N"> tag with potentially multiple <call> elements within it. <call> elements follow the general form on the left.
callCorrelator	ID	An id that is the same across all distributed instances of this call.
name	String	The associated (human-readable) name for the call with a maximum length of 200 characters.
coSpace	ID	If the call represents the instantiation of a coSpace, this value will be present and hold the id of the coSpace
tenant	ID	The specific tenant that the call belongs to

8.1.2 Creating a New Call and Modifying an Active call

POST method performed on the "/calls" node or PUT method to "/calls/<call id>

Parameters	Type/Value	Description/Notes
coSpace *	ID	For POST only: Specifies the coSpace for which the call is being instantiated. If not set then " name" must be set. If not set then " name" or " callCorrelator" must be set.
name *	String	For POST only: Specifies the name of the new call which is being instantiated. If not set then " coSpace" must be set. If not set then " coSpace" or " callCorrelator" must be set.
callCorrelator *	String	For POST only: Specifies the callCorrelator id of an existing call on another Call Bridge in the cluster. If not set " coSpace" or " name" must be set. (From version 2.6)
locked	true|false	Allows the locking/unlocking of a meeting lobby in order to control the process of activating participants. Participants requiring activation are typically guests to a coSpace that have not yet been 'activated'. Members of a coSpace are not affected, and can join the coSpace at any time. When a meeting is locked, the guests requiring activation wait in the meeting lobby until the host unlocks the coSpace, at which point they are activated and join the coSpace. Participants that are already activated are NOT deactivated when the conference goes from the unlocked state to the locked state. If set to true, new participants that need activation are not activated even if there are activators in the call.

Parameters	Type/Value	Description/Notes
recording	true|false	If true, this call is currently being recorded.
streaming	true|false	If true, this call is currently being streamed. (From version 2.1)
allowAllMuteSelf	**true**|false	If true, participants have the permission to mute and unmute themselves.
allowAllPresentationContribution	true|**false**	If true, participants have the permission to present. If false, this permission is dependent on presentationContributionAllowed in the call leg profile. Default is false.
joinAudioMuteOverride	true|false	If true, new participants will be muted when joining the call. If false, new participants will be unmuted when joining the call. If unset, new participants will use the audio mute value from the call leg profile.
messageText	String	Text to display to every participant in the call (only displayed if configured messageDuration is non zero). (From version 2.1)
messagePosition	top| middle| bottom	Position to display configured messageText on screen (for SIP endpoints).(From version 2.1)
messageDuration	Number| permanent	Time in seconds to display configured messageText on screen. Typing the string “permanent” will result in the string being permanently shown until it is reconfigured. (From version 2.1)
activeWhenEmpty	**true**|false	If true, this call is considered “active for load balancing” when no participants are present. This means that the first call to the empty conference is preferentially load balanced. You can prevent the load balancing preferentially using the empty conferences by setting this parameter to false. If this parameter is not supplied in a create (POST) operation, it defaults to “true”. (From version 2.2)
panePlacementHighestImportance	Number	If panePlacementHighestImportance is provided, pane placement will be activated for this active call- the active range of importance values will be from " highest importance" down to 1(inclusive). If set, this will override the coSpace counterpart. (From version 2.7)

Parameters	Type/Value	Description/Notes
panePlacementSelfPaneMode	**skip**|self|blank |<unset>	If panePlacementSelfPaneMode is provided, the self pane mode will be activated for this active call. skip - same as the pre-2.7 version behavior, on a per-viewer basis the screen layout skips the self pane and displays the pane of the next important participant (default). blank - leaves a blank pane instead of displaying the important participant, so the important participant still sees the other participants in the same pane position as all other viewers. <unset> - follows this order of precedence: - use the value set for panePlacementHighestImportance on /calls, - if panePlacementHighestImportance on /calls is unset, then use the value set for panePlacementHighestImportance on /coSpace (if the call is to a space), - if panePlacementHighestImportance on /coSpace is also unset, then it reverts to the skip behavior defined above. By default, panePlacementSelfPaneMode is set to <unset>. If set, this will override the coSpace counterpart. (From version 2.7)
messageBannerText	String	The string is the message to be displayed on the screen. The default value is an empty string, which does not display the message banner. (From version 3.2)
chatAllowed	**true** | false |	• true - Indicates that chat is allowed in the call. • false - Indicates that chat is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to true.
raiseHandEnabled	**true** | false	An administrator can enable or disable the feature for the whole call. By default, the parameter is <**unset**>, but if unset at all levels in the call / callProfile hierarchy, then it defaults to **true**. (From version 3.2)

Parameters	Type/Value	Description/Notes
panePlacementActiveSpeakerMode	allowself | suppressself | **none**	• allowself – displays the participant in the first pane if they are the active speaker, additional to displaying the participant in the pane set by pane placement. The same view is displayed for all participants. • suppressself – displays the active speaker in the first pane for all other participants other than the active speaker. For the active speaker, the previous speaker is displayed in the first pane. • none – the feature is disabled.
notesAllowed	true| **false**	• true – Indicates that notes is allowed in the call. • false – Indicates that notes is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
captionsAllowed	true | **false**	• true – Indicates that captions are allowed in the call. • false – Indicates that captions are not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
backgroundBlurAllowed	true | **false**	• true – Indicates that background blur is allowed in the call. • false – Indicates that background blur is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false

Parameters	Type/Value	Description/Notes
fileReceiveAllowed	true | **false**	• true – Indicates that file sharing is allowed in the call. • false – Indicates that file sharing is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
logoFileName	String	Name of the image file that is uploaded using SFTP with file name restricted to 128 characters.
logoPosition	**leftTop**| leftBottom| rightTop| rightBottom	The position where the logo needs to be rendered on the recipient screen. If unset at all levels, the logo position defaults to **`leftTop`**.
surveyAllowed	`true`| **`false`**	True – Indicates that surveys are allowed in the meeting and participants can take survey. False – Indicates that surveys are not allowed in the meeting. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

If a call is instantiated for the coSpace successfully, that call's ID is returned in the "Location" field of the response header. You can now add call legs to this call as described below.

8.1.3 Retrieving Information on an Individual Active Call

GET method performed on a "/calls/<call id>" node. If the call ID supplied is valid, a "200 OK" response is received, with XML content of the form:

Response elements	Type/Value	Description/Notes
coSpace	ID	If the call represents the instantiation of a coSpace, this value will be present and hold the id of the coSpace
callCorrelator	ID	An id that is the same across all distributed instances of this call.

Response elements	Type/Value	Description/Notes
callType	coSpace| forwarding| adHoc| lyncConferencing	Indicates the call type: coSpace – this call is the instantiation of a space forwarding – this is a forwarded or "gateway" call adHoc – this is an ad hoc multiparty call lyncConferencing – this call leg is participating in a Lync conference (From version 2.3)
tenant	ID	The ID of the tenant that owns the call
durationSeconds	Number	The duration of the call, as the number of seconds since the call started
name	String	The associated human-readable name for the call with a maximum length of 200 characters. (From version 2.3)
numCallLegs	Number	The number of call legs currently active within this call
maxCallLegs	Number	The highest number of call legs that have been simultaneously present within this call
numParticipantsLocal	Number	The number of participants within this call locally hosted by the Call Bridge to which the request is being made
numParticipantsRemote	Number	The number of participants in this call hosted by other Call Bridges
numDistributedInstances	Number	The number of other Call Bridges hosting participants in this call
presenterCallLeg	ID	The presenterCallLeg value is only present if a call leg is actively presenting within this call
locked	true|false	Indicates whether the call is locked (true) or unlocked (false).
recording	true|false	If true, this call is configured to be recorded.
streaming	true|false	If true, this call is configured to be streamed. (From version 2.1)
recordingStatus	true|false	if true, this call is being recorded. (From version 2.4)
allowAllMuteSelf	true|false	If true, participants have the permission to mute and unmute themselves. If false, this permission is dependent on muteSelfAllowed in the call leg profile.

Response elements	Type/Value	Description/Notes
allowAllPresentationContribution	true|**false**	If true, participants have the permission to present. If false, this permission is dependent upon presentationContributionAllowed in the call leg profile. Default is false.
joinAudioMuteOverride	true|false	If true, new participants will be muted when joining the call. If false, new participants will be unmuted when joining the call. If unset, new participants will use the audio mute value from the call leg profile.
messageText	String	Text to display to every participant in the call (only displayed if configured messageDuration is non zero). (From version 2.1)
messagePosition	top| middle| bottom	Position to display configured messageText on screen (for SIP endpoints). (From version 2.1)
messageDuration	Number	Time in seconds to display configured messageText on screen (can also be " permanent" to cause the message to be displayed until it is reconfigured). (From version 2.1)
messageTimeRemaining	Number	Time remaining in seconds for configured messageText to be displayed on screen. (From version 2.1)
ownerName	String	If set, displays the owner of this call. This can be the meetingScheduler of the coSpace of this call, or the name of the owner of this call or the Jid of the owner. (From version 2.2)
activeWhenEmpty	**true**|false	If true, this call is considered active for load balancing when no participants are present. If this parameter is not supplied in a create (POST) operation, it defaults to “true”. (From version 2.2)
endpointRecording	true|false	If set, one of the participants in this call is recording the conference externally (currently only flags up Skype or Lync clients recording the conference). (From version 2.4)
lyncAudienceMute	true|false	Whether the audience was muted by a Skype or Lync client. Only present if this call is a Skype/Lync conference. (From version 2.4)

Response elements	Type/Value	Description/Notes
panePlacementHighestImportance	Number	If provided, pane placement will be activated for this active call - the active range of importance values will be from " highest importance" down to 1(inclusive). (From version 2.7)
panePlacementSelfPaneMode	**skip**|self|blank |<unset>	Indicates if self pane mode is activated for this active call. skip - same as the pre-2.7 version behavior, on a per-viewer basis the screen layout skips the self pane and displays the pane of the next important participant (default). blank - leaves a blank pane instead of displaying the important participant, so the important participant still sees the other participants in the same pane position as all other viewers. <unset> - follows this order of precedence: use the value set for panePlacementHighestImportance on /calls, if panePlacementHighestImportance on /calls is unset, then use the value set for panePlacementHighestImportance on /coSpace (if the call is to a space), if panePlacementHighestImportance on /coSpace is also unset, then it reverts to the skip behavior defined above. By default, panePlacementSelfPaneMode is set to <unset>. (From version 2.7)
coSpaceMetadataConfigured	true | false	If the call represents the instantiation of a coSpace this value will be present and is true if coSpace has metadata configured.

Response elements	Type/Value	Description/Notes
chatAllowed	**true** | false	• true – Indicates that chat is allowed in the call. • false – Indicates that chat is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to true.
raiseHandEnabled	**true** | false	If set, whether or not participants are allowed to raise their hands in this call.
messageBannerText	String	The string is the message to be displayed on the screen. The default value is an empty string, which does not display the message banner. (From version 3.2)
panePlacementActiveSpeakerMode	allowself | suppressself | **none**	Indicates if the active speaker will be displayed for this coSpace. • allowself – displays the participant in the first pane if they are the active speaker, additional to displaying the participant in the pane set by pane placement. The same view is displayed for all participants. • suppressself – displays the active speaker in the first pane for all other participants other than the active speaker. For the active speaker, the previous speaker is displayed in the first pane. • none – the feature is disabled.
notesAllowed	true| **false**	• true – Indicates that notes is allowed in the call. • false – Indicates that notes is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

Response elements	Type/Value	Description/Notes
captionsAllowed	true | **false**	• true – Indicates that captions are allowed in the call. • false – Indicates that captions are not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
backgroundBlurAllowed	true | **false**	• true – Indicates that background blur is allowed in the call. • false – Indicates that background blur is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false
fileReceiveAllowed	true | **false**	• true – Indicates that file sharing is allowed in the call. • false – Indicates that file sharing is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
logoFileName	String	Name of the image file that is uploaded using SFTP with file name restricted to 128 characters.
logoPosition	**leftTop**| leftBottom| rightTop| rightBottom	The position where the logo needs to be rendered on the recipient screen. If unset at all levels, the logo position defaults to **`leftTop`**.
surveyAllowed	`true`| **`false`**	True – Indicates that surveys are allowed in the meeting and participants can take survey. False – Indicates that surveys are not allowed in the meeting. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

8.1.4 Generating diagnostics for an individual call

POST method performed on “/calls/<call id>diagnostics" generates call diagnostics for the call in question.

8.1.5 Retrieve participants in a conference

GET method performed on the " /calls/<call id>/participants" node. Retrieves a list of all of the participants associated with the specified call.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve active participants other than the first " page" in the notional list (see Section 4.2.2).
coSpaceFilter	ID	Supply an ID to return just those active participants that match the filter.
tenantFilter	ID	Supply an ID to return just those active participants that belong to the specified tenant.
callBridgeFilter	ID	Supply an ID to return just those active participants located on the specified Call Bridge.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <participants total="N"> tag with potentially multiple <participant> elements within it. <participant> elements follow the general form on the left.
name	String	The associated (human-readable) name associated with this participant. It can have a maximum length of 200 characters.
call	ID	The call that this participant is part of
tenant	ID	The specific tenant with which this participant is associated
callBridge	ID	The remote, clustered Call Bridge to which this participant is connected

8.1.6 Creating a new participant for a specified call

POST method on the " /calls/<call id>/participants" node.

Note: Due to load balancing across clustered Meeting Servers, an explicit selection of a Call Bridge or Call Bridge Group or from configured dial plan rules, may result in the call leg instantiation (" owned" by the participant object) occurring on a remote clustered Call Bridge.

Note: See also the section on participant related methods Section 8.11.

Parameters	Type/Value	Description/Notes
remoteParty * or movedParticipant*	String ID	For POST only, remoteParty specifies the participant's address; this could be a SIP URI or a phone number to invite that user to the call. remoteParty makes a call out to the specified URI from the conference corresponding to <call id>. From 2.6, if remoteParty is not specified then movedParticipant must be specified instead. For POST only, movedParticipant specifies the ID of the participant to be moved to the conference corresponding to <call id>. Both callBridge and callBridgeGroup parameters are ignored, the call is hosted on the callBridge or callBridgeGroup that the moved participant is homed on. (From version 2.6) Note: If both remoteParty and movedParticipant are provided, then the movedParticipant parameter will be ignored.
bandwidth	Number	For POST only, if supplied, sets the bandwidth for the participant, in bits per second (e.g. 2000000 for 2Mbit/s). If not supplied, the Call Bridge configured value will be used.
confirmation	true|false	For POST only, if supplied, this overrides the automatic choice of whether to require a confirmation from the remote party to join the call true – always require a confirmation from the remote party; typically this takes the form of a voice prompt requiring them to hit a key to join false – never require a confirmation from the remote party; the remote party will be joined into the coSpace when they accept the incoming call
ownerId	ID	If supplied must be an ID for the Meeting Server to associate with this participant. This will be returned by the Meeting Server when the participant's call leg is later queried and therefore should be a value that has meaning to the requestor.
callLegProfile	ID	If provided, associates the specified call leg profile with this participant's call leg.

Parameters	Type/Value	Description/Notes
		Supply any of these parameters to override the call leg profile values for this call leg.
needsActivation		
defaultLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	If left blank, then the setting at the /callLegProfile associated with this /calls/<call id>/participants is used.
participantLabels		
presentationDisplayMode		
presentationContributionAllowed	true|false	
presentationViewingAllowed	true|false	
endCallAllowed	true|false	
muteOthersAllowed	true|**false**	
videoMuteOthersAllowed	true|**false**	
muteSelfAllowed	**true**|false	
videoMuteSelfAllowed	**true**|false	
changeLayoutAllowed	true|false	
joinToneParticipantThreshold	true|false	
leaveToneParticipantThreshold		
videoMode		
rxAudioMute		
txAudioMute		
rxVideoMute		
txVideoMute		
sipMediaEncryption		

Parameters	Type/Value	Description/Notes
audioPacketSizeMs		
deactiviationMode		
deactivationModeTime		
telepresenceCallsAllowed		
sipPresentationChannelEnabled	true|false	
bfcpMode	serverOnly | serverAndClient	If presentation video channel operations are enabled for SIP calls, this setting determines the BFCP behavior: serverOnly – this is the normal setting for a conferencing device, and is intended for use with BFCP client mode devices (e.g. SIP endpoints) serverAndClient – allows the Meeting Server to operate in either BFCP client or BFCP server mode in calls with remote devices – this can allow improved presentation video sharing with a remote conference-hosting device such as a third-party MCU
layout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	This parameter is provided for backward compatibility with very early versions of Acano server – it has the equivalent functionality to changing " defaultLayout" , and if both are supplied then " defaultLayout" will take precedence.
disconnectOthersAllowed	true|false	
addParticipantAllowed	true|false	(From version 2.3)

Parameters	Type/Value	Description/Notes
qualityMain	**unrestricted**| max1080p30| max720p30| max480p30	Restricts the maximum negotiated main video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and framerate. Note that call legs may operate at lower resolutions or framerates due to endpoint limitations or overall Call Bridge load unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or frame rate max1080p30 – restricts the bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources, for example 1280x720 screen size at 60 frames per second max720p30 – restricts the bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources max480p30 – restricts the bridge to negotiating at most 868x480 screen size at 30 frames per second or equivalent transcoding resources. (From version 2.2) **Note**: This parameter does not apply to incoming video from Cisco Meeting App or Cisco Meeting App WebRTC app. **Note**: For calls between the Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to theMeeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences.

Parameters	Type/Value	Description/Notes
qualityPresentation	**unrestricted**| max1080p30| max720p5	Restrict the maximum negotiated presentation video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and frame rate. This only affects legs which use a separate presentation stream. unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or framerate max1080p30 – restricts the Call Bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources max720p5 – restricts the Call Bridge to negotiating at most 1280x720 screen size at 5 frames per second or equivalent transcoding resources. (From version 2.2)Restricts the maximum negotiated main video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and framerate. Note that call legs may operate at lower resolutions or framerates due to endpoint limitations or overall Call Bridge load unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or frame rate max1080p30 – restricts the bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources, for example 1280x720 screen size at 60 frames per second max720p30 – restricts the bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources max480p30 – restricts the bridge to negotiating at most 868x480 screen size at 30 frames per second or equivalent transcoding resources. (From version 2.2)

Parameters	Type/Value	Description/Notes
		Note: This parameter does not apply to incoming video from Cisco Meeting App or Cisco Meeting App WebRTC app. Note: For calls between the Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to theMeeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences.
participantCounter	never| auto| always	Controls the behavior of the on-screen participant counter. never - never show an on-screen participant count value auto - show the on-screen participant count value when appropriate; typically this will be to indicate that there are additional participants present that you cannot currently see always - always show the on-screen participant count value (From version 2.2)
callBridge	ID	If supplied, attempt to add the participant from the specified Call Bridge (from version 2.2). If a participant is moved between conferences using the movedParticipant parameter then the callBridge parameter is ignored. (From version 2.6)
callBridgeGroup	ID	If supplied, attempt to add the participant from the specified Call Bridge Group (from version 2.2). If a participant is moved between conferences using the movedParticipant parameter then the callBridgeGroup parameter is ignored. (From version 2.6)
importance	Number	The importance value of the participant to be created. Maximum value is 2,147,483,647. To remove importance leave the importance parameter as unset (leave value as blank). (From version 2.2)

Parameters	Type/Value	Description/Notes
nameLabelOverride	String (maximum of 50 bytes of UTF-8)	If supplied, overrides the name for this participant. Setting an empty string clears the value and restores the original name. Overriding the name of a participant and its associated call leg(s) is interchangeable and affects both; the latest change takes precedence. It changes the name of the participant in the following: - on-screen name label viewed by other conference participants, - ActiveControl roster list, - any place that the web app sees the name of the participant in a call, - CDR records, - where the name appears in the web interface. (From version 2.4)
dtmfSequence	String	A sequence of DTMF key press commands to send to the far end either when the participant is initially created or during the call. The DTMF sequence is played out from the Call Bridge where the call for this participant is placed. In the supplied sequence, you can use the digits 0 to 9, * and #, as well as one or more comma characters (" ,") which add a pause between digits. (From version 2.4)
controlRemoteCameraAllowed	**true**|false	Allows the participant to use Far End Camera Control (FECC). (From version 2.8)
layoutTemplate	ID	If specified, associates the layout template for the participant. (From version 2.8)

Parameters	Type/Value	Description/Notes
audioGainMode	disabled|**agc**	Specifies whether audio for this participant should use automatic gain control or not. (From version 2.8) Note: - AGC will be applied to any endpoint (physical endpoints or soft clients) connected directly to the Meeting Server. It will not be applied to TIP calls or AVMCU (because this is a mixed audio stream). - Skype participants connected to AVMCU will not be subject to any AGC as the AVMCU controls the audio - AGC is not applied to distribution links between Meeting Servers because this is a mixed audio stream.
deactivated	true | false	• true - Participant will wait in the lobby or can be moved to the lobby From version 3.4. • false - Participants are allowed into the meeting from the lobby.
meetingTitlePosition	top | middle | **bottom**	Enables and places the meeting title at the specified position. If unspecified, it takes the value bottom. (From version 3.2)
chatContributionAllowed	**true** | false	Determines whether or not the participant can send chat messages. If unset at all levels of the hierarchy, then it defaults to true (From version 3.2)
handStatus	string	If set, indicates whether the hand is raised or lowered for this participant or call leg. The value is not returned if the handStatus was not changed during the call. (From version 3.2)
accessMethod	ID	Associates the specified **`accessMethod`** as the access method for the callLeg/participant and overrides any default or primary access method on the coSpace.

Parameters	Type/Value	Description/Notes
changeRoleAllowed	`true`, **`false`**, or `<unset>`	Determines whether or not the participant can change the role of another participant in a call when using the web app. When set to **true**, participant can change the role of other participants. When set to **false**, participants cannot change the role of other participants. For details on permissions to change the participant role, refer to Appendix A.+ The usual rules for the hierarchy of call leg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
noteContributionAllowed	true| **false**	Determines whether or not the participant can publish notes. If unset at all levels of the hier-archy, then it defaults to false.
captionContributionAllowed	true| **false**	Determines whether or not the participant can send captions in a meeting. The usual rules for the hierarchy of callLeg and callLeg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false. **Note:** If **`captionsAllowed`** is set to **false** at the call level, the participant cannot send closed captions even if **`captionContributionAllowed`** is set as **true**.
audioPrompt	On/Off	• On – Blast dial audio prompt will be played, and user has to enter DTMF key to join the conference. • Off – Blast dial audio prompt will not be played, and user doesn't have to enter any DTMF key to join the conference. This parameter is only supported on POST methods.
fileUploadAllowed	true | false	• true - Indicates that the participant can share files in the call. • false - Indicates that the participant cannot share files in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

Parameters	Type/Value	Description/Notes
surveyOpsAllowed	`true`| **`false`**	True – Indicates that the participant can create/launch/delete/view results of surveys. False – Indicates that the participant cannot create/launch/delete/view results of surveys. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

8.1.7 Set properties for all participants in a conference

PUT to " /calls/<call id>/participants/*" node. Set properties for all participants associated with the specified call.

Parameters	Type/Value	Description/Notes
rxAudioMute	true|false	If true, mute the receiving of audio from all endpoints.
txAudioMute	true|false	If true, mute the transmission of audio from all endpoints.
rxVideoMute	true|false	If true, mute (block) the receiving of video from all endpoints.
txVideoMute	true|false	If true, mute (block)the transmission of video from all endpoints.
layout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	Set the layout for all endpoints.
importance	Number	Set importance of all participants. Maximum value is 2,147,483,647. To remove importance leave the importance parameter as unset (leave value as blank). (From version 2.2)
filterIds		Optional comma-separated list of up to 20 participant ids to be either included or excluded from this operation (depending on value of 'mode' parameter. operation See More information on using bulk operation on participants below. (From version 2.4)

Parameters	Type/Value	Description/Notes
mode	**exclude**| selected	exclude - the participant ids in filterIds are excluded from the operation selected - only the participant ids in filterIds are included in the operation See More information on using bulk operation on participants below. If this parameter is not supplied in a create (POST) operation, it defaults to " exclude." (From version 2.4)
deactivated	true/false	• true - Participant will wait in the lobby or can be moved to the lobby. • false - Participants are allowed into the meeting from the lobby.

8.1.7.1 More information on using bulk operation on participants

PUT to **`/calls/<call id>/participants/*?filterIds=<id1>,<id2>&mode=(exclude|selected)`**

Mode	filterIds	Notes
exclude	empty (no ids)	This is the default settings, equivalent to the behavior prior to version 2.4. The operation will act on all of the participants in the selected call.
exclude	one or more ids	The operation will act on all of the participants in the selected call, except those listed.
selected	empty (no ids)	This will have no impact, as no ids are suppllied.
selected	one or more ids	The operation will only act on the selected participants in the call.

The maximum size of a list is fixed at 20, trying to include more ids than this will generate an error.

Return values are for acceptance of operation, failure or success of individual participants will not be returned.

For example:

PUT to **`/calls/<call id>/participants/*?filterIds=<smith>,<green>&mode=exclude`**

will have the effect that all participants that match filter ids of smith or green will be excluded from the bulk operation.

Errors:

`callDoesNotExist` call ID does not exist,

If more than 20 filterIds are included in the filter id list, then. a parameterError is generated with error attribute equal to " valueTooLong" .

8.1.8 Send closed captions using third party API tools

POST method on **`calls/<call id>/captions/`** enables third party API tools to send captions in the meeting. Only web app participants can view these captions.

Parameters	Type/Value	Description/Notes
captionsText	String	The text to be displayed as caption on the screen in the meeting.

8.2 Call Profile Methods

Call profiles control the maximum number of active simultaneous participants and the in-call experience for SIP (including Lync) calls. For more information see also Section 16.

8.2.1 Retrieving call profiles

GET method performed on the "/callProfiles" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
usageFilter	unreferenced| referenced	Supply " usageFilter=unreferenced" in the request to retrieve only those call profiles that are not referenced by global settings or any other object. This is a useful check before deleting the profile. To retrieve just those call profiles which are referenced in at least one place, you can supply " usageFil-ter=referenced"
name	String	Name of the **`callProfiles`**.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <callProfiles total=" N" > tag with potentially multiple <callProfile> elements within it. Each <callProfile> tag may include the following elements: See the next section

8.2.2 Setting up and modifying call profiles

- Creating: POST method to the " /callProfiles" node
- Modifying: PUT to " /callProfiles/<call profile id>"

Parameters	Type/Value	Description/Notes
participantLimit	Number	Sets the maximum number of participants for calls (coSpace instantiations or ad hoc calls) using this call profile that can be active simultaneously; new participants beyond this limit are not permitted
locked	true|false	Allows the locking/unlocking of a meeting lobby in order to control the process of activating participants. Participants requiring activation are typically guests to a coSpace that have not yet been 'activated'. Members of a coSpace are not affected, and can join the coSpace at any time. When a meeting is locked, the guests requiring activation wait in the meeting lobby until the host unlocks the coSpace, at which point they are activated and join the coSpace. Participants that are already activated are NOT deactivated when the conference goes from the unlocked state to the locked state. If set to true, new participants that need activation are not activated even if there are activators in the call.
recordingMode	disabled| **manual**| automatic	Controls how this coSpace or ad hoc call can be recorded • disabled – call is not recorded • manual – users can start/stop recording • automatic – call is automatically recorded and users cannot start/stop recording If this parameter is not supplied in a create (POST) operation, it defaults to " manual" .
lockMode	all|needsActivation	Defines the behavior of locking a call • all – when the call is locked any new participants won't be admitted into the meeting and will be in the lobby, this includes participants that don't need activation • needsActivation – when the call is locked new participants that don't need activation will go into the call however new participants that need activation will go into the lobby. Participants that are memebers of the cospace will bypass the lock and enter the call even if they require activation as long as there is an activator already in the call (From version 2.9)
sipRecorderUri	string	The SIP recorder dial out URI. (From version 2.9)

Parameters	Type/Value	Description/Notes
sipStreamerUri	string	The SIP streamer dial out URI. (From version 3.0)
streamingMode	disabled| manual| automatic	Controls how this coSpace or ad hoc call can be streamed • disabled - call is not streamed • manual - users can start/stop streaming • automatic - call is automatically streamed and users cannot start/stop streaming If this parameter is not supplied in a create (POST) operation, it defaults to " manual" .(From version 2.1)
passcodeMode	required| timeout	Determines the behavior for passcode entry when a mixture of blank and set passcodes can be used to access a coSpace via the same URI/call Id. • required - requires passcode to be entered, with blank passcode needing to be explicitly entered • timeout - after an amount of time has elapsed with no passcode being entered, interpret this as a blank passcode. Amount of timeout is determined by value of " passcodeTimeout"
passcodeTimeout	numeric	If specified, this is the amount of time, in seconds, that the Call Bridge will wait before before interpreting passcode as a blank passcode (if " passcodeMode" is set to " timeout"). Timeout time is measured from the end of the passcode prompt.
gatewayAudioCallOptimization	true|false	If set to true, outgoing gateway call legs will be audio-only if the incoming call leg was audio-only. (From version 2.3)
lyncConferenceMode	dualHomeCluster| dualHomeCallBridge| gateway	Defines the behavior of the Call Bridge when connecting participants to Lync conferences. • dualHomeCluster - all the Call Bridges will share the same conference • dualHomeCallBridge - each Call Bridge will host their own conference and each will connect to the AVMCU • gateway - each participant will have dedicated connection to the Lync AVMCU server (From version 2.3)

Parameters	Type/Value	Description/Notes
muteBehavior	**linked** | separate	Defines the mute behavior of the call. • linked – in this mode, when a user's call is muted on the Meeting Server, their endpoint or web app session may also automatically perform a local mute of their device; this means it is not possible to reverse the effect of the Meeting Server mute with just another API command - the user themselves must unmute their device. • separate – in this mode, the mute statuses of a user's call on the Meeting Server and on their local device are independent of one another, meaning that other users/admins can video/audio mute/unmute all participants. If this parameter is not supplied in a create (POST) operation, it defaults to " linked" . (From version 3.1)
messageBannerText	String	The string is the message to be displayed on the screen. The default value is an empty string, which does not display the message banner.
chatAllowed	**true** | false	• true - Indicates that chat is allowed in the call. • false - Indicates that chat is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to true.
raiseHandEnabled	**true** | false	An administrator can enable or disable the feature for the whole call. By default, the parameter is <unset>, but if unset at all levels in the call / callProfile hierarchy, then it defaults to true.
notesAllowed	true | **false**	• true - Indicates that notes is allowed in the call. • false - Indicates that notes is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

Parameters	Type/Value	Description/Notes
captionsAllowed	true | **false**	• true – Indicates that captions are allowed in the call. • false – Indicates that captions are not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
backgroundBlurAllowed	true | **false**	• true – Indicates that background blur is allowed in the call. • false – Indicates that background blur is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false
fileReceiveAllowed	true | **false**	• true – Indicates that file sharing is allowed in the call. • false – Indicates that file sharing is not allowed in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
logoFileName	String	Name of the image file that is uploaded using SFTP with file name restricted to 128 characters.
logoPosition	**leftTop**| leftBottom| rightTop| rightBottom	The position where the logo needs to be rendered on the recipient screen. If unset at all levels, the logo position defaults to **`leftTop`**.
surveyAllowed	`true`| **`false`**	True – Indicates that surveys are allowed in the meeting and participants can take survey. False – Indicates that surveys are not allowed in the meeting. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
name	String	Name of the **`callProfiles`**.

8.2.3 Retrieving detailed information about an individual call profile

GET method performed on a "/callProfiles/<call profile id>" node. If the call profile id ID supplied is valid, a "200 OK" response is received, with XML content matching the section above.

8.3 Call Leg Methods

8.3.1 Retrieving Information on Active Call Legs

GET method performed on the "/callLegs" node (to retrieve information on all active call legs within the system).

Alternatively, a GET method performed on the "/calls/<call id>/callLegs" node (to retrieve information on active call legs for a specific call).

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (seeSection 4.2.2).
filter	String	Supply filter=<string> in the URI to return just those call legs that match the filter
participantFilter	ID	Supply participantFilter to return only those call legs associated with the specified participant.
tenantFilter	ID	Supply tenantFilter to return only those call legs associated with the specified tenant.
activeLayoutFilter	String	If supplied, this filter will restrict results returned to those call legs using the specified layout.
availableVideoStreamsLowerBound	Number	If supplied, this filter will restrict results returned to those call legs with this many or more available video streams.
availableVideoStreamsUpperBound	Number	If supplied, this filter will restrict results returned to those call legs with many or fewer available video streams.
ownerIdSet	true|false	Used to return those call legs that have an owner Id set, or those that do not

Parameters	Type/Value	Description/Notes
alarms	Text	Used to return just those call legs for which the specified alarm names are currently active. Either “all”, which covers all supported alarm conditions, or one or more specific alarm conditions to filter on, separated by the ‘|’ character. The supported alarm names are: • packetLoss – packet loss is currently affecting this call leg • excessiveJitter – there is currently a high level of jitter on one or more of this call leg’s active media streams • highRoundTripTime – the Meeting Server measures the round trip time between itself and the call leg destination; if a media stream is detected to have a high round trip time (which might impact call quality), then this alarm condition is set for the call leg

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <callLegs total=”N”> tag with potentially multiple <callLeg> elements within it. <callLeg> elements follow the general form on the left.
callLeg id	ID	
name	String	
remoteParty	String	
call	ID	
tenant	ID	
<alarms>		For call legs which have active alarm conditions, there will be an additional “<alarms>” tag under the encompassing “<callLeg>” which details the currently active alarms. Within this “<alarms>” tag there will be one or more subsidiary indications—also see the note below:
packetLoss	String	Present if packet loss is being experienced on one or more of the call leg’s active media streams
excessiveJitter	String	Present if there is a high level of jitter on one or more of the call leg’s active media streams

Response elements	Type/Value	Description/Notes
highRoundTripTime	String	Present if a high round trip time has been detected for one or more of the call leg's media streams
accessMethod	ID | " coSpace"	Returns the access method used to join the call on dial in or as set on dial out. Where an accessMethod is not specified on the POST operation, the GET still returns an access method ID if defaultAccessMethod was configured on the coSpace or if the primary access method was employed. The API will return " coSpace" if the coSpace was not joined through an access method. This could be on dial in or on dial out if no accessMethod or defaultAccessMethod is specified and no primary access method exists.

Note on alarms:
Call leg alarms provide information that may be useful in raising alarms or troubleshooting issues after they have occurred but they should not necessarily be treated as if they are alarm conditions in themselves—unlike system level alarms.

A call leg alarm may be triggered by a number of factors, not necessarily a set percentage packet loss for example. An alarm condition is attached to a call leg when the Meeting Server believes that the call leg may be being degraded. These " conditions" may include a simple threshold, but potentially other things too such as a more adaptive threshold and taking other factors into account. This does not necessarily mean that the user's experience was poor but it provides information to troubleshoot in the event that it was. Therefore you could consider adding filters to this alarm information and deciding when to flag an event as an alarm to the operator (i.e. setting your own thresholds) and/or storing call leg alarm information alongside CDRs so that if a user reports a poor quality call you can retrieve this information after the event to determine what the cause might have been.

8.3.2 Adding and Modifying Call Legs

- Adding: POST method to a "/calls/<call ID>/callLegs" node. The <call ID> is learnt from a GET on "/calls" or from a newly created call (see Creating a new call above). If a profile has been applied to this call leg, it starts with the values set in the profile. Note: these added or modified call legs will not be load balanced across clustered Meeting Servers.
- Modifying: PUT method performed on a "/callLegs/<callLeg ID>" node. It makes live, dynamic, changes to an in-progress connection to a remote party.

Note: You cannot modify the remoteParty, bandwidth or confirmation.

Parameters	Type/Value	Description/Notes
remoteParty *	String	For POST only, specifies the call leg's address; this could be a SIP URI or a phone number to invite that user to the call.
bandwidth	Number	For POST only, if supplied, sets the bandwidth for the call leg, in bits per second (e.g. 2000000 for 2Mbit/s). If not supplied, the Call Bridge configured value will be used.
confirmation	true|false	For POST only, if supplied, this overrides the automatic choice of whether to require a confirmation from the remote party to join the call true - always require a confirmation from the remote party; typically this takes the form of a voice prompt requiring them to hit a key to join false - never require a confirmation from the remote party; the remote party will be joined into the coSpace when they accept the incoming call
ownerId	ID	If supplied must be an ID for the Meeting Server to associate with this call leg. This will be returned by the Meeting Server when the call leg is later queried and therefore should be a value that has meaning to the requestor.
chosenLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	This parameter overrides the prevailing default layout for this call leg.
dtmfSequence	string	A sequence of DTMF key press commands to send to the far end either when the call leg initially connects, or during the call. In the supplied sequence, you can use the digits 0 to 9, * and #, as well as one or more comma characters (" ,") which add a pause between digits.

Parameters	Type/Value	Description/Notes
callLegProfile	ID	If provided, associates the specified call leg profile with this call leg. You can also supply individual values for all parameters that can be part of a call leg profile to override this call leg profiles' value. See below
		Supply any of these parameters to override the call leg profile values for this call leg.
needsActivation		
defaultLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	If provided changes the call leg to use the specified video stream layout. **Note:** that the callLegProfile " defaultLayout" parameter replaces an equivalent " layout" parameter. If both are supplied, defaultLayout takes precedence.
participantLabels		
presentationDisplayMode		
presentationContributionAllowed	true|false	
presentationViewingAllowed	true|false	
endCallAllowed	true|false	
muteOthersAllowed	true|**false**	
videoMuteOthersAllowed	true|**false**	
muteSelfAllowed	**true**|false	
videoMuteSelfAllowed	**true**|false	
changeLayoutAllowed	**true**|false	
joinToneParticipantThreshold	Number	
leaveToneParticipantThreshold		
videoMode		
rxAudioMute		

Parameters	Type/Value	Description/Notes
txAudioMute		
rxVideoMute		
txVideoMute		
sipMediaEncryption		
audioPacketSizeMs		
deactiviationMode		
deactivationModeTime		
telepresenceCallsAllowed	true|false	
sipPresentationChannelEnabled	true|false	**Note**: The sipPresentationChannelEnabled parameter should not be modified in an active call.
bfcpMode	true|false	
layout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	This parameter is provided for backward compatibility with very early versions of Acano server - it has the equivalent functionality to changing " defaultLayout" , and if both are supplied then " defaultLayout" will take precedence.
disconnectOthersAllowed	true|false	
addParticipantAllowed	true|false	

Parameters	Type/Value	Description/Notes
qualityMain	**unrestricted**| max1080p30| max720p30| max480p30	Restricts the maximum negotiated main video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and framerate. Note that call legs may operate at lower resolutions or framerates due to endpoint limitations or overall Call Bridge load. (From version 2.2) unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or frame rate max1080p30 – restricts the bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources, for example 1280x720 screen size at 60 frames per second max720p30 – restricts the bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources max480p30 – restricts the bridge to negotiating at most 868x480 screen size at 30 frames per second or equivalent transcoding resources. **Note**: For calls between the Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to the Meeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences. **Note**: This parameter does not apply to incoming video from Cisco Meeting Appor Cisco Meeting App WebRTC app.

Parameters	Type/Value	Description/Notes
qualityPresentation	**unrestricted**| max1080p30| max720p5	Restrict the maximum negotiated presentation video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and frame rate. This only affects legs which use a separate presentation stream. (From version 2.2) unrestricted - this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or framerate max1080p30 – restricts the Call Bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources max720p5 – restricts the Call Bridge to negotiating at most 1280x720 screen size at 5 frames per second or equivalent transcoding resources. **Note**: For calls between the Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to the Meeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences. **Note**: This parameter does not apply to incoming video from Cisco Meeting Appor Cisco Meeting App WebRTC app.
participantCounter	never| auto| always	Controls the behavior of the onscreen participant counter. (From version 2.2) never – never show an onscreen participant count value auto – show the onscreen participant count value when appropriate. Typically this will be to indicate that there are additional participants present that you cannot currently see. always - always show the onscreen participant count value

Parameters	Type/Value	Description/Notes
nameLabelOverride	String (maximum of 50 bytes of UTF-8)	If supplied, overrides the name for this call leg. Setting an empty string clears the value and restores the original name. Overriding the name of a participant and its associated call leg (s) is interchangeable and affects both; the latest change takes precedence. It changes the name of the participant in the following: - on-screen name label viewed by other conference participants, - ActiveControl roster list, - any place that the web app sees the name of the participant in a call, - CDR records, - where the name appears in the web interface. (From version 2.4)
controlRemoteCameraAllowed	**true**|false	Determines if call legs are allowed to control a remote participant's camera (using FECC). (From version 2.8)

Parameters	Type/Value	Description/Notes
layoutTemplate	ID	If specified, associates the layout template with this call leg. (From version 2.8)
audioGainMode	disabled|**agc**	Specifies whether this call leg audio should use automatic gain control or not. (From version 2.8) Note: - AGC will be applied to any endpoint (physical endpoints or soft clients) connected directly to the Meeting Server. It will not be applied to TIP calls or AVMCU (because this is a mixed audio stream). - Skype participants connected to AVMCU will not be subject to any AGC as the AVMCU controls the audio - AGC is not applied to distribution links between Meeting Servers because this is a mixed audio stream.
meetingTitlePosition	disabled| top | middle | bottom	Enables and places the meeting title at the specified position. If unspecified, it takes the value bottom. The value disabled removes the meeting title.
chatContributionAllowed	**true** | false	Determines whether or not the participant can send chat messages. If unset at all levels of the hierarchy, then it defaults to true.
handStatus	raised | lowered	Specifies whether or not to raise or lower hand for this call leg.
accessMethod	ID	Associates the specified **`accessMethod`** as the access method for the callLeg/participant and overrides any default or primary access method on the coSpace. For details on permissions to change the participant role, refer to Appendix A.

Parameters	Type/Value	Description/Notes
changeRoleAllowed	**true**, **false**, or <**unset**>	Determines whether or not the participant can change the role of another participant in a call when using the web app. When set to **true**, participant can change the role of other participants. When set to **false**, participants cannot change the role of other participants. For details on permissions to change the participant role, refer to Appendix A. The usual rules for the hierarchy of call leg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
noteContributionAllowed	true| **false**	Determines whether or not the participant can publish notes. If unset at all levels of the hierarchy, then it defaults to false.
captionContributionAllowed	true| **false**	Determines whether or not the participant can send captions in a meeting. The usual rules for the hierarchy of callLeg and callLeg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false. **Note:** If **captionsAllowed** is set to **false** at the call level, the participant cannot send closed captions even if **captionContributionAllowed** is set as **true**.
audioPrompt	On/Off	• On – Blast dial audio prompt will be played, and user has to enter DTMF key to join the conference. • Off – Blast dial audio prompt will not be played, and user doesn't have to enter any DTMF key to join the conference. This parameter is only supported on POST methods.
fileUploadAllowed	true | **false**	• true - Indicates that the participant can share files in the call. • false - Indicates that the participant cannot share files in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

Parameters	Type/Value	Description/Notes
surveyOpsAllowed	`true`| **`false`**	True – Indicates that the participant can create/launch/delete/view results of surveys. False – Indicates that the participant cannot create/launch/delete/view results of surveys. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

8.3.3 /callLegs/<call leg id>/generateKeyframe

POST to /callLegs/<call leg id>/generateKeyframe to trigger the generation of a new keyframe in outgoing video streams for the call leg in question. This is a debug facility, and Cisco Support may ask you to use the feature when diagnosing an issue.

8.3.4 Allowing Far End Camera Control (FECC) using the API

To allow FECC on a remote system's camera, from version 2.9:

- PUT to /callLegs/<call leg id>/cameraControl

This object supports the optional request parameters:

Parameter	Type/Value	Description/Notes
pan	left|right	Pans the remote camera left or right.
tilt	up|down	Tilts the remote camera up or down.
zoom	in|out	Zooms the remote camera in or out.
focus	in|out	Focuses the remote camera in or out.

8.3.5 Taking snapshots of participants in a meeting

Snapshots of the participant's video and/ or snapshot of the meeting video can be taken during a meeting. A GET method on **`callLegs/<call leg id>/snapshot`** is introduced to take snapshots. This option is available at participant level. It supports the following two optional parameters:

Parameter	Type/Value	Description
Direction (optional)	String	It accepts RX or TX • RX/incoming - Participant to CMS • TX/outgoing - CMS to participant If not specified, it defaults to RX/incoming.
maxWidth (optional)	Numeric	The maximum resolution of snapshot supported is 720 pixel. This API accepts maxwidth value. Therefore, to achieve a maximum resolution of 720P for the snapshot, this parameter has to be set to 1280. Although it accepts other values too, Meeting Server responds the snapshot with maximum resolution of 720 pixel. **Note:** The maximum resolution of the snapshot image is 720p (1280*720 pixel) If maxwidth value is not specified, Meeting Server responds to the snapshot with default resolution of 96p.

Note: The Snapshot API cannot be accessed using the Meeting Server web admin interface or API explorer.

8.3.6 Retrieving Information on an Individual Call Leg

GET method performed on a "/callLegs/<callLeg ID>" node.

If the call leg ID supplied is valid, a "200 OK" response is received, with XML content:

Response elements	Type/Value	Description/Notes
callLeg id	ID	
name	String	
remoteParty	String	
originalRemoteParty	String	For outbound calls this is the original destination address . For inbound calls this is the remote address first signaled to the Call Bridge. From version 2.3.
localAddress	String	
call	ID	
tenant	ID	
type	sip|acano	

Response elements	Type/Value	Description/Notes
subType	lync|avaya| distributionLink| lyncDistribution|webApp (from 2.9)	
lyncSubType	audioVideo| applicationSharing| instantMessaging	A further specialization of the call leg type if the call leg sub type is " lync" . audioVideo - this is a Lync call leg used for exchange of audio and video between the Call Bridge and Lync applicationSharing - this is a Lync call leg used for application or desktop sharing between Lync and the Call Bridge instantMessaging - this is a Lync call leg used for the exchange of instant messages between Lync and the Call Bridge If present in the reply, these parameters are overrides currently active specifically for this call leg (i.e. not those values in force because of a “higher level” such as the call leg's associated tenant)
direction	incoming| outgoing	
canMove	true| false	Indicates whether the participant owning this call leg can be moved. (From version 2.6)
movedCallLeg	ID	If this call leg was created by moving a participant, the ID indicates the original call leg that the participant was moved from. (From version 2.6)
movedCallLegCallBridge	ID	If this call leg was created by moving a participant, the ID indicates the Call Bridge that homed the original call leg that the participant was moved from. (From version 2.6)

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>configuration</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>ownerId</td><td>ID</td><td></td></tr>
<tr><td>chosenLayout</td><td>one of:
speakerOnly | telepresence| stacked | allEqual | allEqualQuarters | allEqualNinths | allEqualSixteenths| allEqualTwentyFifths | onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN</td><td></td></tr>
<tr><td></td><td></td><td rowspan="15">callLegProfile fields as described above, if present these show the over-rides currently active specifically for this call leg (i.e. not those in force because of a higher level such as the call leg's associated tenant)</td></tr>
<tr><td>needsActivation</td><td></td></tr>
<tr><td>defaultLayout</td><td></td></tr>
<tr><td>participantLabels</td><td></td></tr>
<tr><td>presentationDisplayMode</td><td></td></tr>
<tr><td>presentationContributionAllowed</td><td></td></tr>
<tr><td>presentationViewingAllowed</td><td></td></tr>
<tr><td>endCallAllowed</td><td></td></tr>
<tr><td>muteOthersAllowed</td><td></td></tr>
<tr><td>videoMuteOthersAllowed</td><td></td></tr>
<tr><td>muteSelfAllowed</td><td></td></tr>
<tr><td>videoMuteSelfAllowed</td><td></td></tr>
<tr><td>changeLayoutAllowed</td><td></td></tr>
</table>
</td><td></td></tr>
</table>

Response elements	Type/Value	Description/Notes

Name	Type	Description
joinToneParticipantThreshold		
leaveToneParticipantThreshold		
videoMode		
rxAudioMute		
txAudioMute		
rxVideoMute		
txVideoMute		
sipMediaEncryption		
audioPacketSizeMs		
deactivationMode		
deactivationModeTime		
telepresenceCallsAllowed		
sipPresentationChannelEnabled		
bfcpMode		
disconnectOthersAllowed		
addParticipantAllowed (from version 2.3)		
qualityMain (from version 2.2)		**Note**: For calls between the Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to the Meeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences. **Note**: These parameters do not apply to incoming video from Cisco Meeting Appor Cisco Meeting App WebRTC app.
qualityPresentation (from version 2.2)		
participantCounter (from version 2.2)		

Response elements	Type/Value	Description/Notes

Name	Type	Description
nameLabelOverride (from version 2.4)		
controlRemoteCameraAllowed (from version 2.8)		
layoutTemplate (from version 2.8)		
audioGainMode (from version 2.8)		
meetingTitlePosition	top | middle | bottom	Enables and places the meeting title at the specified position. If unspecified, it takes the value bottom. The value disabled removes the meeting title.
handStatus	raised | lowered	If set, indicates whether the hand is raised or lowered for this participant or call leg. The value is not returned of the handStatus was not changed during the call.
accessMethod	ID | " coSpace"	Returns the access method used to join the call on dial in or as set on dial out. Where an accessMethod is not specified on the POST operation, the GET still returns an access method ID if defaultAccessMethod was configured on the coSpace or if the primary access method was employed. The API will return " coSpace" if the coSpace was not joined through an access method. This could be on dial in or on dial out if no accessMethod or defaultAccessMethod is specified and no primary access method exists.
handStatusLastModified	String	Returns a UTC date-time for the last time the hand status was modified. The value is not returned if the handStatus was not changed during the call.

Response elements	Type/Value	Description/Notes

Name	Type	Description
chatContributionAllowed	**true** | false	Determines whether or not the participant can send chat messages. If unset at all levels of the hierarchy, then it defaults to true.
changeRoleAllowed	**`true`**, **`false`**, or **`<unset>`**	Determines whether or not the participant can change the role of another participant in a call when using the web app. When set to **true**, participant can change the role of other participants. When set to **false**, participants cannot change the role of other participants. The usual rules for the hierarchy of call leg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
noteContributionAllowed	true| **false**	Determines whether or not the participant can publish notes. If unset at all levels of the hierarchy, then it defaults to false.
captionContributionAllowed	true| **false**	Determines whether or not the participant can send captions in a meeting. The usual rules for the hierarchy of callLeg and callLeg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false. **Note:** If **`captionsAllowed`** is set to **false** at the call level, the participant cannot send closed captions even if **`captionContributionAllowed`** is set as **true**.

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>fileUploadAllowed</td><td>true | false</td><td>• true – Indicates that the participant can share files in the call.
• false – Indicates that the participant cannot share files in the call.

The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.</td></tr>
<tr><td>surveyOpsAllowed</td><td><code>true</code>| <code>false</code></td><td>True – Indicates that the participant can create/launch/delete/view results of surveys.

False – Indicates that the participant cannot create/launch/delete/view results of surveys.

The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>status</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>state</td><td>initial|
ringing|
connected|
onHold</td><td></td></tr>
<tr><td>durationSeconds</td><td>Number</td><td></td></tr>
<tr><td>sipCallID</td><td>String</td><td></td></tr>
<tr><td>groupId</td><td>ID</td><td></td></tr>
<tr><td>recording</td><td>true only</td><td>Only present if the callLeg is recording</td></tr>
<tr><td>streaming</td><td>true only</td><td>Only present if the callLeg is streaming</td></tr>
<tr><td>deactivated</td><td>true only</td><td>Only present if it is true</td></tr>
<tr><td>encryptedMedia</td><td>true| false</td><td></td></tr>
<tr><td>unencryptedMedia</td><td>true| false</td><td></td></tr>
<tr><td>cipherSuite</td><td>one of:
AEAD_AES_256_
GCM AEAD_AES_
128_GCM AES_CM_
128_HMAC_SHA1_
80
AES_CM_128_
HMAC_SHA1_32</td><td>If any of this call leg's media is encrypted, this gives information on the SRTP encryption cipher suite in use:
- AEAD_AES_256_GCM - AES encryption, 256 bit, GCM
- AEAD_AES_128_GCM - AES encryption, 128 bit, GCM
- AES_CM_128_HMAC_SHA1_80 - AES encryption, 128 bit, 80 bit SHA1 authentication tag
- AES_CM_128_HMAC_SHA1_32 - AES encryption, 128 bit, 32 bit SHA1 authentication tag
(from version 2.9)</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>layout</td><td>one of:
speakerOnly | telepresence| stacked | allEqual | allEqualQuarters | allEqualNinths | allEqualSixteenths| allEqualTwentyFifths | onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN</td><td></td></tr>
<tr><td>activeLayout</td><td>one of:
speakerOnly | telepresence| stacked | allEqual | allEqualQuarters | allEqualNinths | allEqualSixteenths| allEqualTwentyFifths | onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN</td><td></td></tr>
<tr><td>availableVideoStreams</td><td>Number</td><td></td></tr>
<tr><td>layoutTemplate</td><td>ID</td><td>Present if a custom layout template is currently being used to generate the layout for this call leg, and, if so, identifies which layout template is being used. (from version 2.8)</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>cameraControlAvailable</td><td>true|false</td><td>Whether this call leg has advertised the ability for its camera to be controlled remotely.
true - camera control for this call leg is possible
false - camera control for this call leg is not possible (from version 2.8)</td></tr>
<tr><td>confirmationStatus</td><td>required | notRequired | confirmed</td><td>required: confirmation=true was configured and the user has not yet provided the DTMF confirmation to join the call
notRequired: confirmation=true was not set
confirmed: a DTMF sequence was entered to confirm that the participant wants to join the call</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>status</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>rxAudio</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>codec</td><td>one of:
g711u
g711a
g722
g728
g729
g722_1
g722_1c
aac
speexNb
speexWb
speexUwb
isacWb
opus</td><td>the audio codec used:
g711u - G.711 mu law
g711a - G.711 a law
g722 - G.722
g728 - G.728
g729 - G.729
g722_1 - G.722.1
g722_1c - G.722.1C (G.722.1 Annex C)
aac - AAC
speexNb - Speex narrowband
speexWb - Speex wideband
speexUwb - Speex ultra-wideband
isacWb - iSAC (internet Speech Audio Codec) wideband
isacSwb - iSAC (internet Speech Audio Codec) superwideband</td></tr>
<tr><td>jitter</td><td>Number</td><td></td></tr>
<tr><td>bitRate</td><td>Number</td><td>The actual measured bit rate of the incoming audio data</td></tr>
<tr><td>codecBitRate</td><td>Number</td><td>Present for audio codec types with variants that can only be distinguished via bit rate (for example, G.722.1), wherein this field will be the expected audio bit rate rather than the observed, measured value (from version 2.1 onwards)</td></tr>
<tr><td>packetLossPercentage</td><td>Number</td><td></td></tr>
<tr><td>gainApplied</td><td></td><td>Value, in decibels, for any gain that has been applied to the audio received from the remote party using AGC (from version 2.8)</td></tr>
</table>
</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>status</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>txAudio</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>codec</td><td>one of:
g711u
g711a
g722
g728
g729
g722_1
g722_1c
aac
speexNb
speexWb
speexUwb
isacWb
opus</td><td>the audio codec used:
g711u - G.711 mu law
g711a - G.711 a law
g722 - G.722
g728 - G.728
g729 - G.729
g722_1 - G.722.1
g722_1c - G.722.1C (G.722.1 Annex C)
aac - AAC
speexNb - Speex narrowband
speexWb - Speex wideband
speexUwb - Speex ultra-wideband
isacWb - iSAC (internet Speech Audio Codec) wideband
isacSwb - iSAC (internet Speech Audio Codec) superwideband</td></tr>
<tr><td>jitter</td><td>Number</td><td></td></tr>
<tr><td>bitRate</td><td>Number</td><td>The actual measured bit rate of the incoming audio data</td></tr>
<tr><td>roundTripTime</td><td>Number</td><td></td></tr>
<tr><td>packetLossPercentage</td><td>Number</td><td></td></tr>
</table>
</td><td></td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>status</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>rxVideo</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td colspan="3">Parameters included in parent tag</td></tr>
<tr><td>role</td><td>main | present-ation</td><td>The type of video stream: main or presentation</td></tr>
<tr><td colspan="3">Response values</td></tr>
<tr><td>codec</td><td>one of:
h261
h263
h263+
h264
h264Lync
vp8
rtVideo</td><td>the video codec used
h261 - H.261
h263 - H.263
h263+ - H.263+
h264 - H.264
h264Lync - H.264 SVC for Lync
vp8 - VP8
rtVideo - RTVideo</td></tr>
<tr><td>width</td><td>Number</td><td></td></tr>
<tr><td>height</td><td>Number</td><td></td></tr>
<tr><td>frameRate</td><td>Number</td><td></td></tr>
<tr><td>jitter</td><td>Number</td><td></td></tr>
<tr><td>bitRate</td><td>Number</td><td></td></tr>
<tr><td>packetLossPercentage</td><td>Number</td><td></td></tr>
</table>
</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>txVideo</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td colspan="3">Parameters included in parent tag</td></tr>
<tr><td>role</td><td>main | presentation</td><td>The type of video stream: main or presentation</td></tr>
<tr><td colspan="3">Response values</td></tr>
<tr><td>codec</td><td>one of:
h261
h263
h263+
h264
h264Lync
vp8
rtVideo</td><td>the video codec used
h261 - H.261
h263 - H.263
h263+ - H.263+
h264 - H.264
h264Lync - H.264 SVC for Lync
vp8 - VP8
rtVideo - RTVideo</td></tr>
<tr><td>width</td><td>Number</td><td></td></tr>
<tr><td>height</td><td>Number</td><td></td></tr>
<tr><td>frameRate</td><td>Number</td><td></td></tr>
<tr><td>jitter</td><td>Number</td><td></td></tr>
<tr><td>bitRate</td><td>Number</td><td></td></tr>
<tr><td>roundTripTime</td><td>Number</td><td></td></tr>
<tr><td>packetLossPercentage</td><td>Number</td><td></td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>activeControl
(from version 2.1)</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>encrypted</td><td>true|false</td><td>If Active Control has been negotiated with the remote party, this indicates whether the Active Control connection is encrypted.

true – an encrypted Active Control connection has been negotiated with the remote party

false – an Active Control connection has been negotiated with the remote party but it is not encrypted</td></tr>
<tr><td>localSubscriptions
(from version 2.2)</td><td colspan="2">see table below.</td></tr>
<tr><td>remoteSubscriptions
(from version 2.2)</td><td colspan="2">see table below.</td></tr>
</table>
</td><td></td></tr>
<tr><td>multiStreamVideo
(from version 2.2)</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>numScreens</td><td>Number</td><td>Indicates that multistream video is active for this call leg</td></tr>
<tr><td>numCameras</td><td>Number</td><td>The number of multistream main video camera streams currently active for this call leg. (from version 2.9)</td></tr>
<tr><td>numCamerasAvailable</td><td>Number</td><td>The number of multistream main video camera streams advertised by the far end as being available for this call leg. (from version 2.9)</td></tr>
</table>
</td><td></td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>lyncRole</td><td>presenter | attendee</td><td>Only present if the participant associated with this call leg is in a Lync conference.
presenter – The participant associated with this call leg is a presenter in the Lync conference.
attendee – The participant associated with this call leg is an attendee in the Lync conference.
(From version 2.4)</td></tr>
</table>
</td><td></td></tr>
</table>

Status of Active Control parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>localSubscriptions
(from version 2.2)</td><td colspan="2"><table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>capabilities</td><td>If present, this indicates that the local Meeting Server has subscribed to the far end's XCCP capabilities.</td></tr>
<tr><td>conferenceInfo</td><td>If present, this indicates that the local Meeting Server has subscribed to the far end's XCCP conference information (this includes the participant list and some conference-wide information such as whether recording is active).</td></tr>
<tr><td>layouts</td><td>If present, this indicates that the local Meeting Server has subscribed to the far end's XCCP layout information.</td></tr>
<tr><td>selfInfo</td><td>If present, this indicates that the local Meeting Server has subscribed to the far end's XCCP self information.</td></tr>
<tr><td>speakerInfo</td><td>If present, this indicates that the local Meeting Server has subscribed to the far end's XCCP speaker information.</td></tr>
</table></td></tr>
<tr><td>remoteSubscriptions
(from version 2.2)</td><td colspan="2"><table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>capabilities</td><td>If present, this indicates that the far end Meeting Server has subscribed to the local XCCP capabilities.</td></tr>
<tr><td>conferenceInfo</td><td>If present, this indicates that the far end Meeting Server has subscribed to the local XCCP conference information (this includes the participant list and some conference-wide information such as whether recording is active).</td></tr>
<tr><td>layouts</td><td>If present, this indicates that the far end Meeting Server has subscribed to the local XCCP layout information.</td></tr>
<tr><td>selfInfo</td><td>If present, this indicates that the far end Meeting Server has subscribed to the local XCCP self information.</td></tr>
<tr><td>speakerInfo</td><td>If present, this indicates that the far end Meeting Server has subscribed to the local XCCP speaker information.</td></tr>
</table></td></tr>
</table>

Note: See Appendix B for an example of the call leg configuration and status returned.

8.4 Call Leg Profile Methods

8.4.1 General information

A call leg profile defines a set of in-call behaviors. coSpace, coSpaceUser, accessMethod, and tenant objects can optionally have a callLegProfile association – if so, call legs that correspond to those objects inherit the in-call behavior defined by the call leg profile. For more information see also Section 16.

8.4.2 Retrieving call leg profiles

GET method on the " /callLegProfiles/" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" on the notional list (see Section 4.2.2).
usageFilter	unreferenced| referenced	Using unreferenced retrieves only those call leg profiles that are not referenced by global settings or any other object. This is a useful check before deleting a call Leg profile.

Response elements	Type/Value	Description/Notes
		Response is a collection of <callLegProfile id=<call leg profile id> objects contained within an " <callLegProfiles>" object <callLegProfile> elements follow the general form on the left.
needsActivation	true|**false**	If set to " true" , the participant is unable to receive or contribute audio and video until one or more “full/activator” participants join.

Response elements	Type/Value	Description/Notes
defaultLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	The default layout to be used for call legs using this call leg profile.
changeLayoutAllowed	true|false	If set to “true” all legs using this call leg profile are allowed to change their screen layout on a SIP endpoint.
participantLabels	true|false	If set to " true" , call legs using this call leg profile will have participant pane labels shown on their video panes.
presentationDisplayMode	dualStream|singleStream	singleStream provides a single composited content+video BFCP stream rather than outgoing content being in a separate stream
presentationContributionAllowed	true|false	If true, the call legs using this call leg profile are allowed to share content
presentationViewingAllowed	true|false	If true, the call legs using this call leg profile are allowed to view shared content
endCallAllowed	true|false	If true, the call legs using this call leg profile are allowed to end calls in which they are participating
disconnectOthersAllowed	true|false	If true, the call legs using this call leg profile are allowed to disconnect other participants, typically via Active Control.

Response elements	Type/Value	Description/Notes
addParticipantAllowed	true|false	If true, the call legs using this call leg profile are allowed to add other participants, typically via Active Control
muteOthersAllowed	true|**false**	If true, the call legs using this call leg profile are allowed to mute or unmute the audio of other par-ticipants
videoMuteOthersAllowed	true|**false**	If true, the call legs using this call leg profile are allowed to mute or unmute (start/stop) the video of other participants
muteSelfAllowed	**true**|false	If true, the call legs using this call leg profile are allowed to mute or unmute their own audio.
videoMuteSelfAllowed	**true**|false	If true, the call legs using this call leg profile are allowed to mute or unmute (start/stop) their own video
joinToneParticipantThreshold	Number	Number of participants using SIP endpoints that will have a " join tone" played; maximum 100, a value of 0 "disables" the feature. web app users do not receive these audio indications because the Roster List provides a visual indication of who is joining and leaving.
leaveToneParticipantThreshold	Number	Number of participants up to which a " leave tone" will be played out (a value of 0 "disables" the feature)

Response elements	Type/Value	Description/Notes
videoMode	auto|disabled	If disabled is set then call legs using this call leg profile will be audio-only, or audio and content – depending on the values for presentationViewingAllowed and txAudioMute. No main stream video will be shown. For devices which show content in the main video stream, content but no participant video will be shown in the main video stream when appropriate.
rxAudioMute	true|false	If true, other participants will not hear audio from call legs using this call leg profile.
txAudioMute	true|false	If true, audio to call legs using this call leg profile will be muted.
rxVideoMute	true|false	If true, contributing (" camera") video from call legs using this call leg profile will not be seen by other participants.
txVideoMute	true|false	If true, video streams to call legs using this call leg profile will be muted (a SIP endpoint's screen just shows the logo, for example, and a web app will be sent no video at all)
sipMediaEncryption	optional|required|prohibited	Same as Web Admin Interface set-ting
audioPacketSizeMs	Number	Numeric value for preferred packet size for outgoing audio streams (in milliseconds, the default value is 20ms)
deactivationMode	deactivate|disconnect| remainActivated	Action for " needsActivation" call legs when the last " activator" leaves
deactivationModeTime	Number	Number of seconds after the last " activator" leaves before which the deactivationMode action is taken

Response elements	Type/Value	Description/Notes
telepresenceCallsAllowed	true|false	If true, a call leg using this call leg profile is allowed to make TIP (Telepresence Interoperability Protocol) calls
sipPresentationChannelEnabled	true|false	If true, a call leg using this call leg profile is permitted to perform presentation video channel operations
bfcpMode	serverOnly| serverAndClient	If presentation video channel operations are enabled for SIP calls, this setting determines the Call Bridge's BFCP behaviour ■ serverOnly – this is the normal setting for a conferencing device, and is intended for use with BFCP client mode devices (for instance, SIP endpoints) ■ serverAndClient – this setting allows the Call Bridge to operate in either BFCP client or BFCP server mode in calls with remote devices. This can allow improved presentation video sharing with a remote conference-hosting device such as a third party MCU
callLockAllowed	true|false	Determines whether or not call legs using this call leg profile are allowed to lock the call.
setImportanceAllowed	true|false	Whether or not call legs using this call leg profile are allowed to change the importance of participants in the call. (From version 2.3)
allowAllMuteSelfAllowed	false|true	Whether or not call legs using this call leg profile are allowed to change the permission of all call legs to mute and unmute themselves.

Response elements	Type/Value	Description/Notes
allowAllPresentationContributionAllowed	false|true	Whether call legs using this call leg profile are allowed to change the permission to present of all call legs.
changeJoinAudioMuteOverrideAllowed	false|true	Whether call legs using this call leg profile are allowed to set the initial mute state of new participants.
recordingControlAllowed	true|false	Whether call legs using this call leg profile are allowed to start/stop recording the call.
streamingControlAllowed	true|false	Whether call legs using this call leg profile are allowed to start/stop streaming the call.
name	String	Name of profile. This parameter is present from version 2.0 onwards.
maxCallDurationTime	Number	The maximum amount of time in seconds that the call leg will exist. This parameter is present from version 2.0 onwards.

Response elements	Type/Value	Description/Notes
qualityMain	**unrestricted**| max1080p30| max720p30| max480p30| max1080p60	Restricts the maximum negotiated main video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and framerate. Note that call legs may operate at lower resolutions or framerates due to endpoint limitations or overall Call Bridge load. (From version 2.2) unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or frame rate max1080p30 – restricts the bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources, for example 1280x720 screen size at 60 frames per second max720p30 – restricts the bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources max480p30 – restricts the bridge to negotiating at most 868x480 screen size at 30 frames per second or equivalent transcoding resources. max1080p60 – restricts the bridge to negotiating at most 1920x1080 screen size at 60 frames per second or equivalent transcoding resources. **Note:** This parameter does not apply to incoming video from Cisco Meeting Server web app or Cisco Meeting App WebRTC app. **Note:** For calls between the

Response elements	Type/Value	Description/Notes
		Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to the Meeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences.

Response elements	Type/Value	Description/Notes
qualityPresentation	**unrestricted**| max1080p30| max720p5| max720p30	Restrict the maximum negotiated presentation video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and frame rate. This only affects callLegs which use a separate presentation stream. (From version 2.2) unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or framerate max1080p30 – restricts the Call Bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources max720p5 – restricts the Call Bridge to negotiating at most 1280x720 screen size at 5 frames per second or equivalent transcoding resources . max720p30 – Restricts the Call Bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources. **Note:** This parameter does not apply to incoming video from Cisco Meeting Server web app or Cisco Meeting App WebRTC app. **Note:** if qualityPresentation is set, when a SIP endpoint shares a presentation with a Lync user, the Meeting Server applies the max resolution frame rate to the incoming presentation video, and the outgoing presentation video to Lync matches this max frame rate. However, when Lync client shares a presentation, the Meeting Server

Response elements	Type/Value	Description/Notes
		does not apply the qualityPresentation setting to incoming presentation video from Lync, but does apply it to the outgoing presentation video to the SIP endpoint.
participantCounter	never| auto| always	Controls the behavior of the onscreen participant counter. (From version 2.2) never – never show an onscreen participant count value auto – show the onscreen participant count value when appropriate. Typically this will be to indicate that there are additional participants present that you cannot currently see. always – always show the onscreen participant count value

8.4.3 Creating and modifying a call leg profile

- Creating: POST method to the "/callLegProfiles" node. If the call leg profile is created successfully, a "200 OK" response will be received, and the "Location" header in the response will contain the ID of the new call leg profile
- Modifying a call leg profile is a PUT method on a " /callLegProfiles/<call leg profile id>" node

Parameters	Type/Value	Description/Notes
needsActivation	true|**false**	If set to " true" , the participant is unable to receive or contribute audio and video until one or more "full/activator" participants join.

Parameters	Type/Value	Description/Notes
defaultLayout	allEqual| speakerOnly| telepresence| stacked| allEqualQuarters| allEqualNinths| allEqualSixteenths| allEqualTwentyFifths| onePlusFive| onePlusSeven| onePlusNine| automatic| onePlusN	The default layout to be used for call legs using this call leg profile.
changeLayoutAllowed	true|false	If set to “true” all legs using this call leg profile are allowed to change their screen layout on a SIP endpoint and web app.
participantLabels	true|false	If set to " true" , call legs using this call leg profile will have participant pane labels shown on their video panes.
presentationDisplayMode	dualStream|singleStream	singleStream provides a single composited content+video BFCP stream rather than outgoing content being in a separate stream
presentationContributionAllowed	true|false	If true, the participant using the call leg can contribute content
presentationViewingAllowed	true|false	If true, the participant using the call leg can view content contributed by others
endCallAllowed	true|false	If true, the participant using the call leg profile can end the meeting for everyone
disconnectOthersAllowed	true|false	If true, call legs using this call leg profile are allowed to disconnect other participants, only via Active Control. (From version 2.1)
addParticipant allowed	true|false	If true, call legs using this call leg profile are allowed to add other participants, typically via Active Control. (From version 2.3)

Parameters	Type/Value	Description/Notes
muteOthersAllowed	true|**false**	If true, the participant using the call leg profile can mute other participants
videoMuteOthersAllowed	**true**|false	Whether call legs using this call leg profile are allowed to mute or unmute (start/stop) the video of other participants
muteSelfAllowed	true|**false**	Whether call legs using this call leg profile are allowed to mute or unmute their own audio
videoMuteSelfAllowed	**true**|false	Whether call legs using this call leg profile are allowed to mute or unmute (start/stop) their own video
joinToneParticipantThreshold	Number	Number of participants using SIP endpoints that will have a " join tone" played; maximum 100, a value of 0 "disables" the feature. web app users do not receive these audio indications because the Roster List provides a visual indication of who is joining and leaving.
leaveToneParticipantThreshold	Number	Number of participants up to which a " leave tone" will be played out (a value of 0 "disables" the feature)
videoMode	auto|disabled	If disabled is set then call legs using this call leg profile will be audio-only, or audio and content - depending on the values for presentationViewingAllowed and txAudioMute. No main stream video will be shown. For devices which show content in the main video stream, content but no participant video will be shown in the main video stream when appropriate.
rxAudioMute	true|false	If true, other participants will not hear audio from call legs using this call leg profile.
txAudioMute	true|false	If true, audio to call legs using this call leg profile will be muted.

Parameters	Type/Value	Description/Notes
rxVideoMute	true|false	If true, contributing (" camera") video from call legs using this call leg profile will not be seen by other participants.
txVideoMute	true|false	If true, video streams to call legs using this call leg profile will be muted (a SIP endpoint's screen just shows the logo, for example, and n web app will be sent no video at all)
sipMediaEncryption	optional|required|prohibited	Same as Web Admin Interface setting
audioPacketSizeMs	Number	Numeric value for preferred packet size for outgoing audio streams (in mil-liseconds, the default value is 20ms)
deactivationMode	deactivate|disconnect| remainActivated	Action for " needsActivation" call legs when the last " activator" leaves
deactivationModeTime	Number	Number of seconds after the last " activator" leaves before which the deactivationMode action is taken
telepresenceCallsAllowed	true|false	If true, a call leg using this call leg profile is allowed to make TIP (Telepresence Interoperability Protocol) calls
sipPresentationChannelEnabled	true|false	If true, a call leg using this call leg profile is permitted to perform presentation video channel operations
bfcpMode	serverOnly|serverAndClient	If presentation video channel operations are enabled for SIP calls, this setting determines the Call Bridge's BFCP behaviour ■ serverOnly - this is the normal setting for a conferencing device, and is intended for use with BFCP client mode devices (for instance, SIP endpoints) ■ serverAndClient - this setting allows the Call Bridge to operate in either BFCP client or BFCP server mode in calls with remote devices. This can allow improved presentation video sharing with a remote conference-hosting device such as a third party MCU

Parameters	Type/Value	Description/Notes
callLockAllowed	true|false	Determines whether or not call legs using this call leg profile are allowed to lock the call.
setImportanceAllowed	true|**false**	Determines whether or not call legs using this call leg profile are allowed to change the importance of participants in the call. (From version 2.3)
allowAllMuteSelfAllowed	true|false	Whether or not call legs using this call leg profile are allowed to change the permission of all call legs to mute and unmute themselves.
allowAllPresentationContributionAllowed	false|true	Whether call legs using this call leg profile are allowed to change the permission to present of all call legs.
changeJoinAudioMuteOverrideAllowed	false|true	Whether call legs using this call leg profile are allowed to set the initial mute state of new participants.
recordingControlAllowed	true|false	If true, call legs using this call leg profile are allowed to start/stop recording the call.
streamingControlAllowed	true|false	If true, call legs using this call leg profile are allowed to start/stop streaming the call. (From version 2.1)
name	String	Name of profile. This parameter is present from version 2.0 onwards.
maxCallDurationTime	Number	The maximum amount of time in seconds that the call leg will exist. This parameter is present from version 2.0 onwards.

Parameters	Type/Value	Description/Notes
qualityMain	**unrestricted**| max1080p30| max720p30| max480p30| max1080p60	Restricts the maximum negotiated main video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and framerate. Note that call legs may operate at lower resolutions or framerates due to endpoint limitations or overall Call Bridge load. (From version 2.2) unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or frame rate max1080p30 – restricts the bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources, for example 1280x720 screen size at 60 frames per second max720p30 – restricts the bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources max480p30 – restricts the bridge to negotiating at most 868x480 screen size at 30 frames per second or equivalent transcoding resources. max1080p60 – restricts the bridge to negotiating at most 1920x1080 screen size at 60 frames per second or equivalent transcoding resources. **Note:** This parameter does not apply to incoming video from Cisco Meeting Server web app or Cisco Meeting App WebRTC app. **Note:** For calls between the Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to the Meeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences.

Parameters	Type/Value	Description/Notes
qualityPresentation	**unrestricted**| max1080p30| max720p5| max720p30	Restrict the maximum negotiated presentation video call quality for this call leg based on limiting transcoding resources. Specified using a typical resolution and frame rate. This only affects callLegs which use a separate presentation stream. (From version 2.2) unrestricted – this is the default setting if not specified, and matches the behavior of older Call Bridge versions, where no restrictions are placed on resolution or framerate max1080p30 – restricts the Call Bridge to negotiating at most 1920x1080 screen size at 30 frames per second or equivalent transcoding resources max720p5 – restricts the Call Bridge to negotiating at most 1280x720 screen size at 5 frames per second or equivalent transcoding resources . max720p30 – Restricts the Call Bridge to negotiating at most 1280x720 screen size at 30 frames per second or equivalent transcoding resources. **Note:** This parameter does not apply to incoming video from Cisco Meeting Server web app or Cisco Meeting App WebRTC app. **Note:** if qualityPresentation is set, when a SIP endpoint shares a presentation with a Lync user, the Meeting Server applies the max resolution frame rate to the incoming presentation video, and the outgoing presentation video to Lync matches this max frame rate. However, when Lync client shares a presentation, the Meeting Server does not apply the qualityPresentation setting to incoming presentation video from Lync, but does apply it to the outgoing presentation video to the SIP endpoint.

Parameters	Type/Value	Description/Notes
participantCounter	never| auto| always	Controls the behavior of the onscreen participant counter. (From version 2.2) never – never show an onscreen participant count value auto – show the onscreen participant count value when appropriate. Typically this will be to indicate that there are additional participants present that you cannot currently see. always – always show the onscreen participant count value
controlRemoteCameraAllowed	**true**|false	Determines if call legs using this call leg profile are allowed to control a remote participant's camera (via FECC). (From version 2.8)
layoutTemplate	ID	If specified, associates the layout template with this call leg profile. (From version 2.8)
audioGainMode	disabled|**agc**	Specifies whether or not audio for call legs using this call leg profile should use automatic gain control (From version 2.8) Note: - AGC will be applied to any endpoint (physical endpoints or soft clients) connected directly to the Meeting Server. It will not be applied to TIP calls or AVMCU (because this is a mixed audio stream). - Skype participants connected to AVMCU will not be subject to any AGC as the AVMCU controls the audio - AGC is not applied to distribution links between Meeting Servers because this is a mixed audio stream.
meetingTitlePosition	disabled| top | middle | **bottom**	Enables and places the meeting title at the specified position. If unspecified, it takes the value bottom. The value disabled removes the meeting title.

Parameters	Type/Value	Description/Notes
chatContributionAllowed	**true**, false	Determines whether or not the participant can send chat messages. If unset at all levels of the hierarchy, then it defaults to true. .
changeRoleAllowed	**`true`**, **`false`**, or **`<unset>`**	Determines whether or not the participant can change the role of another participant in a call when using the web app. When set to **true**, participant can change the role of other participants. When set to **false**, participants cannot change the role of other participants. For details on permissions to change the participant role, refer to Appendix A. The usual rules for the hierarchy of call leg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
noteContributionAllowed	true| **false**	Determines whether or not the participant can publish notes. If unset at all levels of the hierarchy, then it defaults to false.
captionContributionAllowed	true| **false**	Determines whether or not the participant can send captions in a meeting. The usual rules for the hierarchy of callLeg and callLeg profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false. **Note:** If **`captionsAllowed`** is set to **false** at the call level, the participant cannot send closed captions even if **`captionContributionAllowed`** is set as **true**.

Parameters	Type/Value	Description/Notes
fileUploadAllowed	true | **false**	• true - Indicates that the participant can share files in the call. • false - Indicates that the participant cannot share files in the call. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.
surveyOpsAllowed	`true`| **`false`**	True – Indicates that the participant can create/launch/delete/view results of surveys. False - Indicates that the participant cannot create/launch/delete/view results of surveys. The usual rules for the hierarchy of calls and call profiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to false.

In all cases, if you explicitly setting a parameter to an empty value in the POST or PUT, that parameter is " unset" for that profile. Those call legs then " inherit" the value for that parameter from the level above's call leg profile.

8.4.4 Retrieving information on an individual call leg profile

GET method on a /callLegProfiles/<call leg profile id> node. If the call leg profile ID supplied is valid, a " 200 OK" response and a single " <callLegProfile id=call leg profile<ID>>" object will be returned with data as per the previous section.

8.4.5 Example call leg profile & access method use

The main use for being able to associate call leg profiles with access methods is to be able to construct separate URI / call ID / passcode combinations giving different in-call behaviors. For example, one call leg profile whose “needsActivation” value is “true” could be associated with one access method, and another call leg profile whose “needsActivation” value is “false” could be associated with a different access method.

Effectively this sets up separate “activator” and “guest” access methods for that coSpace, with callers to the “needsActivation=true” access method needing to wait until a successful call in to the other access method before their conference audio and video become active. For the multiple access methods linked to different call leg profiles in this way, you can choose to

distinguish between them only by passcode; essentially, activator and guest users dial the same URI but enter a different PIN depending on whether they're an activator or guest participant.

8.4.6 /callLegProfiles/<call leg profile id>/usage object method

There is a /callLegs/<call leg id>/usage object in the hierarchy. Performing a GET on this object retrieves, for the queried call leg, a list of where the specified call leg profile is used: whether it is set to be the global call leg profile or any associations it has with tenants, coSpaces, coSpace users, coSpace access methods.

8.4.7 /callLegs/<call leg id>/callLegProfileTrace object method

There is a /callLegs/<call leg id>/callLegProfileTrace object in the hierarchy. Performing a GET on this object retrieves, for the call leg that you queried, how its in-force call leg profile has been arrived at; that is, the hierarchy of overrides that have contributed to the currently " in force" call leg profile. Specifically, the response includes a section for each level in the profile hierarchy, and details which call leg profile elements have been applied at which level.

The end result for each parameter is the lowest level's override for that parameter; for example, if the tenant-level call leg profile has set "participantLabels" to true but the coSpace call leg profile has it set to false, then call legs in that coSpace will not display participant labels.

GET method on the " /callLegProfileTrace/" node to retrieve call leg profile trace.

Parameters	Type/Value	Description/Notes
scope	one of:	Indicates at which level the set of profile parameters have been applied
	global	the profile parameters given have been applied at the system-wide level; specifically, this means those parameters are present in the top-level callLegProfile, configured under system/profiles
	tenant	the profile parameters given have been applied at the tenant level
	coSpace	the profile parameters given have been applied at the coSpace level
	accessMethod	the profile parameters given have been applied at the access method level
	coSpaceUser	the profile parameters given have been applied via a callLegProfile associated with a coSpaceUser
	callOut	the profile parameters given have been applied via a callLegProfile supplied when the call leg was created
	callLeg	the profile parameters given have been applied specifically to this call leg (i.e. not via any configured callLegProfile object)
id	id	If present, the callLegProfile applicable to the scope of this entry
meetingTitlePosition	top | middle | bottom	Enables and places the meeting title at the specified position. If unspecified, it takes the value bottom.

Response elements	Type/Value	Description/Notes
needsActivation defaultLayout changeLayoutAllowed participantLabels presentationDisplayMode presentationContributionAllowed presentationViewingAllowed endCallAllowed muteOthersAllowed videoMuteOthersAllowed muteSelfAllowed videoMuteSelfAllowed joinToneParticipantThreshold leaveToneParticipantThreshold videoMode rxAudioMute txAudioMute rxVideoMute txVideoMute sipMediaEncryption audioPacketSizeMs deactivationMode deactivationModeTime telepresenceCallsAllowed sipPresentationChannelEnabled bfcpMode callLockAllowed setImportanceAllowed allowAllMuteSelfAllowed allowAllPresentationContributionAllowed changeJoinAudioMuteOverrideAllowed recordingControlAllowed streamingControlAllowed		Parameters that show which call leg profile values have been overridden at this level

Response elements	Type/Value	Description/Notes
name		
maxCallDurationTime		
disconnectOthersAllowed		
audioGainMode (from version 2.8)		
addParticipantAllowed (from version 2.3)		
participantCounter (from version 2.2)		
qualityMain (from version 2.2) qualityPresentation (from version 2.2)		**Note:** The qualityMain and qualityPresentation parameters do not apply to incoming video from Cisco Meeting Server web app or Cisco Meeting App WebRTC app. **Note**: For calls between the Cisco Meeting Server and Lync/Skype for Business, this limit is only applied to incoming media to the Cisco Meeting Server, outgoing media is unaffected. This applies to both Lync user calls and Dual Homed conferences.

8.5 Setting Individual Features for a Call Leg

You can set, modify and retrieve on a per active call leg basis, these settings include whether a presentation should be restricted to single screen mode (i.e. one combined main and presentation video stream), or allowed to use separate video streams if this is supported by the receiving party.

Note: Setting individual parameters for a call leg overrides the values of the call leg profile.

8.6 Layout template Methods

Meeting Server supports customizable layouts. This allows Administrators more flexibility to create and apply custom layouts that suit their specific needs. This feature can work on single and dual screen endpoints and web app.

Note: Customizable layouts are not supported on triple screen endpoints.

The layout template methods allow you to implement customized JSON layout template files.

For more information on customizable layouts, see Cisco Meeting Server Administrator Guide to Screen Layouts, Pane Placement and Customized Layouts.

8.6.1 Adding and Modifying Layout Templates

POST method performed on "/layoutTemplates" node or PUT method to a "/layoutTemplates/<layout template id>" node.

Parameters	Type/Value	Description/Notes
name	string	The human-readable name associated with this layout template with a maximum length of 200 characters.

8.6.2 Retrieving (Enumerate) of Layout Templates

GET method on the /layoutTemplates node.

URI Parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve layout templates other than the first page in the notional list
Limit		
filter	string	supply filter=<string> to return just those layout templates that match the filter

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <layoutTemplates total=" N" > tag with potentially multiple <layoutTemplate> elements within it. Each <layoutTemplate> tag may include the following elements:
name	string	The human-readable name associated with this layout template with a maximum length of 200 characters.
templateSize	numeric	Size (in bytes) of the template

8.6.3 Retrieving Information on Layout Templates

GET method performed on the /layoutTemplates/<layout template id> node.

Response elements	Type/Value	Description/Notes
name	string	The human-readable name associated with this layout template with a maximum length of 200 characters.
templateSize	numeric	Size (in bytes) of the template

8.6.4 Assigning a layout template description to Layout Templates

PUT method to " /layoutTemplates/<layout template id>/template" to assign a layout template description to a layout template; requires " Content-Type" to be set to " application/json"

8.6.5 Retrieving a layout template description on Layout Templates

GET method on " /layoutTemplates/<layout template id>/template" to retrieve the layout template description associated with a layout template.

URI parameter	Type/Value	Description/Notes
source	parsed	This optional URI parameter " source" can be specified on the GET operation. If it's not provided, the GET operation returns the original JSON description as given on the PUT on " /layoutTemplates/<layout template id>/template" If specified, source=parsed returns the layout template data after it has been processed by the Meeting Server; this allows you to see if any changes have been made to the uploaded template data by the Meeting Server, for instance to satisfy one or more validation constraints.

8.7 Call Branding Profile Methods

Call branding profiles control the in-call experience for SIP (including Lync) calls, and the ability to customize text within invitations. For more information see also Section 16.

Note: From version 2.4, the use of callBrandingProfiles no longer requires a branding license.

8.7.1 Retrieving Call Branding Profiles

GET method performed on the “/callBrandingProfiles” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
usageFilter	referenced| unreferenced	Using unreferenced retrieves only those call branding profiles that are not referenced by global settings or any other object. This is a useful check before deleting a call branding profile.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <callBrandingProfiles total=" N" > tag with potentially multiple <callBrandingProfiles> elements within it.

8.7.2 Setting up and modifying call branding profiles

- Creating: POST method to the /callBrandingProfiles" node
- Modifying: PUT to " /callBrandingProfiles/<call branding profile id>"

Parameters	Type/Value	Description/Notes
invitationTemplate	URL	The HTTP or HTTPS URL of the invitation template text that web apps will use when constructing textual invitations. Refer to the Cisco Meeting Server Customization Guide for details on how to customize the text within the invitation.
resourceLocation	URL	The HTTP or HTTPS URL that the Call Bridge call branding files will be retrieved from. This is the " directory" in which the individual audio and graphic files reside. Details of these files are in the Cisco Meeting Server Customization Guide.

8.7.3 Retrieving detailed information about an individual call branding profile

GET method performed on a “/callBrandingProfiles/<call branding profile id>" node. If the call branding profile id ID supplied is valid, a “200 OK” response is received, with XML content matching the section above.

8.8 DTMF Profile Methods

dtmfProfiles can be used to define a number of DTMF sequences that can be used to control audio – as explained in this section. The dtmfProfile does not define the ability to perform the actions, it defines the DTMF string that will invoke action. The definition for who has the authority to invoke that action within the coSpace is defined at the callLegProfile level. For more information see also Section 16.

If you are using the Meeting Server alongside third party solutions, or to replace an existing solution, then set the values to match the values that solution uses e.g. Lync conferencing uses *6 for both mute and unmute, so set toggleMuteSelfAudio to *6.

8.8.1 Retrieving DTMF Profiles

GET method performed on the “/dtmfProfiles” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
usageFilter	referenced| unreferenced	Using unreferenced retrieves only those DTMF profiles that are not referenced by global settings or any other object. This is a useful check before deleting the profile. To retrieve just those DTMF profiles which are referenced in at least one place, supply " usageFilter=referenced" .

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <dtmfProfiles total=" N" > tag with potentially multiple <dtmfProfile> elements within it.
muteSelfAudio	String	DTMF sequence to be used by a participant to mute the audio being contributed to their call
unmuteSelfAudio	String	DTMF sequence to be used by a participant to unmute their audio
toggleMuteSelfAudio	String	DTMF sequence to be used by a participant to toggle between mute and unmute audio from themselves
muteAllExceptSelfAudio	String	DTMF sequence to be used by a participant to mute all other participants in the call.
unmuteAllExceptSelfAudio	String	DTMF sequence to be used by a participant to unmute all other participants in the call.
startRecording	String	DTMF sequence to be used by a participant to start recording the active call.
stopRecording	String	DTMF sequence to be used by a participant to stop recording the active call.
muteAllNewAudio	String	DTMF sequence to be used by a participant to mute all new participants. Sets joinAudioMuteOverride Call object to true.
unmuteAllNewAudio	String	DTMF sequence to be used by a participant to unmute all new participants. . Sets joinAudioMuteOverride Call object to false.
defaultMuteAllNewAudio	String	DTMF sequence to be used by a participant to use the audio mute value from the call leg profile for new participants.

Response elements	Type/Value	Description/Notes
muteAllNewAndAllExceptSelfAudio	String	DTMF sequence to be used by a participant to mute all new participants and all other participants in the call. Sets joinAudioMuteOverride in the call object to `true` and mutes all call legs except for the issuer. This requires 'muteOthersAllowed' and 'changeJoinAudioMuteOverrideAllowed' to be 'true' in the call leg profile of the issuer.
unmuteAllNewAndAllExceptSelfAudio	String	DTMF sequence to be used by a participant to unmute all new participants and all other participants in the call. Sets joinAudioMuteOverride in the call object to `false` and unmutes all call legs except for the issuer. This requires 'muteOthersAllowed' and 'changeJoinAudioMuteOverrideAllowed' to be 'true' in the call leg profile of the issuer.

8.8.2 Setting up and modifying DTMF Profiles

- Creating: POST method to the /dtmfProfiles" node
- Modifying: PUT to " /dtmfProfiles/<dtmfprofile id>"

Parameters	Type/Value	Description/Notes
lockCall	String	DTMF sequence to be used by a participant to lock the call; new call legs that require activation will not be activated even if an activator has joined the call.
unlockCall	String	DTMF sequence to be used by a participant to unlock the call; this will activate all call legs that need activation if there is an activator in the call.
nextLayout	String	DTMF sequence to be used by a participant to change the participant's video layout to the next one in the list.
previousLayout	String	DTMF sequence to be used by a participant to change the participant's video layout to the previous one in the list.
muteSelfAudio	String	DTMF sequence to be used by a participant to mute the audio being contributed to their call
unmuteSelfAudio	String	DTMF sequence to be used by a participant to unmute their audio
toggleMuteSelfAudio	String	DTMF sequence to be used by a participant to toggle between mute and unmute audio from themselves

Parameters	Type/Value	Description/Notes
muteAllExceptSelfAudio	String	DTMF sequence to be used by a participant to mute all other participants in the call.
unmuteAllExceptSelfAudio	String	DTMF sequence to be used by a participant to unmute all other participants in the call.
startRecording	String	DTMF sequence to be used by a participant to start recording the active call.
stopRecording	String	DTMF sequence to be used by a participant to stop recording the active call.
startStreaming	String	DTMF sequence to be used by a participant to start streaming the active call.
stopStreaming	String	DTMF sequence to be used by a participant to stop streaming the active call.
allowAllMuteSelf	String	DTMF sequence to be used by a participant to allow all participants to mute and unmute themselves. Sets allowAllMuteSelf in the Call object to true.
cancelAllowAllMuteSelf	String	DTMF sequence to be used by a participant to cancel the permission to allow all participants to mute and unmute themselves. Sets allowAllMuteSelf in the Call object to false.
allowAllPresentationContribution	String	DTMF sequence to be used by a participant to allow all participants to present.
cancelAllowAllPresentationContribution	String	DTMF sequence to be used by a participant to cancel the permission for all participants to present.
muteAllNewAudio	String	DTMF sequence to be used by a participant to mute all new participants. Sets joinAudioMuteOverride Call object to true.
unmuteAllNewAudio	String	DTMF sequence to be used by a participant to unmute all new participants. . Sets joinAudioMuteOverride Call object to false.
defaultMuteAllNewAudio	String	DTMF sequence to be used by a participant to use the audio mute value from the call leg profile for new participants.
muteAllNewAndAllExceptSelfAudio	String	DTMF sequence to be used by a participant to mute all new participants and all other participants in the call. Sets joinAudioMuteOverride in the call object to `true` and mutes all call legs except for the issuer. This requires 'muteOthersAllowed' to be 'true' in the call leg profile of the issuer.

Parameters	Type/Value	Description/Notes
unmuteAllNewAndAllExceptSelfAudio	String	DTMF sequence to be used by a participant to unmute all new participants and all other participants in the call. Sets joinAudioMuteOverride in the call object to `false` and unmutes all call legs except for the issuer. This requires 'muteOthersAllowed' to be 'true' in the call leg profile of the issuer.
endCall	String	DTMF sequence to be used by a participant to end the call; this will disconnect all participants including the participant who initiated the operation
getTotalParticipantCount	String	DTMF sequence to be used by a participant to retrieve the total number of partipants in the call. (From version 2.8.)

8.8.3 Retrieving detailed information about an individual dtmfProfile

GET method performed on a "/dtmfProfiles/<dtmfprofile id>" node. If the dtmfProfile ID supplied is valid, a "200 OK" response is received, with XML content matching the section above.

8.9 IVR Methods

8.9.1 Retrieving IVRs

GET method performed on the "/ivrs" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those IVRs that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
tenantFilter	ID	Supply tenantFilter to return only those IVRs associated with the specified tenant.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <ivrs total=" N" > tag with potentially multiple <ivr> elements within it. Each <ivr> tag may include the following elements:
uri	URI user part	The URI to be used for this IVR. It can have a maximum length of 200 characters.

8.9.2 Setting up and modifying IVRs

- Creating: POST method to the /ivrs" node
- Modifying: PUT to " /ivrs/<ivr id>"

Parameters	Type/Value	Description/Notes
uri *	URI user part	The URI to be used for this IVR and can have a maximum length of 200 characters.
tenant	ID	If specified, calls to this IVR will only be able to join coSpaces associated with the specified tenant. If no tenant is supplied, calls to this IVR will be able to join any call on the system that has a call ID configured
tenantGroup	ID	Calls to this IVR will only be able to join coSpaces associated with tenants within the specified tenant group. If no tenant group is supplied, calls to this IVR will only be able to join coSpaces without a tenant, or associated with a tenant in no tenant group.
ivrBrandingProfile	ID	If supplied, specifies an IVR branding profile to be used for calls to this IVR - an IVR branding profile supplied here will take precedence over any top-level or tenant-level IVR branding profile
resolveCoSpaceCallIds	true|false	Whether calls to this IVR will accept coSpace and coSpace access method call IDs for the purpose of allowing callers to join Meeting Server coSpaces. If this parameter is not supplied in a create (POST) operation, it defaults to " true" .
resolveLyncConferenceIds	true|false	Whether calls to this IVR will accept IDs to be resolved to Lync scheduled conference IDs. If this parameter is not supplied in a create (POST) operation, it defaults to " false" .

8.9.3 Retrieving detailed information about an individual IVR

GET method performed on a “/ivrs/<ivr id>" node. If the IVR ID supplied is valid, a “200 OK” response is received, with XML content matching the section above.

8.10 IVR Branding Profile Methods

IVR branding profiles can define the experience when dialing into an IVR. For more information see alsoSection 16.

8.10.1 Retrieving IVR branding profiles

GET method performed on the “/ivrBrandingProfiles” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
usageFilter	referenced| unreferenced	Using unreferenced retrieves only those ivr branding profiles that are not referenced by global settings or any other object. This is a useful check before deleting an ivr branding profile.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <ivrBrandingProfiles total=" N" > tag with potentially multiple <ivrBrandingProfile> elements within it.
resourceLocation	URL	The HTTP or HTTPS URL that the IVR branding files will be retrieved from. This should be the " directory" in which the individual audio and graphic files reside. Details of these files are in the Cisco Meeting Server Customization Guide

8.10.2 Setting up and modifying an IVR branding profile

- Creating: POST method to the /ivrBrandingProfiles " node
- Modifying: PUT to " /ivrBrandingProfiles/<ivr branding profile id>"

Parameters	Type/Value	Description/Notes
resourceLocation	URL	The HTTP or HTTPS URL that the IVR branding files will be retrieved from. This should be the " directory" in which the individual audio and graphic files reside. Details of these files are in the Cisco Meeting Server Customization Guide.

8.10.3 Retrieving detailed information about an individual IVR branding profile

GET method performed on a “/ivrBrandingProfiles/<ivr branding profile id>" node. If the IVR branding profile ID supplied is valid, a “200 OK” response is received, with XML content matching the section above.

8.11 Participant Related Methods

“participant” should not be confused with “callLeg” objects. A “participant” can be a user's Lync session in which there might be separate call legs for audio and video, application sharing and IM.

Each Call Bridge involved in a distributed call has an overall picture of the “participant” list for that call, including participants hosted on other Call Bridges. For participants hosted on the Call

Bridge being queried, you can enumerate the constituent call legs but for participants hosted on another Call Bridge, querying those participants yields the ID of the Call Bridge on which they are hosted. (You can then query the " owning" Call Bridge using the same participant ID in order to retrieve call leg-level detail.)

Note: Also see the section on /call/callLegId/participants Section 8.3.6.

8.11.1 Retrieving participants

GET method performed on the “/participants” node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those active participants that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2)
tenantFilter	ID	Supply the tenantFilter to return only those participants belonging to that tenant
callBridgeFilter	ID	Supply the callBridgeFilter to return only those participants located on that Call Bridge

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <participants total=”N”> tag with potentially multiple <participant> elements within it. <participant> elements follow the general form on the left.
name	String	The human-readable display name associated with this participant with a maximum length of 200 characters.
call	ID	The call that this participant is part of
tenant	ID	If present, the id of the tenant that this participant is associated with
callBridge	ID	If present, the remote clustered Call Bridge that this participant is connected to
uri	String	The URI associated with this participant with a maximum length of 200 characters.
originalUri	String	The remote address first used by or signaled to the Call Bridge. (From version 2.3)

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>numCallLegs</td><td>Number</td><td>The current number of active call legs associated with this participant. This value will only be present for those participants local to the Call Bridge to which the request is made</td></tr>
<tr><td>userJid</td><td>String</td><td>The userJid associated with this participant</td></tr>
<tr><td>isActivator</td><td>true|false</td><td>Whether this participant is considered to be an " activator"
■ true - this participant is an activator - it is itself " activated" and that it will cause any currently-connected " de-activated" participants to become activated
■ false - this participant is not an activator - it needs to wait for one or more " activator" participants to be present before it is fully " activated"</td></tr>
<tr><td>canMove</td><td>true|false</td><td>Whether this participant can be moved to a different conference using the movedParticipant parameter on object /calls/<call id>/participants. (From version 2.6)</td></tr>
<tr><td>movedParticipant</td><td>ID</td><td>If this participant was created by moving a participant (by POSTing the movedParticipant parameter to object /calls/<call id>/participants), then the ID indicates the original participant that this participant was moved from.(From version 2.6)</td></tr>
<tr><td>movedParticipantCallBridge</td><td>ID</td><td>If this participant was created by moving a participant, then the ID indicates the Call Bridge that homed the original participant that this participant was moved from. (From version 2.6)</td></tr>
<tr><td>status (from version 2.2)</td><td colspan="2"><table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>state</td><td>initial|ringing|
connected|
onHold</td><td>The call state of this participant.</td></tr></table></td></tr>
<tr><td>configuration
(from version 2.2)</td><td colspan="2"><table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>importance</td><td>Number</td><td>The importance of this participant.</td></tr></table></td></tr>
</table>

8.11.2 Changing settings for a participant already in a conference

- Modifying: PUT to " /participants/<participant id>"

Parameters	Type/Value	Description/Notes
importance	Number	Set the importance of this participant already in a conference. For example to 1. Maximum value is 2,147,483,647. To remove importance leave the importance parameter as unset (leave value as blank). (From version 2.2)
dtmfSequence	String	Set DTMF sequence to get played to this participant. (From version 2.4)
nameLabelOverride	String	Override the name of this participant. (From version 2.4)
deactivated	true | false	• true - Participant will wait in the lobby or can be moved to the lobby. From version 3.4. • false - Participants are allowed into the meeting from the lobby.
handStatus	raised | lowered	Specifies whether or not to raise or lower hand for this participant or call leg. (From version 3.2)

8.11.3 Retrieving detailed information on an individual participant

GET method performed on a "/participants/<participant ID>" node

If the participant ID supplied is valid, a "200 OK" response is received, with XML content of the form:

Response elements	Type/Value	Description/Notes
name	String	The human-readable display name associated with this participant. It can have a maximum length of 200 characters.
call	ID	The call that this participant is part of
tenant	ID	If present, the id of the tenant that this participant is associated with
callBridge	ID	If present, the remote clustered Call Bridge that this participant is connected to
uri	String	The URI associated with this participant with a maximum length of 200 characters.
originalUri	String	The remote address first used by or signalled to the Call Bridge. (From version 2.3)
numCallLegs	Number	The current number of active call legs associated with this participant. This value will only be present for those participants local to the Call Bridge to which the request is made

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>userJid</td><td>String</td><td>The userJid associated with this participant</td></tr>
<tr><td>isActivator</td><td>true|false</td><td>Whether this participant is considered to be an " activator"
■ true - this participant is an activator - it is itself " activated" and that it will cause any currently-connected " de-activated" participants to become activated
■ false - this participant is not an activator - it needs to wait for one or more " activator" participants to be present before it is fully " activated"</td></tr>
<tr><td>canMove</td><td>true|false</td><td>Whether this participant can be moved to a different conference using the movedParticipant parameter on object /calls/<call id>/participants. (From version 2.6)</td></tr>
<tr><td>canMoveToLobby</td><td>true|false</td><td>• true - Participant can be moved to lobby.
• false - Participant cannot be moved to lobby.</td></tr>
<tr><td>movedParticipant</td><td>ID</td><td>If this participant was created by moving a participant (by POSTing the movedParticipant parameter to object /calls/<call id>/participants), then the ID indicates the original participant that this participant was moved from.(From version 2.6)</td></tr>
<tr><td>movedParticipantCallBridge</td><td>ID</td><td>If this participant was created by moving a participant, then the ID indicates the Call Bridge that homed the original participant that this participant was moved from. (From version 2.6)</td></tr>
<tr><td>status (from version 2.2)</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>state</td><td>initial|ringing|connected|onHold</td><td>The call state of this participant.</td></tr>
<tr><td>cameraControlAvailable</td><td>true | false</td><td>Whether this participant has advertised the ability for its camera to be controlled remotely (from version 2.8)
■ true - camera control for this participant is possible
■ false - camera control for this participant is not possible</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>configuration

(from version 2.2)</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>importance</td><td>Number</td><td>The importance of this participant.</td></tr>
<tr><td>nameLabelOverride</td><td>String</td><td>The overridden name of this participant (from version 2.4)</td></tr>
<tr><td>defaultLayout</td><td>allEqual|
speakerOnly|
telepresence|
stacked|
allEqualQuarters|
allEqualNinths|
allEqualSixteenths|
allEqualTwentyFifths|
onePlusFive|
onePlusSeven|
onePlusNine|
automatic|
onePlusN</td><td>If set, the default layout returned is that of this participant as obtained from the resultant call leg profile (from version 2.8)</td></tr>
<tr><td>layoutTemplate</td><td>ID</td><td>If set, the layout template ID is returned that is associated with this participant as obtained from the resultant call leg profile (from version 2.8)</td></tr>
<tr><td>handStatus</td><td>raised | lowered</td><td>If set, indicates whether this participants's hand is raised or lowered.</td></tr>
<tr><td>handStatusLastModified</td><td>String</td><td>Returns a UTC date-time to indicate the last time the hand status was modified.</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>accessMethod</td><td>ID | " coSpace"</td><td>• Returns the access method used to join the call on dial in or as set on dial out.
Where an accessMethod is not specified on the POST operation, the GET still returns an access method ID if defaultAccessMethod was configured on the coSpace or if the primary access method was employed.

• The API will return " coSpace" if the coSpace was not joined through an access method. This could be on dial in or on dial out if no accessMethod or defaultAccessMethod is specified and no primary access method exists.</td></tr>
</table>
</td><td></td></tr>
<tr><td>highestIm-portance</td><td>Numeric</td><td>This will take the highest importance value that can be assigned to a particular participant. When highestIm-portance=n is set, the first n panes are placed with participants who have the importance value assigned to them, ranging from n to 1. Set the minimum value to 0 to disable pane place-ment for particular participant.</td></tr>
</table>

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td>import-anceData</td><td>List of JSON</td><td>List of JSON data where one object contains participant guid and importance.
<table>
<tr><th>Parameters</th><th>Type</th><th>Description</th></tr>
<tr><td>id</td><td>ID</td><td>The user Jid (<participant id>) associated with the participant whose pane needs to be updated.</td></tr>
<tr><td>importance</td><td>Numeric</td><td>The importance value assigned to a specific participant.</td></tr>
</table>
Note: The JSON list can have a maximum of 25 entries, as it is the maximum number of panes that a screen can have.</td></tr>
</table>

8.11.4 Retrieving a participant's call legs

GET method performed on a "/participant/<participant ID>/callLegs" node retrieves the participant's active call legs. If successful, the parameters described above for call legs are returned.

Note that if this call leg is part of a distributed meeting (one hosted by more than one Call Bridge) then these details are only returned for local participants. If the participant's call legs are hosted by another call Bridge the id of that Call Bridge is returned.

8.11.5 Limiting a call's participants

You can set a limit on the number of participants that are permitted to be in a call. You can set:

- A per-tenant participantLimit value, which imposes a limit on the total number of participants that are allowed to be active for that tenant.
- A " participantLimit" value within a 'callProfile' object; this means that calls (e.g. coSpace instantiations) for whom that ''callProfile'' is in force will have the limit enforced

The callProfiles can be attached at the system, tenant or coSpace level, with the most specific taking effect

Therefore a call's participantLimit depends on a number of factors.

If a call's ''participantLimit'' has been reached:

- No new participants can be added to it

Note: The participantLimit value also includes the number of participants on Skype, not just those on the Meeting Server side.

However:

- A participant on a web app can use any combination of chat, use video, audio and show/receive a presentation. These elements comprise one callLeg, and count as one participant

  Using a video conferencing system paired to web app does not increase the participant count.
- A participant in a meeting on a SIP endpoint can use video, audio and receive a presentation. These elements comprise one callLeg, and count as one participant
- A participant on a Lync client can use any combination of chat, use video, audio and send a presentation. Any combination of these elements counts as one participant but each element is a separate callLeg. (A received presentation is displayed in the main video stream.)
- New call legs for existing participants can still be added; for example, a Lync presentation call leg to go with a Lync audio/video call leg

If creation of a call leg or participant via an API method fails because a limit has been reached, you see the appropriate " failureReason" . If an incoming connection attempt is unsuccessful because a limit has been reached, you also see an error message (with a separate callLegEnd reason for whether the call's own limit has been reached or that of its owning tenant).

9 User Related Methods

Users are created by synchronizing against LDAP servers (as discussed later); however, there are a number of methods for retrieving user information. This chapter covers:

- retrieving information on users
- retrieving detailed information on an individual user
- configuring user profiles
- applying coSpace templates to users
- retrieving users coSpace template information

9.1 Retrieving Users

GET method performed on the "/users" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those users that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see above)
tenantFilter	ID	Supply tenantFilter to return only those users associated with the specified tenant
emailFilter	String	Supply emailFilter to restrict results returned to those users whose email value exactly matches the specified email address (from version 2.1).
cdrTagFilter	String	Supply cdrTagFilter to restrict results returned to those users whose cdrTag value exactly matches the specified cdrTag (from version 2.1).

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <users total="N"> tag with potentially multiple <user> elements within it.
user id	ID	<user> elements follow the general form on the left.
userJid	String	For example, first.last@example.com
tenant	ID	The id of the tenant with which this user is associated, if applicable

9.2 Retrieving Detailed Information on an Individual User

GET method performed on a "/users/<user ID>" node

If the user ID supplied is valid, a "200 OK" response is received, with XML content of the form:

Response elements	Type/Value	Description/Notes
user id	ID	
userJid	String	e.g. first.last@example.com
tenant	ID	The id of the tenant with which this user is associated, if applicable
name	String	User's display name. It can have a maximum length of 200 characters.
email	String	e.g. first.last@mail.example.com
authenticationId	String	The id used for authentication; this value is checked against values from the certificate that the user presents during certificate-based authentication.
userProfile	ID	If present, this is the ID of the user profile associated with this user. (from version 2.0)

9.2.1 Retrieving a User's coSpace Associations

GET method performed on a "/users/<user ID>/usercoSpaces" node retrieves the coSpaces that the user is a member of. (Also see the note on coSpace member permissions for auto-generated members.)

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <usercoSpaces total="N"> tag with potentially multiple <usercoSpace> elements within it.
coSpace	ID	<usercoSpace> elements follow the general form on the left.

9.2.2 Applying coSpace templates to users

From 2.9, you can assign the coSpaceTemplate to a user with the API:

- POST to **`/users/<user id>/userCoSpaceTemplates`**

Parameters	Type/Value	Description/Notes
coSpaceTemplate	ID	the id of a coSpace template that the user is allowed to use to instantiate a coSpace (from version 2.9)

The following operations are also supported:

- DELETE on **`/users/<user ID>/userCoSpaceTemplates/<user coSpace template ID>`**

9.2.3 Retrieving users coSpace template information

- GET on **`/users/<user ID>/userCoSpaceTemplates/<user coSpace template ID>`** will give the following response parameters:

Response parameters	Type/Value	Description/Notes
coSpaceTemplate	ID	The id of a coSpace template that the user is allowed to use to instantiate a coSpace. (from version 2.9)
autoGenerated	one of true | false	Whether this coSpace template has been added automatically or manually: true - this template has been added automatically as part of an LDAP sync operation, therefore it is not possible to remove it except by modifying the parameters of the sync operation false - this template has been added via an API method. It can be modified or removed via the API. (from version 2.9)

- Enumerate GET on **`/users/<user ID>/userCoSpaceTemplates`** supporting the standard URI parameters " limit" and " offset" . The response is structured as a top-level <userCoSpaceTemplates total=" N" > tag with potentially multiple <userCoSpaceTemplate> elements underneath. Each <userCoSpaceTemplate> tag includes the request and response parameters (" coSpaceTemplate" and " autoGenerated").

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve access methods other than those in the first page of the notional list (from version 2.9)
limit		

9.2.4 Applying userCoSpaceTemplates with LDAP

From 2.9, the API object **`/ldapUserCoSpaceTemplateSources`** is introduced to allow users to create spaces using LDAP methods. This allows the template to be included directly in the source object.

The API object **`/ldapUserCoSpaceTemplateSources`** supports the following operations:

- POST to **`/ldapUserCoSpaceTemplateSources`**
- PUT to **`/ldapUserCoSpaceTemplateSources`**

Request Parameters	Type/Value	Description/Notes
coSpaceTemplate	ID	ID of the cospace template to be applied for these users

Request Parameters	Type/Value	Description/Notes
ldapSource	ID	ID of the LDAP source to be used to locate users
filter	String	Additional LDAP filter string to be applied when reading the source

- GET on **`/ldapUserCoSpaceTemplateSources/<LDAP user coSpace template source id>`**, gives the following responses:

Response elements	Type/Value	Description/Notes
coSpaceTemplate	ID	ID of the cospace template to be applied for these users
ldapSource	ID	ID of the LDAP source to be used to locate users
filter	String	Additional LDAP filter string to be applied when reading the source

- Enumerate GET on **`/ldapUserCoSpaceTemplateSources`**, gives the following responses:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve entries other than those in the first page of the notional list
limit		

Response is structured as a top-level <ldapUserCoSpaceTemplateSources total=" N" > tag with potentially multiple <ldapUserCoSpaceTemplateSource> elements within it.

Each <ldapUserCoSpaceTemplateSource> tag may include the following elements:

Response elements	Type/Value	Description/Notes
coSpaceTemplate	ID	ID of the cospace template to be applied for these users
ldapSource	ID	ID of the LDAP source to be used to locate users

9.2.5 Retrieving user provisioned coSpace information

Version 3.1 introduces this API object to support the following operations:

- Enumeration of GET on **`/users/<user id>/userProvisionedCoSpaces`**
- GET on **`/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id>`**

Enumeration of **`/users/<user id>/userProvisionedCoSpaces`** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		An offset and limit can be supplied to retrieve user provisioned coSpaces other than those in the first page of the notional list. (from version 3.1)
limit		

Response is structured as a top-level <userProvisionedCoSpaces total=" N" > tag with potentially multiple <userProvisionedCoSpace> elements within it.

Each <userProvisionedCoSpace> tag may include the following elements:

Response elements	Type/Value	Description/Notes
coSpaceTemplate	ID	coSpaceTemplate that this coSpace will be based on when it is instantiated. (from version 3.1)
uriHint	string	Basis for the uri of this coSpace (if this clashes with other uris when the space is instantiated, a unique uri will be generated based on this hint). (from version 3.1)
name	string	name that this coSpace will have when it is instantiated. (from version 3.1)
coSpace	ID	If present, ID of the coSpace this userProvisionedCoSpace was instantiated into. (from version 3.1)

GET on individual user provisioned coSpace with **`/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id>`** gives the following response:

Response elements	Type/Value	Description/Notes
coSpaceTemplate	ID	coSpaceTemplate that this coSpace is based on when it is instantiated. (from version 3.1)
uriHint	string	Basis for the uri of this coSpace (if this clashes with other uris when the space is instantiated, a unique uri will be generated based on this hint). (from version 3.1)
name	string	name that this coSpace has when it is instantiated. (from version 3.1)
coSpace	ID	If present, ID of the coSpace this userProvisionedCoSpace was instantiated into. (from version 3.1)

9.2.6 Creating, modifying, and retrieving LDAP user provisioned coSpace sources

This new API object supports the following operations:

- POST to **`/ldapUserProvisionedCoSpaceSources`**
- PUT to **`/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace source id>`**

Parameter	Type/Value	Description/Notes
ldapSource (*)	ID	ID of the LDAP source to be used to locate users (from version 3.1)

Parameter	Type/Value	Description/Notes
ldapUserProvisionedCoSpaceMapping (*)	ID	Mapping to be used to generate the name and uri-hint of the user provisioned coSpaces (from version 3.1)
filter	string	Additional LDAP filter string to be applied when reading the source (from version 3.1)

- Enumeration of **`/ldapUserProvisionedCoSpaceSources`** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve entries other than those in the first page of the notional list (from version 3.1)
limit		

Response is structured as a top-level <ldapUserProvisionedCoSpaceSources total=" N" > tag with potentially multiple <ldapUserProvisionedCoSpaceSource> elements within it.

Each <ldapUserProvisionedCoSpaceSource> tag may include the following elements:

Response elements	Type/Value	Description/Notes
ldapSource	ID	ID of the LDAP source used to locate users (from version 3.1)
ldapUserProvisionedCoSpaceMapping	ID	Mapping used to generate the name and uri-hint of the user provisioned coSpaces (from version 3.1)
filter	string	Additional LDAP filter string applied when reading the source (from version 3.1)

- GET on individual LDAP user provisioned coSpace sources with **`/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace mapping id>`** gives the following response:

Response elements	Type/Value	Description/Notes
ldapSource	ID	ID of the LDAP source used to locate users (from version 3.1)
ldapUserProvisionedCoSpaceMapping	ID	Mapping used to generate the name and uri-hint of the user provisioned coSpaces (from version 3.1)
filter	string	Additional LDAP filter string applied when reading the source (from version 3.1)

9.2.7 Creating, modifying, and retrieving LDAP user provisioned coSpace mappings

This new API object supports the following operations:

- POST to **`/ldapUserProvisionedCospaceMappings`**
- PUT to **`/ldapUserProvisionedCospaceMappings/<LDAP user provisioned coSpace mapping id>`**

Parameter	Type/Value	Description/Notes
coSpaceUriMapping (*)	string	similar to the mappings of the ldapMappings object, template for generating URI of the user provisioned coSpaces. (from version 3.1)
coSpaceNameMapping	string	similar to the mappings of the ldapMappings object, template for generating the name of the user provisioned coSpaces. (from version 3.1)
coSpaceTemplate (*)	ID	coSpace template to use for the user provisioned coSpaces. (from version 3.1)

- Enumeration of **`/ldapUserProvisionedCospaceMappings`** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve entries other than those in the first page in the notional list
limit		

Response is structured as a top-level <ldapUserProvisionedCospaceMappings total=" N" > tag with potentially multiple <ldapUserProvisionedCospaceMapping> elements within it.

Each <ldapUserProvisionedCospaceMapping> tag may include the following elements:

Response elements	Type/Value	Description/Notes
coSpaceUriMapping	string	similar to the mappings of the ldapMappings object, template for generating URI of the user provisioned coSpaces. (from version 3.1)
coSpaceNameMapping	string	similar to the mappings of the ldapMappings object, template for generating name of the user provisioned coSpaces. (from version 3.1)
coSpaceTemplate	ID	coSpace template used for the user provisioned coSpaces. (from version 3.1)

- GET on individual LDAP user provisioned coSpace mappings with

`/ldapUserProvisionedCospaceMappings/<LDAP user provisioned coSpace mapping id>` gives the following response:

Response elements	Type/Value	Description/Notes
coSpaceUriMapping	string	similar to the mappings of the ldapMappings object, template for generating URI of the user provisioned coSpaces. (from version 3.1)
coSpaceNameMapping	string	similar to the mappings of the ldapMappings object, template for generating name of the user provisioned coSpaces. (from version 3.1)
coSpaceTemplate	ID	coSpace template used for the user provisioned coSpaces. (from version 3.1)

9.3 User Profile Methods

User profiles control the facilities provided to the users in the profile, for instance whether they can create new coSpaces, create new calls, make phone calls, pair to video conferencing systems, allowed to send and receive chat messages when in a point to point call with another user. For more information see also Section 16.

9.3.1 Retrieving user profiles

GET method performed on the “/userProfiles” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see above).
usageFilter	Referenced| unreferenced	Using unreferenced retrieves only those user profiles that are not referenced by global settings or any other object. This is a useful check before deleting a user profile. To retrieve just those user profiles which are referenced in at least one place, you can supply " usageFilter=referenced"

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <userProfiles total=" N" > tag with potentially multiple <userProfile> elements within it.

Response elements	Type/Value	Description/Notes
audioParticipationAllowed	true|false	Whether or not a user associated with this user profile and using the web app will be allowed to send or receive live audio when in a call. This restriction does not apply to dialing directly into the call via SIP, or slaving to a SIP endpoint. (from version 2.0)
videoParticipationAllowed	true|false	Whether or not a user associated with this user profile will be allowed to send or receive live video when in a call. This restriction does not apply to dialing directly into the call via SIP, or slaving to a SIP endpoint. (from version 2.0)
presentationParticipationAllowed	true|false	Whether or not a user associated with this user profile will be allowed to send or receive presentation media when in a call. This restriction does not apply to dialing directly into the call via SIP, or slaving to a SIP endpoint. (from version 2.0)
hasLicense	true|false	Whether or not a user associated with this user profile has a Cisco user license. (from version 2.0)

9.3.2 Setting up and modifying user profiles

- Creating: POST method to the /userProfiles node
- Modifying: PUT to /userProfiles/<user profile id>

Parameters	Type/Value	Description/Notes
audioParticipationAllowed	true|false	Whether or not a user associated with this user profile and using the web app will be allowed to send or receive live audio when in a call. This restriction does not apply to dialing directly into the call via SIP, or slaving to a SIP endpoint. This parameter is present from version 2.0 onwards.
videoParticipationAllowed	true|false	Whether or not a user associated with this user profile will be allowed to send or receive live video when in a call. This restriction does not apply to dialing directly into the call via SIP, or slaving to a SIP endpoint. This parameter is present from version 2.0 onwards.

Parameters	Type/Value	Description/Notes
presentationParticipationAllowed	true|false	Whether or not a user associated with this user profile will be allowed to send or receive presentation media when in a call. This restriction does not apply to dialing directly into the call via SIP, or slaving to a SIP endpoint. This parameter is present from version 2.0 onwards.
hasLicense	true|false	Whether or not a user associated with this user profile has a Cisco user license. This parameter is present from version 2.0 onwards.

9.3.3 Retrieving detailed information about an individual user profile

GET method performed on a " /userProfiles/<user profile id >" node. If the user profile ID supplied is valid, a “200 OK” response is received, with XML content matching the section above.

10 System Related Methods

This chapter details the API methods related to management of the system. The chapter covers:

- retrieving system status
- retrieving system alarm status
- retrieving system database status
- retrieving and setting the URI of the CDR receivers
- retrieving and setting the global profile
- retrieving licensing information
- configuring the TURN server
- configuring the Web Bridge
- configuring Web Bridge Profiles
- configuring the Call Bridge
- configuring Call Bridge groups
- configuring Call Bridge clustering
- system load
- system diagnostics
- system timedlogging

10.1 Retrieving System Status

GET method performed on the "/system/status" node.

Response elements	Type/Value	Description/Notes
		Returns the elements on the left within a "<status>" element:
hostId	String	An ID that uniquely identifies this Meeting Server instance.
softwareVersion	String	The software version currently running on the Call Bridge
uptimeSeconds	Number	The length of time that the unit has been running
activated	true|false	(Deprecated in 3.0) Whether the Call Bridge is currently activated (licensed). Currently always true for a Meeting Server. Currently always true for a Meeting Server.
clusterEnabled	true|false	If set to true then the Call Bridge is currently running with clustering enabled. This parameter is present from version 2.0 onwards

Response elements	Type/Value	Description/Notes
clusterId	ID	The Id that represents cluster that the Call Bridge is in, constant throughout the lifetime of the cluster. For the purposes of this parameter an unclustered Call Bridge is regarded as a cluster of 1, and so this parameter will still have a value for a single Meeting Server instance. (3.0 onwards)
cdrTime	Number	The current timestamp as would be written to a CDR generated at the time the request is received. This will be in the same format as the " time" field in CDRs themselves (see RFC 3339, for instance " 2014-02-11T12:10:47Z").
callLegsActive	Number	The number of active call legs at the time of the request
callLegsMaxActive	Number	The highest number of call legs simultaneously active on this Meeting Server.
callLegsCompleted	Number	The total number of call legs that have been active but are no longer connected / present
audioBitRateOutgoing	Number	The current total bit rate (in bits per second) summed over all outgoing audio streams (audio media sent from the Meeting Server to a remote party)
audioBitRateIncoming	Number	The current total bit rate for incoming audio streams
videoBitRateOutgoing	Number	The current total bit rate for outgoing video streams
videoBitRateIncoming	Number	The current total bit rate for incoming video streams
cdrCorrelatorIndex	Number	The correlator index of the next CDR record that will be sent. When no CDR records have been sent, this will have the value of 0. (From version 2.2).

10.2 Retrieving System Alarm Status

GET method performed on the "/system/alarms" node. An offset and limit can be supplied to retrieve alarm conditions other than those in the first page of the notional list. This method returns a table detailing the currently active system-wide alarm conditions.

Response elements	Type/Value	Description/Notes
		Returns a list of individual "<alarm>" elements. If there are no currently active alarm conditions, this list will be empty. Each active alarm condition will have an "alarm" tag, which contains:
Id	ID	A unique ID for this instance of this fault condition
activeTimeSeconds	Number (see right)	The amount of time that this alarm condition has been active.

Response elements	Type/Value	Description/Notes
type		One of: • callBrandingResourceInvalid – the supplied resource has an invalid format; the call branding profile is specified by the accompanying " callBrandingProfiles" GUID parameter and the problematic file is specified by the accompanying " fileName" text parameter • callBridgeConnectionFailure – the Call Bridge has failed to establish a connection to one of its peer clustered Call Bridges which specified by the accompanying " callBridge" GUID parameter • callDistributionFailure – the Call Bridge has failed to establish a distributed link for one of its active calls; the Call Bridge that the link should have been to is identified by the accompanying " callBridgeName" text parameter, and the call is present as the " call" GUID parameter • cdrConnectionFailure – the Meeting Server has failed to establish a connection to its configured CDR receiver, and therefore may be unable to push out new call detail records • c2wConnectionFailure – the Call Bridge has been unable to establish a C2W connection to the configured Web Bridge (3.0 onwards) • databaseClusterNodeOutofSync – a node in the database cluster is out of sync and is not syncing • databaseConnectionError – Meeting Server has failed to establish a connection to its database • guestAccountConnectionFailure – the Meeting Server has been unable to establish a connection to the configured Web Bridge in order to allow guest logins (3.0 removed) • ivrBrandingResourceInvalid – the supplied resource has an invalid format; the IVR branding profile is specified by the accompanying " ivrBrandingProfile" GUID parameter and the problematic file is specified by the accompanying " fileName" text parameter • licenseGrace – a feature license has passed the expiry date and will soon be deactivated (from version 2.1) • licenseExpired – a feature license has expired and has been deactivated (from version 2.1) • recorderLowDiskSpace – a recorder has limited disk space; the recorder is specified by the accompanying " recorder" GUID parameter (3.0 removed) • recorderUnavailable – the Call Bridge has not managed to successfully contact a configured recorder; the recorder is specified by the accompanying " recorderAddress" parameter • streamerUnavailable – the Call Bridge has not managed to successfully contact a configured streamer; the streamer is specified

Response elements	Type/Value	Description/Notes
		by the accompanying " streamerAddress" parameter • turnServerUnavailable – the Call Bridge has not managed to contact a configured TURN server; this TURN server is specified by the accompanying " turnServer" GUID parameter • webBridgeArchivePushFailure – the Call Bridge has not been able to push a required customization archive to a Web Bridge (3.0 removed) • webBridgeArchiveRetrievalFailure–the Call Bridge has not been able to retrieve a required Web Bridge customization archive (3.0 removed) • webBridgeBackgroundImagePushFailure – the Call Bridge has not been able to push a required customized background image file to a Web Bridge (3.0 removed) • webBridgeBackgroundImageRetrievalFailure – the Call Bridge has not been able to retrieve a required customized background image file (3.0 removed) • webBridgeLoginLogoImagePushFailure – the Call Bridge has not been able to push a required customized login logo image to a Web Bridge (3.0 removed) • webBridgeLoginLogoImageRetrievalFailure – the Call Bridge has not been able to retrieve a required customized login logo image file (3.0 removed) • webBridgeXmppCertificatePushFailure – the Call Bridge has not been able to push the XMPP certificate to a Web Bridge (3.0 removed) • xmppAuthenticationRegistrationFailure – the Meeting Server has not been able to register successfully with the named XMPP authentication component (3.0 removed) • xmppRegistrationFailure – the Meeting Server has not been able to register successfully with its configured XMPP Server (3.0 removed)

Response elements	Type/Value	Description/Notes
failureReason		For some of the alarm types above, additional information is provided about the cause of that particular failure: • authenticationFailure • connectFailure – a failure to connect to a remote destination; for example, a TCP or TLS connection was not able to be established • dataFormatInvalid – the Call Bridge is configured to use a particular data set (for example, a remotely-hosted resource file) that it then found was not in a useable format • destinationReadOnly – a critical resource, for example the database, was found by the Call Bridge to be read only when write access was needed • dnsFailure – a failure to resolve the host name of a remote destination; for example, as part of the process of establishing a connection to a remote system • error • fileNotFound – the alarm condition was raised because the Call Bridgefailed to load a required file; for instance, if the resourceArchive needed for a web bridge was not able to be retrieved from a remote server • fileSizeLimitExceeded – the Call Bridge is configured to use a resource, for example, a remotely-hosted resource file, that was not able to be retrieved because it exceeded an internal file size limit • internalServerError – the Call Bridge was not able to perform an operation, for example the upload or download of a resource file, because the remote party returned " internal server error" when the operation was attempted • serviceUnavailable

10.3 Retrieving System Database Status

GET method performed on the “/system/database” node.

Response elements	Type/Value	Description/Notes
clustered	enabled|disabled	Whether database clustering is enabled.
cluster		If clustering is enabled then the <cluster> element includes the elements on the left
Error	String	Error description
totalNodes	Number	Number of database nodes in the cluster

Response elements	Type/Value	Description/Notes
nodeInUse	String	Which database node is currently in use (the primary database)
node		The node element is returned for each database in the cluster with the following details: • hostname: the hostname or IP address of the node • up: if the node is visible from this Call Bridge (true|false) • syncBehind: approximate number of bytes that this node is behind the current state of the primary database. 0 means in sync, and -1 means the calculation is unavailable • master: whether this node is the primary database (true|false), if set to false it is a replica database

10.4 CDR Receiver URI Methods

Note: /system/cdrReceiver is deprecated, use the /system/cdrReceivers object which supports multiple CDR receivers.

10.4.1 Retrieving the CDR Receivers URI

You can find the URIs of the CDR receivers through the API (as well as the Web Admin Interface). Issue a GET on the /system/cdrReceivers node to retrieve the URIs that are the full URLs of the configured CDR receivers.

This method accesses the URIs of the CDR receivers using the Web Admin Interface **Configuration > CDR settings** page.

GET method performed on the "/system/cdrReceivers" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to CDR receivers other than the first " page" in the notional list

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <cdrReceivers total=" N" > tag with potentially multiple <cdrReceiver> elements within it. Each <cdrReceiver> tag may include the following element:
uri	String	Full URL of the configured CDR receiver address

Note: GET of /system/cdrReceivers/<*cdr receiver id*>allows you to retrieve the configuration for a single specified CDR receiver.

10.4.2 Setting the CDR Receivers URI

Set the CDR receiver URI through the API (as well as the Web Admin Interface). You can issue a PUT or a POST on the /system/cdrReceivers node.

Use POST and specify a " url" value to create and configure the CDR receiver in one operation, or use PUT to initially create the CDR receiver but configure the “url” separately later.

Parameters	Type/Value	Description/Notes
uri *	String	Full URL to which CDRs will be sent.

If the creation is successful, you should receive a “200 OK” response and a " Location: /api/v1/system/cdrReceivers/<*cdr receiver id*>" object reference; if too many CDR receivers are already configured, you will receive a " tooManyCdrReceivers" error (in a " failureDetails" section).

Note: If you perform a PUT with an empty " url" to the legacy /system/cdrReceiver node, any GUID associated with that CDR receiver is removed, and effectively that CDR receiver is no longer present. If you later PUT a non-empty " url" value to the same (legacy) node, a new GUID will be generated for that CDR receiver.
If you perform a PUT with an empty " url" to a non-legacy CDR receiver (/system/cdrReceivers/<*cdr receiver id*>) then that CDR receiver remains with the same GUID, but no " url" value. It will continue to show up in GET operations. This is because there is an explicit " DELETE" method (section 4.4) for the new CDR receiver objects, whereas for the legacy CDR receiver the only deconfiguration method is to set its location to the empty value.

To set or update the URIs of the CDR receivers via the Web Admin Interface use the **Configuration > CDR settings** page.

10.5 Global Profile Methods

10.5.1 Retrieving the Global Profile

GET to /system/profiles returns the values described in the following section.

10.5.2 Setting the Global Profile

You can set (or unset) the callLegProfile ID value under api/v1/system/profiles to impose (or remove) a top-level profile.

PUT or POST to /api/v1/system/profiles. Supplying an empty value unsets the top-level profile.

Parameters	Type/Value	Description/Notes
callLegProfile	ID	Sets the top level call leg profile to the one with the specified ID.
callProfile	ID	Sets the top level call profile to the one specified
dtmfProfile	ID	Sets the top level DTMF profile to the one specified
userProfile	ID	Sets the top level user profile to the one specified
ivrBrandingProfile	ID	Sets the top level IVR branding profile to the one specified
callBrandingProfile	ID	Sets the top level call branding profile to the one specified
compatibilityProfile	ID	Sets the top level compatibility profile to the one specified (2.1 onwards)
lobbyProfile	ID	Set the top level lobby profile to the one specified (3.7 onwards)
dialInSecurityProfile	ID	Sets the top level dial in security profile to the one specified (3.0 onwards)
webBridgeProfile	ID	Sets the top level web bridge profile to the one specified (3.0 onwards)

10.6 Licensing Methods

Note: From version 2.4, a license is not required to apply single or multiple branding.

The API methods related to licensing are:

- retrieving information on uploaded licenses and on license consumption

Previously, the existing **`/system/licensing`** API returned the contents of the license file, i.e. the feature components for a Meeting Server, together with each component's license status and expiry date (if applicable) shown. For example, whether the callbridge license was activated or not on that Meeting Server, and if licensed, the expiry date.

From 3.0, the existing **`/system/licensing`** API now only returns the contents of the license file (i.e. the feature components) on a per Meeting Server instance, and the newly introduced API object /clusterLicensing returns the license status and expiry date (if applicable) for a Meeting Server cluster.

Note:
The new **`/clusterLicensing`** API represents the cluster (a single Meeting Server deployment is regarded as a cluster of one). The **`/system/licensing`** API that represents the license file contents continues to be per Meeting Server instance.

10.6.1 Retrieving license file information on a per Meeting Server instance

GET method performed on the "/system/licensing" node. See Appendix C.1 for an example of the structure for the information.

Response elements	Type/Value	Description/Notes
features		If licensing is enabled then the <features> element includes the elements below.

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>callBridge</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense|
activated|
grace|
expired</td><td>No license applied to the Call Bridge, the Call Bridge cannot make any calls
License applied and Call Bridge activated with media encryption. Media will be encrypted in calls
License expired, now in grace period for license renewal
License expired for Call Bridge with media encryption</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
</table>
</td></tr>
<tr><td>callBridgeNoEncryption
(from version 2.4)</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense|
activated|
grace|
expired</td><td>No license applied to the Call Bridge, the Call Bridge cannot make any calls
License applied and Call Bridge activated without media encryption. Media will not be encrypted in calls
License expired, now in grace period for license renewal
License expired for Call Bridge without media encryption</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
</table>
</td></tr>
<tr><td>webBridge</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense|
activated|
grace|
expired</td><td>No license applied to the Web Bridge
License applied and Web Bridge activated
License expired, now in grace period for license renewal
License expired for Web Bridge</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Response elements	Type/Value	Description/Notes

Name	Type/Value	Description
turn		

Name	Type/Value	Description
status	noLicense| activated| grace| expired	No license applied to the TURN server License applied and TURN server activated License expired, now in grace period for license renewal License expired for TURN server
expiry	String	Date of expiry

Name	Type/Value	Description
ldap		

Name	Type/Value	Description
status	noLicense| activated| grace| expired	No license applied to the LDAP server License applied and LDAP server activated License expired, now in grace period for license renewal License expired for LDAP server
expiry	String	Date of expiry

Name	Type/Value	Description
branding		

Name	Type/Value	Description
status	From version 2.4, this is no longer relevant, as a branding license is not required.	
status (prior to version 2.4)	noLicense| activated| grace| expired	No license applied for branding License applied and branding activated License expired, now in grace period for license renewal License expired for branding
expiry	String	Date of expiry

Response elements	Type/Value	Description/Notes

Name	Type/Value	Description
recording		

Name	Type/Value	Description
status	noLicense| activated| grace| expired	No license applied to the Recorder License applied and Recorder activated License expired, now in grace period for license renewal License expired for Recorder
expiry	String	Date of expiry
limit	Number	

streaming	

Name	Type/Value	Description
status	noLicense| activated| grace| expired	No license applied to the Streamer License applied and Streamer activated License expired, now in grace period for license renewal License expired for Streamer
expiry	String	Date of expiry
limit	Number	

personal	

Name	Type/Value	Description
status	noLicense| activated| grace| expired	No Personal Multiparty license applied Personal Multiparty license activated License expired, now in grace period for license renewal Personal Multiparty license expired
expiry	String	Date of expiry
limit	Number	

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>shared</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense|
activated|
grace|
expired</td><td>No Shared Multiparty license applied
Shared Multiparty license activated
License expired, now in grace period for license renewal
SharedI Multiparty license expired</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
<tr><td>limit</td><td>Number</td><td></td></tr>
</table>
</td></tr>
<tr><td>capacityUnits</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense|
activated|
grace|
expired</td><td>No license for Capacity Units applied
License for Capacity Units activated
License expired, now in grace period for license renewal
License for Capacity Units expired</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
<tr><td>limit</td><td>Number</td><td></td></tr>
</table>
</td></tr>
<tr><td>customizations</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense|
activated|
grace|
expired</td><td>You can configure custom layouts with or without the license key, however without the key Meeting Server will behave as though no custom layout is configured.</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
</table>
</td></tr>
</table>
</td><td></td></tr>
</table>

GET method performed on the “/system/multipartyLicensing” node. See Appendix C.2 for an example of the structure for the information.

Response elements	Type/Value	Description/Notes
timestamp	string	UTC time at which the report was generated
personalLicenseLimit	Number	(Deprecated in 3.0) Number of available personal licenses
sharedLicenseLimit	Number	(Deprecated in 3.0) Number of available shared licenses
capacityUnitLimit	Number	(Deprecated in 3.0) Number of available capacity units
users	Number	Number of non-guest users on the system
personalLicenses	Number	Number of personal licenses assigned to users
participantsActive	Number	Number of active participants
callsActive	Number	Number of active calls
weightedCallsActive	Number	Number of weighted active calls (see note below).
capacityUnitUsage	Number	Number of capacity units in use
callsWithoutPersonalLicense	Number	Number of calls without a personal license. Deprecated from version 2.6, change in license counting means that this value can no longer be used to accurately calculate SMP usage.
weightedCallsWithoutPersonalLicense	Number	Number of weighted calls without a personal license (see note below).Number of weighted calls without a personal license. Deprecated from version 2.6, change in license counting means that this value can no longer be used to accurately calculate SMP usage.
capacityUnitUsageWithoutPersonalLicense	Number	Number of capacity units in use in calls without a personal license

GET method performed on the “/system/multipartyLicensing/activePersonalLicenses” .

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve active personal license entries other than those in the first " page" of the notional list

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <activePersonalLicenses total=" N" > tag with potentially multiple <user> elements within it. Each <user> tag may include the following elements:
callsActive	Number	Number of active calls that use the license of this user
weightedCallsActive	Number	Number of weighted active calls that use the license of this user (see note below).

Note: the sum of weighted calls across a cluster matches the number of distinct calls on the cluster. For example, if CMS1 shows 3 callsActive and 2 weightedCallsActive, and CMS2 shows 2 callsActive and1 weightedCallsActive, then there are 3 conferences in total on the cluster and 3 licenses are required.

GET method performed on the "/system/MPLicenseUsage" node (from version 2.6).

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve active personal license entries other than those in the first " page" of the notional list
startTime	String	UTC time (formated as per RFC 3339) specifying the earliest date-time from which license usage snapshots should be fetched (inclusive)
endTime	String	UTC time (formated as per RFC 3339) specifying the latest date-time from which license usage snapshots should be fetched (inclusive)
hostId	ID	I.D. of the host (as returned by system/status) for which you want to retrieve license usage snapshots

Response elements	
event	

Name	Type/Value	Description
Parameters included in parent tag		
time	String	UTC time at which the license usage event was generated
Response values		
pmp	Number	Number of personal licenses in use (each license normalized over the number of Call Bridges that the call spans)
pmpAssigned	Number	Number of personal licenses assigned to users in the cluster (3.0 onwards)
smpAud	Number	Number of shared license audio-only calls (each license normalized over the number of Call Bridges that the call spans)
smpPtP	Number	Number of shared license point-to-point calls, which aren't audio-only (each license normalized over the number of Call Bridges that the call spans)
smpFull	Number	Number of shared licenses in use, which aren't audio-only or point-to-point (each license normalized over the number of Call Bridges that the call spans)
rec	Number	Number of calls being recorded
str	Number	Number of calls being streamed

GET method performed on the “/system/MPLicenseUsage/knownHosts” node (from version 2.6).

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve host ids other than those in the first " page" of the notional list

Response elements	
host	

Name	Type/Value	Description
Parameters included in parent tag		
id	ID	Unique host id of a Call Bridge whose license utilization information can be accessed from the system/MPLicenseUsage API node

10.6.2 Retrieving cluster licensing information

From 3.0 onwards, a GET operation on the existing **`/system/licensing`** API now only returns the contents of the license file (i.e. the feature components) on a per Meeting Server instance. The newly introduced API object **`/clusterLicensing`** returns the license status and expiry date (if applicable) for a Meeting Server cluster.

Note: The expiry date field returned for **`/clusterLicensing`** will only ever be up to a maximum of 90 days in the future.

To retrieve the current license information for your Meeting Server or cluster:

GET method performed on **`/clusterLicensing`** gives the following:

Response elements	Type/Value	Description/Notes
features		If licensing is enabled then the <features> element includes the elements below.

Response elements	Type/Value	Description/Notes

Name	Type/Value	Description
callBridge		

Name	Type/Value	Description
status	noLicense| activated| expired	Status of the license: • noLicense – no license is available for this feature • activated – the feature is licensed and within its expiry date • expired – the license for this feature is past its expiry date
expiry	String	Date of expiry

Name	Type/Value	Description
callBridgeNoEncryption		

Name	Type/Value	Description
status	noLicense| activated| expired	Status of the license: • noLicense – no license is available for this feature • activated – the feature is licensed and within its expiry date • expired – the license for this feature is past its expiry date
expiry	String	Date of expiry

Name	Type/Value	Description
customizations		

Name	Type/Value	Description
status	noLicense| activated| expired	Status of the license: • noLicense – no license is available for this feature • activated – the feature is licensed and within its expiry date • expired – the license for this feature is past its expiry date
expiry	String	Date of expiry

<table>
<tr><th>Response elements</th><th>Type/Value</th><th>Description/Notes</th></tr>
<tr><td></td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>recording</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense| activated| expired</td><td>Status of the license: noLicense - no license is available for this feature activated - the feature is licensed and within its expiry date expired - the license for this feature is past its expiry date</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
</table>
</td></tr>
<tr><td>snapshot</td><td colspan="2">
<table>
<tr><th>Name</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>status</td><td>noLicense| activated| expired</td><td>Status of the license: noLicense - no license is available for this feature activated - the feature is licensed and within its expiry date expired - the license for this feature is past its expiry date</td></tr>
<tr><td>expiry</td><td>String</td><td>Date of expiry</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Version 3.0 introduces the /clusterLicensing/raw API which is purely for Cisco Meeting Management to give / retrieve license information to allow it to administer Smart Licensing. This API is not intended for general use and requires no administrator configuration.

10.7 TURN Server Methods

Note: The TURN Server is not available on the Cisco Meeting Server 2000. It is more suited to the lower capacity Cisco Meeting Server 1000 and specification-based VM servers.

Note: The TURN server component always supports the standard port 3478 for UDP. When deploying Cisco Meeting Server web edge, the API node `/turnServers` " type" parameter should be set to " cms" . If this parameter is unset, it defaults to " standard" , and tells the clients to use TCP/UDP port 443 to connect to the TURN server. For more information on

the " type" parameter values, see the section *Setting up and modifying TURN servers* in Cisco Meeting Server API Reference Guide.

10.7.1 Retrieving Information on TURN Servers

GET method performed on the “/turnServers” node.

Parameters	Type/Value	Description/Notes
filter offset limit	String Number Number	Supply filter=<string> in the URI to return just those TURN servers that match the filter An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see above).

Response elements	Type/Value	Description/Notes
serverAddress clientAddress		Response is structured as a top-level <turnServers total=" N" > tag with potentially multiple <turnServer> elements within it. Each <turnServer> tag may include the following elements. See the next section

10.7.2 Setting up and modifying TURN servers

- Creating: POST method to the /turnServers" node
- Modifying: PUT to " /turnServers/<turn server id>"

Parameters	Type/Value	Description/Notes
serverAddress	String	The address for the Call Bridge to use to reach this TURN server
clientAddress	String	The address that web apps should use to reach this TURN Server
username	String	The username to use when making allocations on this TURN Server (when short term credential mode is disabled)
password	String	The password to use when making allocations on this TURN Server (when short term credential mode is disabled)
useShortTermCredentials	true | false	Whether or not short term credentials should be used on this TURN server. If this parameter is not supplied in a create (POST) operation, it defaults to " false" .

Parameters	Type/Value	Description/Notes
sharedSecret	string	The shared secret that should be used when making allocations on this TURN server (when short term credential mode is enabled).
type	acano| cms| expressway| lyncEdge| **standard**	Select “cms” or “acano” if using the TURN server within the Meeting Server; tells the client to use UDP/TCP port 3478 for lync and TCP port 443 for web app, to connect to the TURN server. Select “expressway” if connecting to the Cisco Expressway rather than using the TURN server in the Meeting Server; tells the client to use UDP/TCP port 3478 for lync and TCP port 443 for web app. Select “lyncEdge” if connecting to a Lync or Skype for Business deployment; tells the client to use port 443 to connect to the TURN server. If the “type” field is not set it defaults to “standard”; tells the client to use port 443 to connect to the TURN server.
numRegistrations	Number	The number of registrations that should be made to this TURN Server. This parameter is only meaningful for configured Lync Edge servers
tcpPortNumberOverride	Number	An optional override for the port number to use when using this TURN Server for TCP media (for example, Lync presentation call legs). This parameter is not needed for configured Lync Edge servers, where the TCP port number can always be determined automatically.
callBridge	ID	If specified, associate this TURN server with the supplied Call Bridge (from version 2.1)
callBridgeGroup	ID	If specified, associate this TURN server with the supplied Call Bridge group (from version 2.1)

10.7.3 Retrieving detailed information about an individual TURN server

GET method performed on a “/turnServers/<turn server id>" node. If the TURN server ID supplied is valid, a “200 OK” response is received, with XML content:

Note: Short-term TURN credentials are never accessible via an API GET operation to /turnServers, nor is the shared secret.

Response elements	Type/Value	Description/Notes
serverAddress	String	The address on which the Call Bridge reaches this TURN server

Response elements	Type/Value	Description/Notes
clientAddress	String	The address on which web apps reach TURN Server
username	String	The username to use when making allocations on this TURN Server
useShortTermCredentials	true | false	Whether or not short term credentials are used on this TURN server.
type	acano|cms| express-way| lyncEdge| **standard**	Indicates type of TURN server, See PUT/POST for /turnServers for details.
numRegistrations	Number	For configured Lync Edge servers only the number of registrations made to this TURN Server.
tcpPortNumberOverride	Number	The port number used when using TURN server for TCP media.
callBridge	ID	If specified, this is the Call Bridge associated with this TURN server.
callBridgeGroup	ID	If specified, this is the Call Bridge Group associated with this TURN server.

10.7.4 Retrieving individual TURN Server status

GET method performed on a “/turnServers/<turn server id>/status" node. If the turn server ID supplied is valid, a “200 OK” response is received, with XML content:

Response elements	Description/Notes
host	Zero, one or more <host> child nodes, with each including the following elements:

<table>
<tr><th>Response elements</th><th>Description/Notes</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>address</td><td>String</td><td></td></tr>
<tr><td>portNumber</td><td>Number</td><td></td></tr>
<tr><td>reachable</td><td>true|false</td><td>true if this TURN server is currently reachable, false if it is not</td></tr>
<tr><td>roundTripTimeMs</td><td>Number</td><td>If this TURN server is reachable, the round trip time (in milliseconds) of the Call Bridge's path to it</td></tr>
<tr><td>mappedAddress</td><td>String</td><td rowspan="2">If populated, indicates the source IP and source port that the TURN server saw the STUN binding request coming from when theCall Bridge performed TURN Server reachability checks. This can be different to the IP address of the Call Bridge in deployments where there is a NAT between the Call Bridge and the TURN Server.</td></tr>
<tr><td>mappedPortNumber</td><td>Number</td></tr>
</table>
</td></tr>
</table>

10.8 Web Bridge Methods

10.8.1 Retrieving Information on Web Bridges

GET method performed on the “/webBridges” node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those Web Bridges that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see above).
tenantFilter	ID	Supply tenantFilter to return only those Web Bridges associated with the specified tenant.
callBridgeGroupFilter	ID	If callBridgeGroupFilter is supplied, only those web bridges within the specified call bridge group will be returned.

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <webBridges total=" N" > tag with potentially multiple <webBridge> elements within it. Each <webBridge> tag may include the following elements. See the next section
url	URL	
resourceArchive	URL	

10.8.2 Setting Up and Modifying a Web Bridge

- Creating: POST method to the " /webBridges" node
- Modifying: PUT to " /webBridges/<web bridge id>"

Parameters	Type/Value	Description/Notes
url	URL	The address for the Call Bridge to use to reach this Web Bridge
tenant	ID	If you supply the ID for a tenant to associate with this Web Bridge, only call IDs for coSpaces owned by that tenant can be joined through it
tenantGroup	ID	Only coSpaces associated with tenants within the specified tenant group can be accessed by call ID through this web bridge. If no tenant group is supplied, only coSpaces without a tenant, or associated with a tenant in no tenant group, can be accessed by call ID.
callBridge	ID	If specified, associate this Web bridge with the supplied Call Bridge (from version 2.1)
callBridgeGroup	ID	If specified, associate this Web bridge with the supplied Call Bridge group (from version 2.1)
webBridgeProfile	ID	If specified, associates this web bridge with the specified web bridge profile. (3.0 onwards)

10.8.3 Retrieving detailed information about an individual Web Bridge

GET method performed on a “/webBridges/<web bridge id>” node. If the web bridge ID supplied is valid, a “200 OK” response is received, with XML content described in the previous section.

10.8.4 Updating the Web Bridge customization

A POST to " /webBridges/<web bridge id>/updateCustomization" node causes any configured customization archive for the specified Web Bridge to be re-retrieved and pushed to that Web Bridge. For example, this allows the contents of a customization archive to be changed, and for those changes to take effect without needing to restart either the Call Bridge or the Web Bridge.

10.8.5 Retrieving diagnostics on a Web Bridge (from 2.2)

GET method performed on a "/webBridges/<web bridge id>/status" node. If the web bridge ID supplied is valid, a "200 OK" response is received, with XML content matching the table below.

Response elements	Type/Value	Description/Notes
status	unused|	The Web Bridge is not used by the queried Call Bridge
	success|	The Web Bridge is connected to the queried Call Bridge
	connectionFailure	The Web Bridge could not connect to the queried Call Bridge
	dnsFailure	The configured Web Bridge url could not be resolved (3.0 onwards)

10.9 Web Bridge Profile Methods

This allows you to configure some Web Bridge configuration options in a common place rather than solely on a per Web Bridge basis – you can now apply the same settings for all, or a specified group of Web Bridges.

To support this change, the /webBridgeProfiles API object is introduced which contains the various Web Bridge configuration options. A newly defined Web Bridge profile can be assigned to the individual webBridge objects, or to the top level (global) profile or tenants.

There is a hierarchy of profiles – values in the profiles lower in the hierarchy override those set above, and if a parameter is unset or no web bridge profile is set then it inherits from the next profile up within the hierarchy.

The hierarchy for webBridgeProfiles is:

- Top level (global) profile (**`/system/profiles`**)
- Tenants (**`/tenants/<tenant id>`**)
- webBridges (**`/webBridges/<webbridge id>`**)

10.9.1 Retrieving web bridge profiles

- Enumeration of **`/webBridgeProfiles`** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve web bridge profiles other than the first page in the notional list (3.0 onwards)
limit		
usageFilter	unreferenced |referenced	Supply " usageFilter=unreferenced" in the request to retrieve only those web bridge profiles that are not referenced by global settings or any other object. This is a useful check before deleting the profile. To retrieve just those web bridge profiles that are referenced in at least one place, you can supply " usageFilter=referenced" (3.0 onwards)

Response is structured as a top-level <webBridgeProfiles total=" N" > tag with potentially multiple <webBridgeProfile> elements within it.

Each <webBridgeProfile> tag may include the following element:

Response element	Type/Value	Description/Notes
name	String	The human-readable name associated with this web bridge profile (3.0 onwards). It can have a maximum length of 200 characters.

- GET on **`/webBridgeProfiles/<web bridge profile id>`** gives the following responses:

Response values	Type/Value	Description/Notes
name	String	The human-readable name associated with this web bridge profile (3.0 onwards). It can have a maximum length of 200 characters.
resourceArchive	url	Whether or not web bridges using this web bridge profile should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. (3.0 onwards)
allowPasscodes	true | false	Whether or not web bridges using this web bridge profile should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. (3.0 onwards)
allowSecrets	true | false	Whether or not web bridges using this web bridge profile should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric id and secret. (3.0 onwards)

Response values	Type/Value	Description/Notes
userPortalEnabled	true | false	Whether or not web bridges using this web bridge profile should display the sign in tab on the index page. (3.0 onwards)
allowUnauthenticatedGuests	true | false	Whether to allow guest access from the landing screen on web bridges using this web bridge profile, or only allow visitor access once users have logged into the User Portal. If false, links work only for logged in users. (3.0 onwards)
resolveCoSpaceCallIds	true | false	Whether or not web bridges using this web bridge profile should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards)
resolveLyncConferenceIds	true | false	Whether or not web bridges using this web bridge profile should accept IDs to be resolved to Lync scheduled conference IDs. (3.0 visible but non-functional. 3.1 removed.)
resolveCoSpaceUris	off | domainSugges-tionDisabled | domainSugges-tionEnabled	Whether or not web bridges using this web bridge profile should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards) • when set to 'off' join by URI is disabled • when set to 'domainSuggestionDisabled' join by URI is enabled but the domain of the URI won't be autocompleted or verified on web bridges using this web bridge profile • when set to 'domainSuggestionEnabled' join by URI is enabled and the domain of the URI can be autocompleted and verified on web bridges using this web bridge profile
sessionLimit	Numeric	Maximum number of web sessions allowed per user, ranging from 0 to 5. If not specified, it defaults to zero indicating that there is no session control.
contentResolution	720p|1080p|**unrestricted**	The maximum resolution of the presenter's screen while sharing content in web app.

10.9.2 Creating and modifying web bridge profiles

This **`/webBridgeProfiles`** object is used to implement web bridge profiles. This API node supports the following operations:

- POST to **`/webBridgeProfiles`** to create a new web bridge profile
- PUT on individual profiles with **`/webBridgeProfiles/<web bridge profile id>`**

with the following request parameters:

Parameters	Type/Value	Description/Notes
name	String	The human-readable name associated with this web bridge profile. (3.0 onwards). It can have a maximum length of 200 characters.
resourceArchive	url	The address of any customization archive file that the Meeting Server should use for web bridges using this web bridge profile. (3.0 onwards)
allowPasscodes	**true** | false	Whether or not web bridges using this web bridge profile should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. If this parameter is not supplied in a create (POST) operation, it defaults to " true" . (3.0 onwards)
allowSecrets	**true** | false	Whether or not web bridges using this web bridge profile should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric id and secret. If this parameter is not supplied in a create (POST) operation, it defaults to " true" . (3.0 onwards)
userPortalEnabled	**true** | false	Whether or not web bridges using this web bridge profile should display the sign in tab on the index page. If this parameter is not supplied in a create (POST) operation, it defaults to " true" . (3.0 onwards)

Parameters	Type/Value	Description/Notes
allowUnauthenticatedGuests	**true** | false	Whether to allow guest access from the landing screen on web bridges using this web bridge profile, or only allow visitor access once users have logged into the User Portal. If false, links work only for logged in users. If this parameter is not supplied in a create (POST) operation, it defaults to " true" . (3.0 onwards)
resolveCoSpaceCallIds	**true** | false	Whether or not web bridges using this web bridge profile should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings. If this parameter is not supplied in a create (POST) operation, it defaults to " true" . (3.0 onwards)
resolveLyncConferenceIds	true | **false**	Whether or not web bridges using this web bridge profile should accept IDs to be resolved to Lync scheduled conference IDs. If this parameter is not supplied in a create (POST) operation, it defaults to " false" . (3.0 visible but non-functional. 3.1 removed.)
resolveCoSpaceUris	**off** | domainSuggestionDisabled | domainSuggestionEnabled	Whether or not this web bridge should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards) • when set to 'off' join by URI is disabled • when set to 'domainSuggestionDisabled' join by URI is enabled but the domain of the URI won't be autocompleted or verified on web bridges using this web bridge profile • when set to 'domainSuggestionEnabled' join by URI is enabled and the domain of the URI can be autocompleted and verified on web bridges using this web bridge profile If this parameter is not supplied in a create (POST) operation, it defaults to " off" .

Parameters	Type/Value	Description/Notes
sessionLimit	Numeric	Maximum number of web sessions allowed per user, ranging from 0 to 5. If not specified, it defaults to zero indicating that there is no session control.
contentResolution	720p|1080p|**unrestricted**	The maximum resolution of the presenter's screen while sharing content in web app.

10.9.3 Finding out the web bridge profile currently in effect on a specified web bridge

The API object **`/webBridges/<web bridge id>/effectiveWebBridgeProfile`** allows you to find out the web bridge profile and its associated values that are currently effective on a specified web bridge.

- GET on **`/webBridges/<web bridge id>/effectiveWebBridgeProfile`** gives the following responses:

Response values	Type/Value	Description/Notes
resourceArchive	url	The address of any customization archive file that Meeting Server should use for this web bridge. (3.0 onwards)
allowPasscodes	true | false	Whether or not this web bridge should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. (3.0 onwards)
allowSecrets	true | false	Whether or not this web bridge should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric id and secret. (3.0 onwards)
userPortalEnabled	true | false	Whether or not this web bridge should display the sign in tab on the index page. (3.0 onwards)
allowUnauthenticatedGuests	true | false	Whether to allow guest access from the landing screen on this web bridge, or only allow visitor access once users have logged into the User Portal. If false, links work only for logged in users. (3.0 onwards)
resolveCoSpaceCallIds	true | false	Whether or not this web bridge should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards)

Response values	Type/Value	Description/Notes
resolveLyncConferenceIds	true | false	Whether or not this web bridge should accept IDs to be resolved to Lync scheduled conference IDs. (3.0 visible but non-functional. 3.1 removed.)
resolveCoSpaceUris	off | domainSuggestionDisabled | domainSuggestionEnabled	Whether or not this web bridge should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards) • when set to 'off' join by URI is disabled • when set to 'domainSuggestionDisabled' join by URI is enabled but the domain of the URI won't be autocompleted or verified on this web bridge • when set to 'domainSuggestionEnabled' join by URI is enabled and the domain of the URI can be autocompleted and verified on this web bridge

10.9.4 Finding out the web bridge profile currently in effect at the top level (global) system level

This API object **`/system/profiles/effectiveWebBridgeProfile`** allows you to find out the web bridge profile and its associated values that are currently effective on this system. It supports the following operation:

- GET on **`/system/profiles/effectiveWebBridgeProfile`** gives the following responses:

Response values	Type/Value	Description/Notes
resourceArchive	url	The address of any customization archive file that the Meeting Server uses as default for web bridges on this system. (3.0 onwards)
allowPasscodes	true | false	Whether or not web bridges on this system should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. (3.0 onwards)

Response values	Type/Value	Description/Notes
allowSecrets	true | false	Whether or not web bridges on this system should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric id and secret. (3.0 onwards)
userPortalEnabled	true | false	Whether or not web bridges on this system should display the sign in tab on the index page. (3.0 onwards)
allowUnauthenticatedGuests	true | false	Whether to allow guest access from the landing screen on web bridges for this system, or only allow visitor access once users have logged into the User Portal. If false, links work only for logged in users. (3.0 onwards)
resolveCoSpaceCallIds	true | false	Whether or not web bridges on this system should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards)
resolveLyncConferenceIds	true | false	Whether or not web bridges on this system should accept IDs to be resolved to Lync scheduled conference IDs. (3.0 visible but non-functional. 3.1 removed.)
resolveCoSpaceUris	off | domainSuggestionDisabled | domainSuggestionEnabled	Whether or not web bridges on this system should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards) • when set to 'off' join by URI is disabled • when set to 'domainSuggestionDisabled' join by URI is enabled but the domain of the URI won't be autocompleted or verified on web bridges on this system • when set to 'domainSuggestionEnabled' join by URI is enabled and the domain of the URI can be autocompleted and verified on web bridges on this system

10.9.5 Retrieving the web bridge profile currently in effect on an individual tenant

The API object **`/tenants/<tenant id>/effectiveWebBridgeProfile`** allows you to find out the web bridge profile and its associated values that are currently effective on a specified tenant. It supports the following operation:

- GET on **`/tenants/<tenant id>/effectiveWebBridgeProfile`** gives the following responses:

Response values	Type/Value	Description/Notes
resourceArchive	url	The address of any customization archive file that Meeting Server should use as default for web bridges for this tenant. (3.0 onwards)
allowPasscodes	true | false	Whether or not web bridges for this tenant should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. (3.0 onwards)
allowSecrets	true | false	Whether or not web bridges for this tenant should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric id and secret. (3.0 onwards)
userPortalEnabled	true | false	Whether or not web bridges for this tenant should display the sign in tab on the index page. (3.0 onwards)
allowUnauthenticatedGuests	true | false	Whether to allow guest access from the landing screen on web bridges for this tenant, or only allow visitor access once users have logged into the User Portal. If false, links work only for logged in users. (3.0 onwards)
resolveCoSpaceCallIds	true | false	Whether or not web bridges for this tenant should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings.
resolveLyncConferenceIds	true | false	Whether or not web bridges for this tenant should accept IDs to be resolved to Lync scheduled conference IDs. (3.0 visible but non-functional. 3.1 removed.)

Response values	Type/Value	Description/Notes
resolveCoSpaceUris	off | domainSuggestionDisabled | domainSuggestionEnabled	Whether or not web bridges for this tenant should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. (3.0 onwards) • when set to 'off' join by URI is disabled • when set to 'domainSuggestionDisabled' join by URI is enabled but the domain of the URI won't be autocompleted or verified on web bridges for this tenant • when set to 'domainSuggestionEnabled' join by URI is enabled and the domain of the URI can be autocompleted and verified on web bridges for this tenant

10.9.6 Creating, modifying, and retrieving Web Bridge addresses for a webBridgeProfile

External access on the Configuration > General page in the Web Admin user interface that allowed configuration of the Web Bridge URI and IVR telephone number is removed from 3.1. These configuration fields are now moved to web bridge profiles.

You can also now specify multiple IVR numbers and Web Bridge addresses — up to 32 IVR numbers and up to 32 Web Bridge addresses per Web Bridge profile. These are used when displaying join information, and for generating email invitations.

Note: We strongly recommend that you use a webBridgeProfile at the system level or at the tenant level (if you are using multi-tenancy) for configuring the ivrNumbers and webBridgeAddresses.

This API object supports the following operations:

- POST to **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses`**
- PUT to **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge address id>`**

Parameter	Type/Value	Description/Notes
label	string	Label name that describes this web bridge address. Example: **`USA web app`** (from version 3.1)
address	url	Address to use when rendering email invites. Example: **`https://usa.mycompany.com/`** (from version 3.1)

- Enumeration of **/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve web bridge addresses other than those in the first page in the notional list
limit		

Response is structured as a top-level <webBridgeAddresses total=" N" > tag with potentially multiple <webBridgeAddress> elements within it.

Each <webBridgeAddress> tag may include the following elements:

Response elements	Type/Value	Description/Notes
label	string	Label name that describes this web bridge address. Example: **USA web app** (from version 3.1)

- GET on individual web bridge addresses on webBridgeProfiles with **/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge address id>** gives the following response:

Response elements	Type/Value	Description/Notes
label	string	Label name that describes this web bridge address. Example: **USA web app** (from version 3.1)
address	url	Address used when rendering email invites. Example: **https://usa.mycompany.com/** (from version 3.1)

10.9.7 Creating, modifying, and retrieving IVR numbers for a webBridgeProfile

This API object supports the following operations:

- POST to **/webBridgeProfiles/<web bridge profile id>/ivrNumbers**
- PUT to **/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id>**

Parameter	Type/Value	Description/Notes
label	string	Label name that describes this IVR number. Example: **USA Call-in Number** (from version 3.1)
number	string	IVR number to use when rendering email invites. Example: **888-123123** (from version 3.1)

- Enumeration of **/webBridgeProfiles/<web bridge profile id>/ivrNumbers** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve IVR numbers other than those in the first page in the notional list
limit		

Response is structured as a top-level <ivrNumbers total=" N" > tag with potentially multiple <ivrNumber> elements within it.

Each <ivrNumber> tag may include the following elements:

Response elements	Type/Value	Description/Notes
label	string	Label name that describes this IVR number. Example: **USA Call-in Number** (from version 3.1)

- GET on individual IVR numbers on webBridgeProfiles with **/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id>** gives the following response:

Response elements	Type/Value	Description/Notes
label	string	Label name that describes this IVR number. Example: **USA Call-in Number** (from version 3.1)
number	string	IVR number used when rendering email invites. Example: **888-123123** (from version 3.1)

10.10 Call Bridge Methods

10.10.1 Retrieving Information on Call Bridges

GET method performed on the “/callBridges” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <callBridges total=" N" > tag with potentially multiple <callBridge> elements within it. Each <callBridge> tag may include the following elements:
name	String	See the next section

10.10.2 Setting Up and Modifying a Call Bridge

- Creating: POST method to the " /callBridges" node
- Modifying: PUT to " /callBridges/<*call bridge id*>"

Parameters	Type/Value	Description/Notes
name *	String	The unique name for this configured clustered Call Bridge
address	String	The address at which this Call Bridge in the cluster can be reached
sipDomain	String	The SIP domain to use for establishing peer-to-peer links with this clustered Call Bridge
callBridgeGroup	ID	If specified, associate this Call Bridge with the supplied Call Bridge group (from version 2.1)

10.10.3 Retrieving detailed information about an individual Call Bridge

GET method performed on a “/callBridges/<call bridge id>” node. If the call bridge ID supplied is valid, a “200 OK” response is received, with XML content described in the previous section.

10.11 Call Bridge Group Methods

10.11.1 Retrieving Information on Call Bridge Groups

GET method performed on the “/callBridgeGroups” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <callBridges total=" N" > tag with potentially multiple <callBridgeGroup> elements within it. Each <callBridgeGroup> tag may include the following elements:
name	String	Name of Call Bridge group

10.11.2 Setting Up and Modifying a Call Bridge Group

- Creating: POST method to the " /callBridgeGroups" node
- Modifying: PUT to " /callBridgeGroups/<*call bridge group id*>"

Parameters	Type/Value	Description/Notes
name	String	Optional name of the Call Bridge group
loadBalancingEnabled	true| **false**	Whether or not Call Bridges in this Call Bridge group will attempt to load balance calls within the group. If this parameter is not supplied in a create (POST) operation, it defaults to " false" (from version 2.1)
loadBalanceLyncCalls	true| **false**	Whether or not incoming calls to coSpaces from Lync are load balanced within the Call Bridge Group. If this parameter is not supplied in a create (POST) operation, it defaults to " false" (from version 2.1) Note: Load balancing of calls from/to Lync clients, is not currently supported by Call Bridge Groups.
loadBalanceOutgoingCalls	true| **false**	Whether or not calls from coSpaces should be load balanced within the group. If this parameter is not supplied in a create (POST) operation, it defaults to " false" (from version 2.2)
loadBalanceUserCalls	**true**| false	Whether or not web app calls to coSpaces should be load balanced within the group. If this parameter is not supplied in a create (POST) operation, it defaults to " true" (from version 2.3)
loadBalanceIndirectCalls	true| **false**	Whether or not incoming calls with Record-Route SIP headers should be load balanced within the group. If this parameter is not supplied in a create (POST) operation, it defaults to " false" (from version 2.4) **Note**: This parameter must be set to true if using Exp C as the call control. This is required for load balancing to work on Expressway.

10.11.3 Retrieving detailed information about an individual Call Bridge Group

GET method performed on a "/callBridgeGroups/<call bridge group id>" node. If the Call Bridge group ID supplied is valid, a "200 OK" response is received, with XML content described in the

previous section.

10.12 Call Bridge Cluster Methods

10.12.1 Retrieving the Call Bridge Cluster details

Issue a GET on the /system/configuration/cluster node to retrieve the information below.

Response elements	Type/Value	Description/Notes
uniqueName	String	The name by which this call bridge is known within the call bridge cluster; this should match the " name" value for its entry in the /callBridges table
peerLinkBitRate	Number	The maximum media bit rate specified to use for call distribution connections between call bridges
participantLimit	Number	If supplied, the maximum number of participants allowed to be active on this Call Bridge; when this limit is reached, new incoming SIP calls will be rejected
loadLimit	Number	If supplied, the maximum number of load units to be used on this Call Bridge (from version 2.1)
newConferenceLoadLimitBasisPoints	Number	Basis points (1 in 10,000) of the load limit at which incoming calls to non-active conferences will be disfavored, ranges from 0 to 10000, defaults to 5000 (50% load). Value is scaled relative to load limit. (From version 2.1)
existingConferenceLoadLimitBasisPoints	Number	Basis points of load limit at which incoming calls to this Call Bridge will be rejected, ranges from 0 to 10000, defaults to 8000 (from version 2.1)
maxPeerVideoStreams	Number	The maximum number of streams sent over a call distribution connection between Call Bridges, defaults to 4 if not supplied. (from version 2.3.3)

10.12.2 Setting Up and Modifying the Call Bridge Cluster

Issue a PUT on the /system/configuration/cluster node

Response elements	Type/Value	Description/Notes
uniqueName	String	The name by which this call bridge is known within the call bridge cluster; this should match the " name" value for its entry in the /callBridges table

Response elements	Type/Value	Description/Notes
peerLinkBitRate	Number	If supplied. the maximum media bit rate to use for call distribution connections between call bridges
participantLimit	Number	If supplied, the maximum number of participants allowed to be active on this Call Bridge; when this limit is reached, new incoming SIP calls will be rejected
loadLimit	Number	If supplied, the maximum number of load units to be used on this Call Bridge (from version 2.1)
newConferenceLoadLimitBasisPoints	Number	Basis points (1 in 10,000) of the load limit at which incoming calls to non-active conferences will be disfavored, ranges from 0 to 10000, defaults to 5000 (50% load). Value is scaled relative to load limit. (From version 2.1).
existingConferenceLoadLimitBasisPoints	Number	Basis points of the load limit at which incoming calls to non-active conferences will be rejected, ranges from 0 to 10000, defaults to 5000 (from version 2.1).

10.13 System Load Method

GET method performed on the "/system/load" node.

Response elements	Type/Value	Description/Notes
mediaProcessingLoad	Number	Current media processing load on the Call Bridge

10.14 Compatibility Profile Methods

10.14.1 Retrieving compatibility profile operations

GET method performed on the "/compatibilityProfiles" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve compatibility profiles other than those in the first " page" of the notional list.

Parameters	Type/Value	Description/Notes
usageFilter	unreferenced| referenced	Supply " usageFilter=unreferenced" in the request to retrieve only those compatibility profiles that are not referenced by another object. This is a useful check before deleting the profile. To retrieve just those compatibility profiles which are referenced in at least one place, you can supply " usageFilter=referenced"

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <compatibilityProfiles total=" N" > tag with potentially multiple <compatibilityProfile> elements within it. Each <compatibilityProfile> tag may include these elements:
sipUdt	true| false	Indicates whether use of UDT is allowed within SIP calls. (From version 2.1)
sipMultistream	true| false	Indicates whether use of Cisco multistream protocols is allowed within SIP calls. (From version 2.2)
sipMediaPayloadTypeMode	auto| broadsoft	Indicates whether the default codec media payload types are used, or a special variant. (From version 2.2)
chromeWebRtcVideoCodec	auto| avoidH264	Indicates which codec Chrome will use for WebRTC calls. (From version 2.3)
h264CHPMode	**auto**| basic	Indicates which parts of the H.264 Constrained High Profile (CHP) are used. (From version 2.4)
safariWebRtcH264interopMode	auto | none	Indicates which H.264 parameters are used by Safari browsers for WebRTC calls.

10.14.2 Setting up and modifying compatibility profile operations

- Creating: POST method to the " /compatibilityProfiles" node
- Modifying: PUT to " /compatibilityProfiles/<compatibility profile id>"

Parameters	Type/Value	Description/Notes
sipUdt	true| false	Controls whether use of UDT is allowed within SIP calls. Active Control uses UDT transport for certain features, for example sending roster lists to endpoints, allowing users to disconnect other participants while in call, and inter-deployment participant lists. (From version 2.1) true - UDT is allowed within SIP calls false - UDT is not allowed within SIP calls
sipMultistream	**true**| false	Controls whether use of Cisco multistream protocols is allowed within SIP calls. The dual video feature for Cisco dual endpoints uses this protocol. If this is disabled, then no calls will be able to use dual screen video. (From version 2.2.3) true - Cisco multistream signalling is allowed within SIP calls (If <unset>, sip multistream behavior defaults to **`true`**) false - Cisco multistream signalling is not allowed within SIP calls
sipMediaPayloadTypeMode	auto| broadsoft	Controls whether the default codec media payload types are used, or a special variant. (From version 2.2) auto - the default mode, where the normal media payload type values are used broadsoft - a special exception mode, where the H.264 video codec is advertised with payload type 109.
chromeWebRtcVideoCodec	auto| avoidH264	Controls which codec Chrome will use for WebRTC calls: auto - allows all codecs, results in Chrome using h264 for WebRTC (default behavior) avoidH264 - results in Chrome using VP8 instead (From version 2.3)
chromeWebRtcH264interopMode	**auto**|none	Controls H264 parameters used by Chrome for WebRTC calls: • auto - default behavior, disables High Profile and forces Chrome to advertise Base Profile level 5.0 in its SDP offer • none - legacy behavior No changes to SDPs. (From version 2.9)

Parameters	Type/Value	Description/Notes
h264CHPMode	**auto**| basic	Controls which parts of the H.264 Constrained High Profile (CHP) are used. auto - default behavior, based on endpoint identification, appropriate parts are used basic - only a minimal subset of parts are used (From version 2.4)
passthroughMode	**enabled**|disabled	Controls whether H264 passthrough feature is allowed • enabled – allows video to avoid being transcoded when possible • disabled – always transcode video (From version 3.1)
passthroughDualCamera	**enabled**|disabled	• enabled – Does not transcode video for dual camera endpoint and peer link. • disabled – Transcodes video for dual camera endpoint and peer link. **Note:** Disable this parameter only if **`passthroughMode`** is enabled. This parameter is void if **`passthroughMode`** is disabled. It is recommended to set the minimum bit rate of SIP RX/TX and peer link to 4 Mbps before placing the calls in conferences with dual camera end points. (From version 3.7)
sipH224	true| false	Controls whether use of H.224 is allowed within SIP calls - this protocol is used for far-end camera control support. true - H.224 is enabled for all SIP calls false - H.224 is disabled for all SIP calls (From version 2.8)

Parameters	Type/Value	Description/Notes
distributionLinkMediaTraversal	enabled|disabled	Controls whether media traversal (ICE / STUN) should be used for distribution links between clustered Meeting Server devices. enabled - media traversal should be used for distribution links disabled - media traversal should not be used for distribution links (From version 2.8)
safariWebRtcH264interopMode	auto | none	Controls H.264 parameters used by Safari browsers for WebRTC calls. auto : SDPs sent to WebRTC clients running on Safari will disable H.264 High Profile, and advertise Base Profile Level 5. This is the default value. none : No change to SDPs. (from version 3.2)

10.15 Lobby Profile Methods

10.15.1 Retrieving lobby profiles operations

GET method performed on the " /lobbyProfiles" node

Parameter	Type/Value	Description
name (optional)	String	Optional name label to help identify this profile (maximum size 200 bytes)
joinMethod (optional)	unset | present-ationOnly | **audioVideo**	One of **presentationOnly** or **audioVideo**. This determines if a webapp participant will join with presentation only or as an audio video participant. The usual rules for the hierarchy of lobbyProfiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to the existing method of **audioVideo**. For guest users, the parameter follows the hierarchy: **accessMethod > coSpaces > tenant > system/profiles**. For users who are trying to join a space that they are a member of, using the **Join** button, the parameter follows the hierarchy: **coSpaceUsers > accessMethod > coSpaces > tenant > system/profiles**.

Response elements	Type/Value	Description/Notes
joinMethod	audioVideo| presentationOnly	Sets the default joining method to audio and video or presentation only modes.

10.15.2 Setting up and modifying lobby profiles operations

- Creating: POST method to the " /lobbyProfiles" node
- Modifying: PUT to " /lobbyProfiles/<lobby Profile id>"

Parameter	Type/Value	Description
name (optional)	String	Optional name label to help identify this profile (maximum size 200 bytes)
joinMethod (optional)	unset | present-ationOnly | **audioVideo**	One of **presentationOnly** or **audioVideo**. This determines if a webapp participant will join with presentation only or as an audio video participant. The usual rules for the hierarchy of lobbyProfiles apply to this parameter. If unset at all levels of the hierarchy, it defaults to the existing method of **audioVideo**. For guest users, the parameter follows the hierarchy: **accessMethod > coSpaces > tenant > system/profiles**. For users who are trying to join a space that they are a member of, using the **Join** button, the parameter follows the hierarchy: **coSpaceUsers > accessMethod > coSpaces > tenant > system/profiles**.

10.16 System Diagnostics Methods

10.16.1 Retrieving system diagnostics

Issue a GET on the new /system/diagnostics node to retrieve the information below.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list.
coSpaceFilter	ID	If supplied, this filter restricts results returned to those diagnostics that correspond to the specified coSpace
callCorrelatorFilter	ID	If supplied, this filter restricts results returned to those diagnostics that correspond to the specified callCorrelator

Response elements	Type/Value	Description/Notes
label	String	Text description associated with the specified diagnostics log

10.16.2 Retrieving an individual system diagnostic

Issue a GET on the /system/diagnostics/<diagnostics ID> node

Response elements	Type/Value	Description/Notes
label	String	Text description associated with the specific diagnostics log
coSpace	ID	If the diagnostics log is associated with a specific coSpace, this parameter holds the ID of that coSpace
callCorrelator	ID	An ID that is the same across all distributed instances of the active call – this value will be the same for other Call Bridge peers' diagnostics file for the same call
timestamp	String	The time at which the diagnostics log was generated
contentsSize	Number	The size of the diagnostics data for this log entry

10.16.3 Retrieving the content of an individual system diagnostic

Issue a GET method on the /system/diagnostics/<diagnostics id>/contents node to retrieve the data stored in the system diagnostic.

10.17 System timedlogging methods

10.17.1 Activating a logging subsystem

Version 3.2 introduces a new API node, **`/system/timedLogging`** to support the following operations:

- PUT to **`/system/timedLogging`**
- GET on **`/system/timedLogging`**

It supports the parameters detailed in the table below. Each parameter can be assigned an integer value, corresponding to the duration of seconds for which that logging subsystem will be activated.

Setting a parameter to 0 or to nothing will deactivate a logging subsystem. For example, a PUT to **`system/timedLogging`** with **`sip=60`** would activate detailed logging for SIP for 60 seconds. A PUT to **`system/timedLogging`** with **`sip=0`** before those 60 seconds have elapsed would deactivate the logging again. You can supply multiple parameters at the same time, for example: **`sip=600&tip=600`** to enable both SIP and TIP logging for the next 10 minutes.

The following parameters are available for this object:

Parameter	Type/ Value	Description/ Notes
activeControl	numeric	time remaining (in seconds) for which detailed Active Control logging should be enabled
activeSpeaker	numeric	time remaining (in seconds) for which detailed active speaker logging should be enabled
api	numeric	time remaining (in seconds) for which detailed API logging should be enabled
bfcp	numeric	time remaining (in seconds) for which detailed BFCP logging should be enabled
cameraControl	numeric	time remaining (in seconds) for which detailed camera control logging is enabled (0 if not enabled)
dns	numeric	time remaining (in seconds) for which detailed DNS logging should be enabled
events	numeric	time remaining (in seconds) for which detailed Events logging should be enabled
ice	numeric	time remaining (in seconds) for which detailed ICE logging should be enabled
sip	numeric	time remaining (in seconds) for which detailed SIP logging should be enabled
tip	numeric	time remaining (in seconds) for which detailed TIP logging should be enabled
webBridge	numeric	time remaining (in seconds) for which detailed web bridge logging should be enabled

11 LDAP Methods

Objects in the hierarchy that reside in the "/ldapMappings", "/ldapServers" and "/ldapSources" nodes in the object tree relate to the Meeting Server's interaction with one or more LDAP servers (for instance, Active Directory) which are used to import user accounts to the Meeting Server.

- One or more LDAP servers should be configured, with each one having associated username and password information for the Meeting Server to use to connect to it for the purpose of retrieving user account information from it
- One or more LDAP mappings are also required, which define the form of the user account names which will be added to the system when users are imported from configured LDAP servers
- A set of LDAP sources then need to be configured, which tie together configured LDAP servers and LDAP mappings, along with parameters of its own, which correspond to the actual import of a set of users.
  An LDAP source takes an LDAP server / LDAP mapping combination and imports a filtered set of users from that LDAP server. This filter is determined by the LDAP source's " baseDn" (the node of the LDAP server's tree under which the users can be found) and a filter to ensure that user accounts are only created for LDAP objects that match a specific pattern

The API LDAP methods allow multiple additional sets of "Active Directory Configuration" as per the Web Admin Interface **Configuration > Active Directory** page. On this page, the Active Directory Server Settings section corresponds to an API-configured LDAP server, the Import Settings to an LDAP source, and the Field Mapping Expressions to an LDAP mapping.

Note: The LDAP server credentials are used to read the following fields:

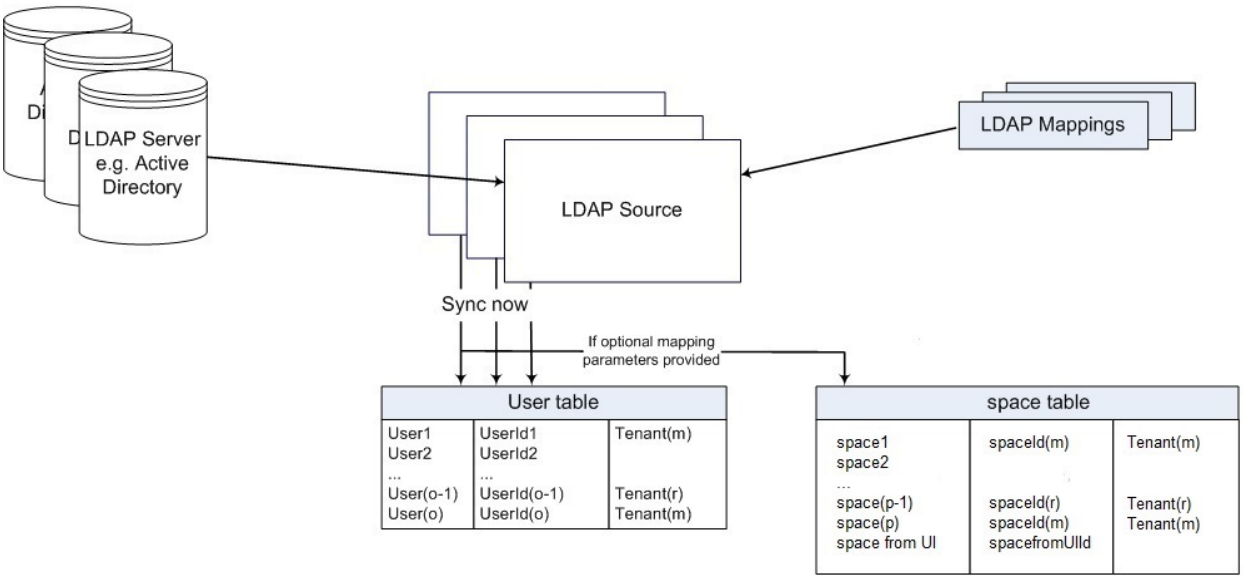
mail	telephoneNumber
objectGUID	mobile
entryUUID	sn
nsuniqueid	givenName

For security reasons you may want to restrict the fields and permissions available using those credentials.

Note: When configuring the Meeting Server for LDAP/AD sync, the fields which accept LDAP/AD attributes require that attributes are entered in their case-sensitive format. For example, if the username mapping uses the attribute userPrincipalName then $userPrincipalName$ can result in successful sync but $UserPrincipalName$ will result in sync failure. You are advised to check that each LDAP attribute is entered in the correct case.

11.1 LDAP Server Methods

Figure 3: Outline LDAP process



11.1.1 Retrieving Information on LDAP Servers

GET method performed on the "/ldapServers" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those LDAP servers that match the filter
offset	Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
limit	Number	

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <ldapServers total="N"> tag with potentially multiple "<ldapServer>" elements within it. "<ldapServer>" elements returned follow the general form on the left.

Response elements	Type/Value	Description/Notes
ldapServer id	ID	
address	String	
portNumber	Number	
secure	true|false	

11.1.2 Adding and modifying an LDAP Server

- Create: POST method performed on the "/ldapServers" node. If the LDAP server is configured on the system successfully, its ID is returned in the "Location" field of the response header
- Modifying an LDAP server is a PUT method on a "/ldapServers/<ldapServer id>" node

Parameters	Type/Value	Description/Notes
address *	String	address of the LDAP server to connect to
name	String	associated name (from version 2.9 onwards)
portNumber *	Number	TCP or TLS port number to connect to on the remote LDAP server
username	String	username to use when retrieving information from the LDAP server
password	String	password of the account associated with username
secure *	true|false	whether to make a secure connection to the LDAP server. If "true" then TLS will be used; if "false", TCP will be used.
usePagedResults	true|false	whether to use the LDAP paged results control in search operations during LDAP sync; if not set the paged results control will be used. Oracle Internet Directory requires this parameter to be set to "false" (from version 2.1).

11.1.3 Retrieving detailed information about an individual LDAP Server

GET method performed on a "/ldapServers/<ldapServer ID>" node. If the ldapServer ID supplied is valid, a "200 OK" response is received, with XML content:

Response elements	Type/Value	Description/Notes
ldapServer id	ID	
address	String	
portNumber	Number	Domain name
username	String	directoryUser
secure	true|false	

11.2 LDAP Mapping Methods

11.2.1 Adding and modifying an LDAP Mapping

- Creating: POST method to the "/ldapMappings" If the LDAP mapping is configured on the system successfully, its ID is returned in the "Location" field of the response header
- Modifying: PUT method on a "/ldapMappings/<ldapMapping id>" node

Parameters	Type/Value	Description/Notes
jidMapping	String	The template for generating user JIDs from the associated LDAP server's entries, for instance $sAMAccountName$@example.com. Note: user JIDs generated by jidMapping are also used as URIs so must be unique and not the same as any URI or call ID.
nameMapping	String	The template for generating user names from the associated LDAP server's entries; for instance "cn" to use the com-mon name.
cdrTagMapping	String	The template for generating a users' cdrTag value. Can be set either to a fixed value or be constructed from other LDAP fields for that user. The user's cdrTag is used in callLegStart CDRs. See the Cisco Meeting Server CDR Reference for details.
authenticationIdMapping	String	The template for generating authentication IDs from the associated LDAP server" s entries, for instance " $userPrincipalName$" .

Parameters	Type/Value	Description/Notes
coSpaceUriMapping	String	If these parameters are supplied, they ensure that each user account generated by this LDAP mapping has an associated personal coSpace. The user is automatically added as a member of the coSpace, with permissions defined above For that coSpace to be set up as required, these parameters provide the template for setting the coSpaces' URI, displayed name and configured Call ID. For example, setting coSpaceNameMapping to "cn personal coSpace" ensures that each user's coSpace is labelled with their name followed by "personal coSpace". Note that the generated coSpace will have its own cdrTag – and it will be the same as the user's cdrTag and cannot be changed other than by changing the cdrTagMapping above and re-syncing. (The coSpace's cdrTag is used in the callStart CDR. See the Cisco Meeting Server CDR Reference for details.) Note that the normal uniqueness rules apply to the URI and Call IDs of coSpaces set up in this way: it is not valid to have the same URI or Call ID for more than one coSpace set up by a given LDAP mapping, nor is it valid for such a coSpace URI or Call ID to be the same as one currently in use elsewhere on the Meeting Server. Note: user JIDs generated by jidMapping are also used as URIs so must be unique and not the same as any URI or call ID.
coSpaceSecondaryUriMapping	String	
coSpaceNameMapping	String	
coSpaceCallIdMapping	String	

11.2.2 Secondary LDAP Mapping parameter

Per LDAP mapping, there is a new optional coSpaceSecondaryUriMapping parameter so that the coSpaces that are created automatically have a secondary URI.

- When creating an LDAP mapping (see the previous section) or modifying the configuration of an existing LDAP mapping you can supply a " coSpaceSecondaryUriMapping" parameter
- When retrieving information on an individual LDAP mapping (a GET method on a " /ldapMappings/<LDAP mapping ID>" node) the coSpaceSecondaryUriMapping value will be returned if it is defined for that LDAP mapping

11.2.3 Retrieving information on LDAP Mappings

GET method performed on the "/ldapMappings" node.

Parameters	Type/Value	Description/Notes
filter	String	Supply filter=<string> in the URI to return just those LDAP mappings that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <ldapMappings total="N"> tag with potentially multiple "<ldapMapping>" elements within it. "<ldapMapping>" elements returned follow the general form on the left.
ldapMapping id jidMapping nameMapping	ID String String	

11.2.4 Retrieving detailed Information about an individual LDAP Mapping

GET method performed on a "/ldapMappings/<ldapMapping ID>" node. If the ldapMapping ID supplied is valid, a "200 OK" response is received, with XML content described in Section 11.2.1

11.3 LDAP Source Methods

11.3.1 Retrieving Information on LDAP Sources

GET method performed on the "/ldapSources" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
tenantFilter	ID	Supply tenantFilter to return only those LDAP sources associated with the specified tenant.

Response is structured as a top-level <ldapSources total=" N" > tag with potentially multiple <ldapSource> elements within it. Each <ldapSource> tag may include the following elements:

Response elements	Type/Value	Description/Notes
name	string	An optional name label for the LDAP source (from version 3.1)

Response elements	Type/Value	Description/Notes
server *	ID	The ID of a previously-configured LDAP server (see above)
mapping *	ID	The ID of a previously-configured LDAP mapping (see above)
baseDn *	String	The distinguished name of the node in the LDAP server's tree from which users should be imported, for instance "cn=Users,dc=<companyname>,dc=com"
filter	String	An LDAP filter string that records must satisfy in order to be imported as users, for instance "(objectClass=person)"
tenant	ID	If supplied, the ID for the tenant to which the LDAP source should be associated. Users imported with this LDAP source will be associated with that tenant
userProfile	ID	If supplied, this is the ID of the user profile to associate with users imported via this LDAP source. This parameter is present from version 2.0 onwards.
nonMemberAccess	**true**| false	This parameter was introduced in version 2.4. It pre-configures newly created spaces to allow or disallow non-member access. When spaces are auto-generated via an LDAP sync, they are all created without a passcode. This parameter enables a company to enforce passcode protection for non-member access to all user spaces. true - no passcode is required to access the space and non-members are able to access the created spaces. This is the default setting and matches behavior before this parameter was introduced in version 2.4. false - ensures the member must configure non-member access and set a passcode as part of the LDAP sync. Note: non-member access to spaces created before version 2.4 are unaffected by any LDAP syncs.

11.3.2 Adding and modifying an LDAP Source

- Creating: POST method to the "/ldapSources" node. If the LDAP source is configured on the system successfully, its ID is returned in the "Location" field of the response header
- Modifying: PUT method on a "/ldapSources/<ldapSource id>" node

Parameters	Type/Value	Description/Notes
name	string	An optional name label for the LDAP source (from version 3.1)
server *	ID	The ID of a previously-configured LDAP server (see above)
mapping *	ID	The ID of a previously-configured LDAP mapping (see above)

Parameters	Type/Value	Description/Notes
baseDn *	String	The distinguished name of the node in the LDAP server's tree from which users should be imported, for instance "cn=Users,dc=<companyname>,dc=com"
filter	String	An LDAP filter string that records must satisfy in order to be imported as users, for instance "(objectClass=person)"
tenant	ID	If supplied, the ID for the tenant to which the LDAP source should be associated. Users imported with this LDAP source will be associated with that tenant
userProfile	ID	If supplied, this is the ID of the user profile to associate with users imported via this LDAP source. This parameter is present from version 2.0 onwards.
nonMemberAccess	**true**| false	This parameter was introduced in version 2.4. It pre-configures newly created spaces to allow or disallow non-member access. When spaces are auto-generated via an LDAP sync, they are all created without a passcode. This parameter enables a company to enforce passcode protection for non-member access to all user spaces. true - no passcode is required to access the space and non-members are able to access the created spaces. This is the default setting and matches behavior before this parameter was introduced in version 2.4. false - ensures the member must configure non-member access and set a passcode as part of the LDAP sync. Note: non-member access to spaces created before version 2.4 are unaffected by any LDAP syncs.

11.3.3 Retrieving detailed information on a LDAP Source

GET method performed on a "/ldapSources/<ldapSource ID>" node. If the ldapSource ID supplied is valid, a "200 OK" response is received, with XML content as per LDAP source creation described above.

11.4 LDAP Sync Methods

API support for LDAP synchronization comprises the ability to:

- Trigger a new sync via the API
- Monitor pending and in-progress LDAP syncs

There is a top-level /ldapSyncs node in the object tree, and associated GET, DELETE, POST methods to use on objects underneath it.

11.4.1 Retrieving scheduled LDAP sync methods

GET method on the " /ldapSyncs" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).

Response elements	Type/Value	Description/Notes
		Returns results in the form " <ldapSyncs total=N> ... </ldapSyncs>" , with N being the total number of pending and in-progress LDAP sync methods. Within the encompassing " <ldapSyncs>" are one or more " <ldapSync id=ID>" nodes; each of the form on the left, where: state is either " inProgress" (happening now) , " pending" (yet to be started), completed or failed
ldapSyncid	ID	
state	inProgress| pending| complete| failed	The current status of this LDAP sync operation: inProgress - this LDAP sync operation is happening now pending - this LDAP sync operation has yet to start complete - this LDAP sync operation has completed successfully failed - this LDAP sync operation has failed
failureReason	tenantDoesNotExist| ldapSourceDoes NotExist| clashOc- curred| ldapError	
numUsersImported	Number	The number of users imported so far for an in-progress LDAP sync
numLdapSourcesComplete	Number	The number of LDAP sources for which the sync method has been completed for an in-progress LDAP sync of multiple LDAP sources. However if the first LDAP source synchronization is still in progress so that numLdapSourcesComplete=0, then the parameter is omitted
creationTime	string	Timestamp of when the sync object was created in UTC. Formatted as per RFC 3339, " 2014-02-11T12:10:47Z" (from version 3.1)
startTime	string	Timestamp of when the sync operation started in UTC. (from version 3.1)
endTime	string	Timestamp of when the sync operation ended (in success or failure) in UTC. (from version 3.1)

11.4.2 Initiating a new LDAP sync

POST method on the " /ldapSyncs" node. If neither parameter in the following table is included, the sync is equivalent to the **Sync now** button on the Web Admin Interface **Configuration > Active Directory** page.

Parameters	Type/Value	Description/Notes
tenant	Text	If supplied the sync will be restricted to that tenant
ldapSource	ID	If supplied the sync will be restricted to that LDAP Source
removeWhenFinished	**true**|false	▪ true – this LDAP sync will, when it has finished (either successfully or with an error), be removed from the system's tracked list, and so can no longer be queried ▪ false – this LDAP sync will persist in the system's tracked list when it has finished, at which point its success or failure status may then be queried. To avoid it remaining permanently on the system's tracked list, it needs to be deleted explicitly. if this parameter is not supplied in a create (POST) operation, it defaults to " true" .

If a new LDAP synchronization happens successfully, the response will include a " Location" of the form " /api/v1/ldapSyncs/<LDAPsync ID>" .

11.4.3 Cancelling a scheduled LDAP sync

DELETE method on a " /ldapSyncs/<LDAP sync ID>" node. This method cancels a scheduled LDAP sync. This method will fail if the sync method has already started (or started and completed).

11.4.4 Retrieving information on a single LDAP sync method

GET method on an " /ldapSyncs/<LDAP sync ID>" node.

If the LDAP sync ID provided is valid, the result is of the form <ldapSync id=LDAP sync ID> ... </ldapSync> with values as described above plus the following if an LDAP sync operation has failed with a failureReason of clashOccurred:

Response elements	Type/Value	Description/Notes
clashingUserJid		If present, these fields include the ID(s) that clashed
clashingUri		
clashingCallId		

11.5 External Directory Search Locations

Via the API you can add to the Call Bridge, additional directory search locations to be consulted when users of web apps perform searches. External directory search locations can be added on a per-tenant level. Results from these locations will be added to the " normal" results (e.g. those from our LDAP-sourced user lists) and presented in the web app.

11.5.1 Retrieving Information on external directory search locations

GET method performed on the “/directorySearchLocations” node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
tenantFilter	ID	Supply tenantFilter to return only those external directory search locations associated with the specified tenant.

Response elements	Type/Value	Description/Notes
		<directorySearchLocations total=" N" > tag with potentially multiple <directorySearchLocation> elements within it. . <directorySearchLocation> elements returned follow the general form on the left.
ldapServer	ID	All as per external directory search location creation described below
tenant	ID	
baseDn	String	
filterFormat	String	
label	String	
priority	Number	

11.5.2 Adding and modifying external directory search locations

- Creating: POST method to the “/directorySearchLocations” node. If the LDAP source is configured on the system successfully, its ID is returned in the “Location” field of the response header
- Modifying: PUT method on a “/directorySearchLocations/<directory search location id>” node

Parameters	Type/Value	Description/Notes
ldapServer *	ID	The ID of a previously-configured LDAP server (see above)
tenant	ID	if supplied, the tenant to which this external directory applies; entries from the remote directory will only be supplied to users associated with this tenant
baseDn	String	The distinguished name of the node in the LDAP server's tree within which to search
filterFormat	String	The LDAP filter used to select directory search results; $1 should be used to represent the user-supplied search string
label	String	The human-readable name that should be associated with search results from this directory when displayed by requesting clients.
priority	Number	Controls the order in which directorySearchLocations should be used in searching; entries with higher priorities will be used first
displayName email sip		These fields specify the name of a field from LDAP to be used when populating the contents of the search result. For example, displayName might be set to " cn" to use the canonical name.

11.5.3 Retrieving detailed information on external directory search locations

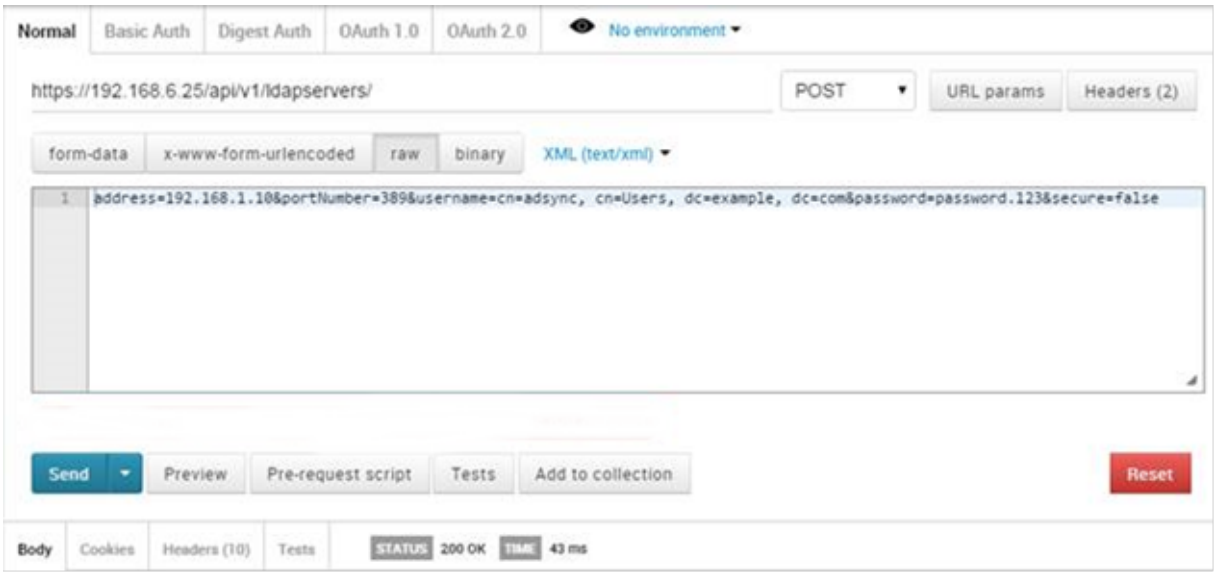
GET method performed on a “/directorySearchLocations/<directory search location id>” node. If the directory search location ID supplied is valid, a “200 OK” response is received, with XML content as per directory search location creation described above.

11.5.4 Example of adding external directory search locations

This section provides an example of adding additional directory search locations that the Call Bridge will consult when users of web apps perform searches.

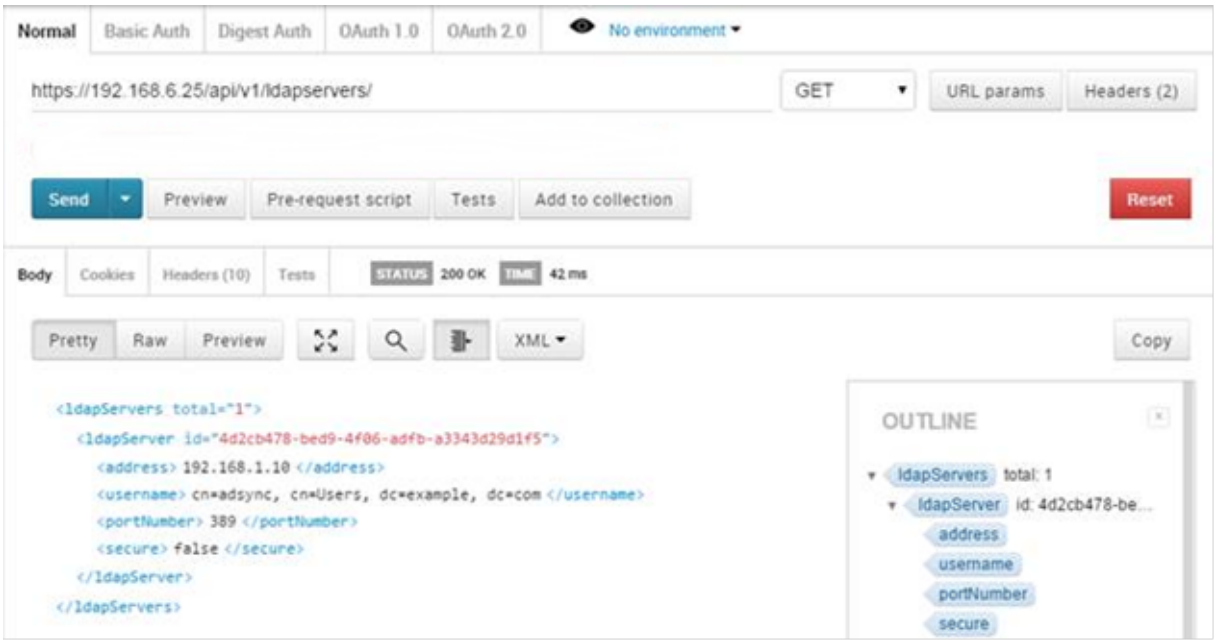
Follow these steps:

1. Use an app such as Chrome Postman to login to the API of the Meeting Server.
2. Create an LDAP server entry in the Meeting Server. The figure below shows an ldapserver entry being POSTed to the Meeting Server at URL 192.168.6.25. The entry is for the LDAP server at URL 192.168.1.10, authorization information is provided.

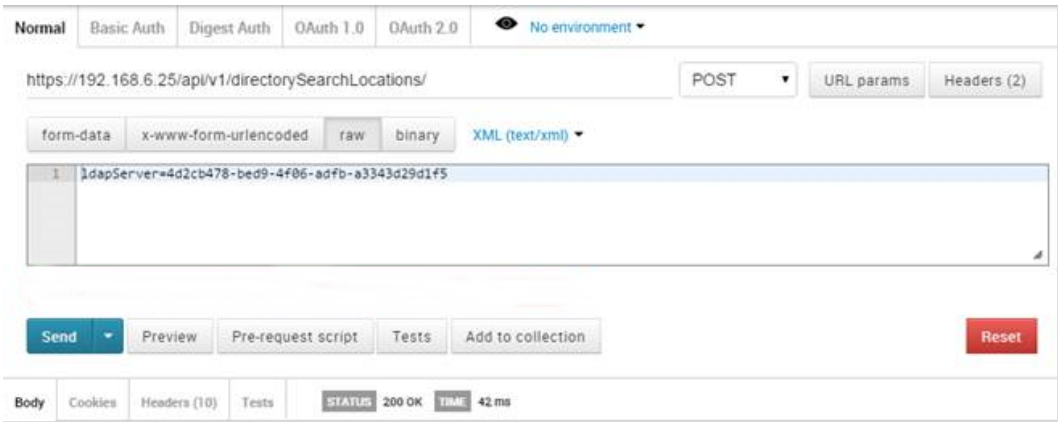


Step 2 provides the necessary information to authorize Call Bridge to access the LDAP server.

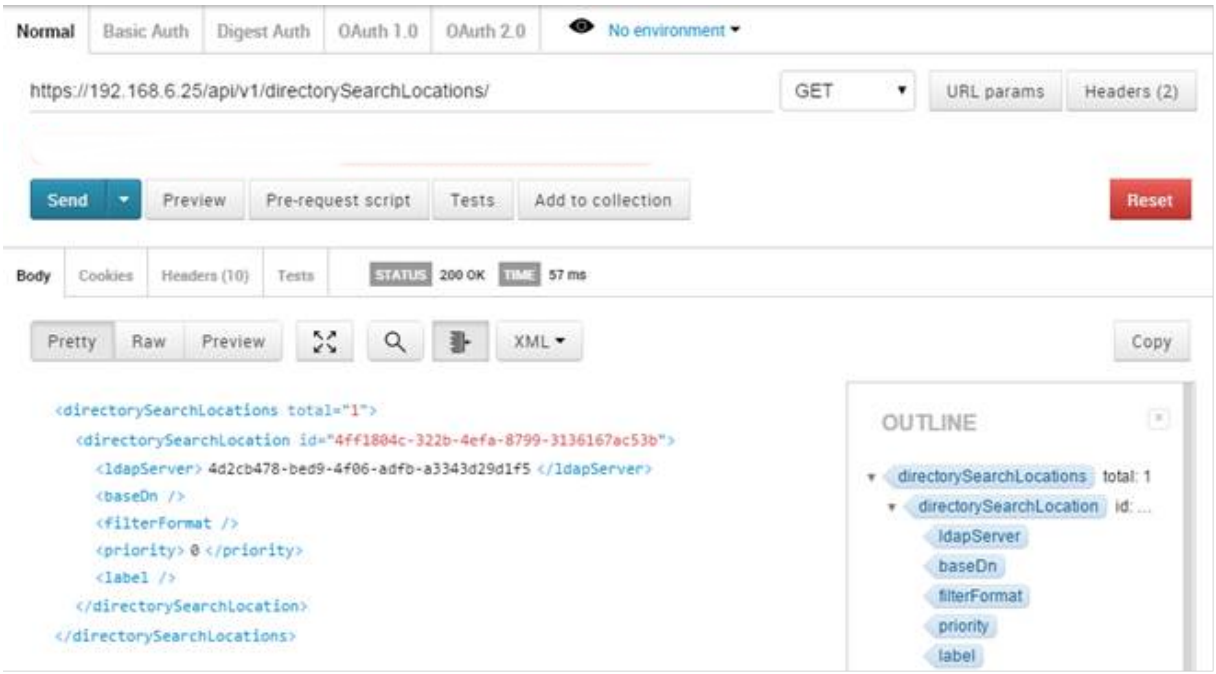
3. Use GET to obtain the ID of the LDAP server entry created in step 2.



4. Create a DirectorySearchLocation by POSTing the LDAP server ID from step 3. The DirectorySearchLocation settings define the behavior of searching the directory. Ensure you set the data format to raw.



5. Use GET to obtain the ID of the Directory Search Location created in step 4.

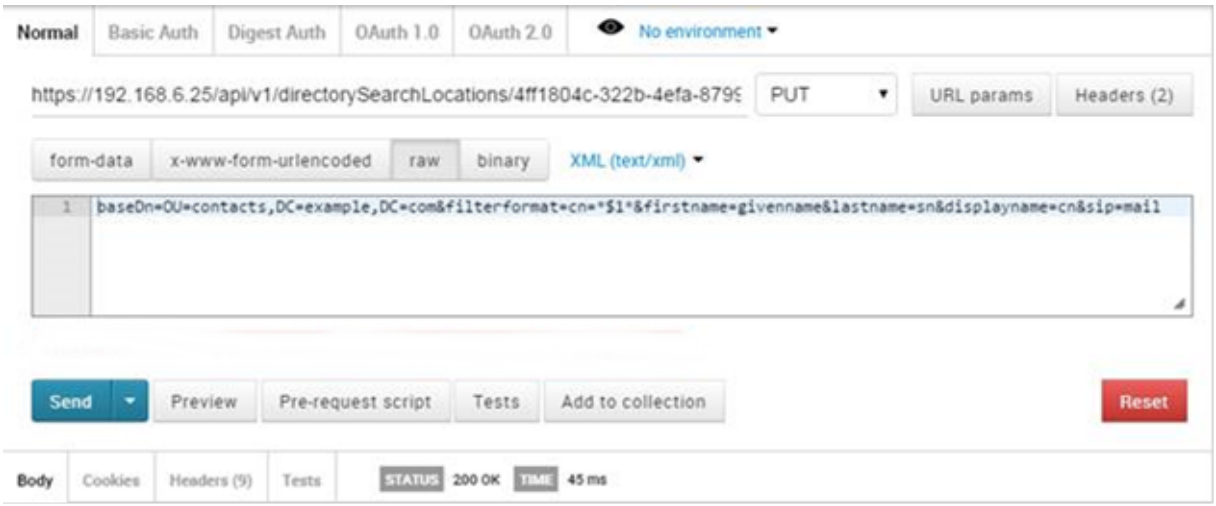


6. Use PUT to edit the DirectorySearchLocation. Add the DirectorySearchLocation ID in the PUT URL and provide the detailed information about baseDN and filterFormat etc. Make sure that you set the data format to raw. For example:

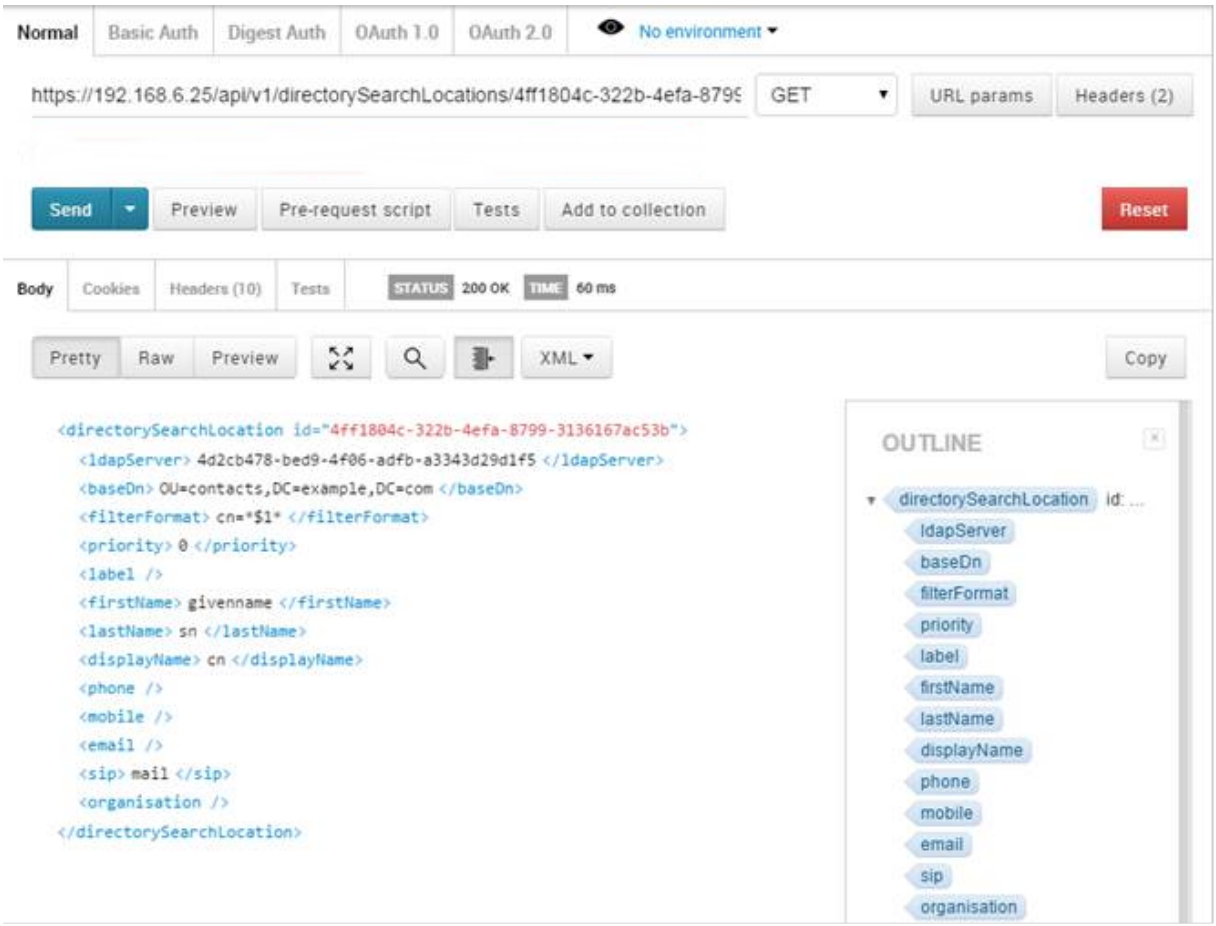
   PUT:

   https://192.168.6.25/api/v1/DirectorySearchLocations/4ff1804c-322b-4efa-8799-3136167ac53b

   baseDn=OU=contacts,DC=example,DC=com&filterformat=cn=*$1*&firstname=givenname&lastname=sn&displayname=cn&sip=mail



7. Check that the configuration of the DirectorySearchLocation is as you expected.



12 Multi-tenancy

The Meeting Server supports multi-tenancy; this refers to sub-dividing its capacity into a set of "islands" where each island has all of the functionality of the unit as a whole, but has no access to the resources (for instance users, coSpaces, or active calls) of other tenants.

There are two main implications of multi-tenancy on the API:

- The API allows tenants to be created, modified and removed, and is the primary means by which tenants are managed
- The API can return results to a specific tenant.
  In multi-tenancy mode, typically each coSpace, call and user is keyed to ("owned by") a tenant; when retrieving information on a specific user, coSpace or call object, the API includes which tenant owns that object, and enumeration-based retrievals include the tenant information for each object in the returned list.
  Equally GET methods can be filtered to only include information for a specific tenant

The majority of the API methods detailed earlier in this document also work in multi-tenancy mode. In most cases, this equates to supplying a tenant ID in creation (POST) methods via the form parameters, such that the system knows which tenant a new object is to be associated with. For instance, creating a coSpace when not in multi-tenant mode would involve a POST method to "/api/v1/coSpaces" with parameters such as "name" as form parameters. To create a coSpace for a specific tenant, the POST would again be to "/api/v1/coSpaces", but additionally include a tenant=<tenant id>" in the form parameters, where "<tenant id>" would have been learnt either as the result of a previous tenant creation or via an earlier enumeration.

In any initial POST method, an absent "tenant" parameter or a zero-length value are treated as equivalent. The effect is that there is that the object is not associated with any tenant.

According to the rules above, <tenant id> values are valid in creation and modification (POST and PUT) methods for the following objects (see Figure 4):

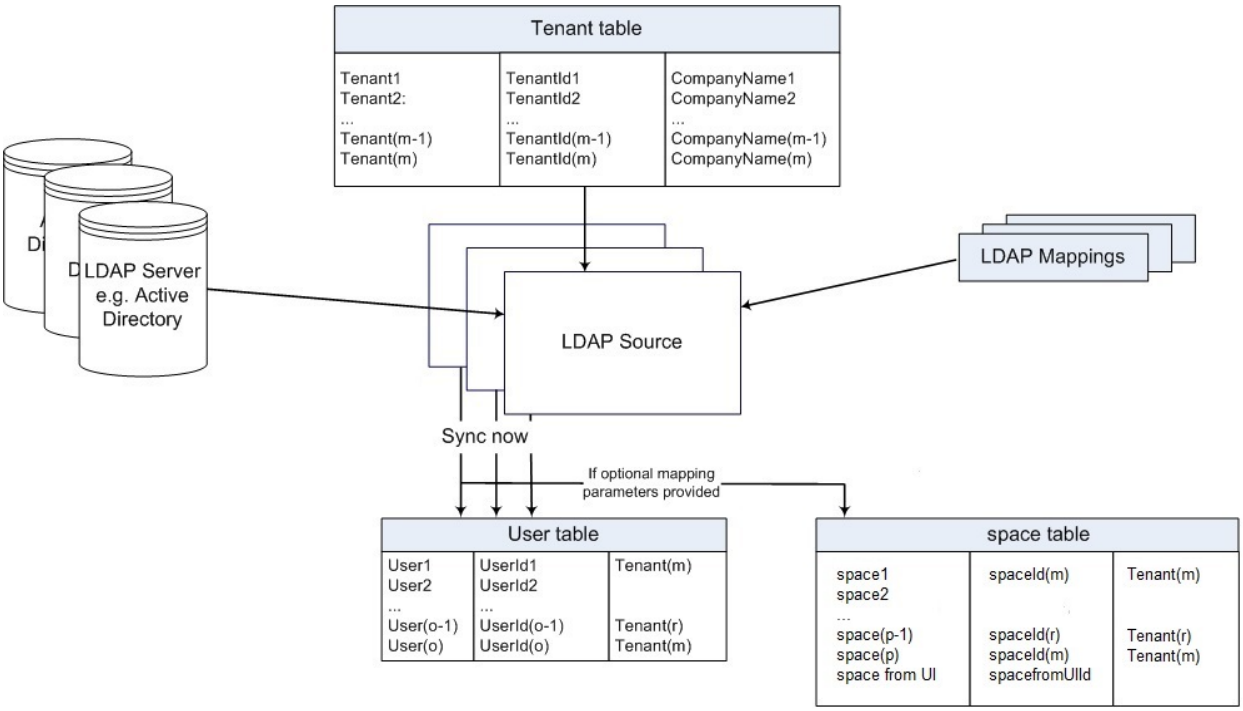
- coSpaces ("/coSpaces" or "/coSpaces/<coSpace ID>")
- LDAP sources ("/ldapSources" or "/ldapSources/<ldap source id>")
- forwarding dial plan rules ("/forwardingDialPlanRules" or "/forwardingDialPlanRules/<forwarding dial plan rule ID>")
- outbound dial plan rules ("/outboundDialPlanRules" or "/outboundDialPlanRules/<outbound dial plan rule ID>")

<tenant id> values are returned by the Meeting Server in retrieval (GET) methods for the following objects:

- coSpaces ("/coSpaces" or "/coSpaces/<coSpace ID>")
- users ("/users" or "/users/<user ID>")
- callLegs ("/callLegs", "/callLegs/<call leg id>", or "/calls/<call id>/callLegs")
- LDAP sources ("/ldapSources" or "/ldapSources/<ldap source id>")
- forwarding dial plan rules ("/forwardingDialPlanRules" or "/forwardingDialPlanRules/<forwarding dial plan rule ID>")
- outbound dial plan rules ("/outboundDialPlanRules" or "/outboundDialPlanRules/<outbound dial plan rule ID>")

For enumerations of all of these objects a "tenantFilter" value can be supplied in the requested URI in order to retrieve only those objects associated with the specified tenant.

Figure 4: Outline multi-tenancy process



12.1 Tenants

12.1.1 Retrieving Tenants

GET method performed on a "/tenants" node.

Parameters	Type/Value	Description/Notes
filter	Text	Supply filter=<tenant> in the URI to return just those tenants that match the filter
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2).
callLegProfileFilter	ID	Supply callLegProfileFilter=<call leg profile id> to return just those coSpaces using that call leg profile

Response elements	Type/Value	Description/Notes
		Response is structured as a top-level <tenants total="N"> tag with potentially multiple <tenant> elements within it.
tenant id name	ID Text	<tenant> elements follow the general form on the left.
tenantGroup	ID	If specified, associate this tenant with the supplied tenant group; the IDs of coSpaces in tenants within the same tenant group must be unique.

12.1.2 Creating and Modifying a Tenant

- Creating: POST method to the "/tenants" node. If the tenant is created successfully, an ID for the new tenant is returned in the "Location" field of the response header
- Modifying: PUT method performed on a "/tenants/<tenant id>" node

Parameters	Type/Value	Description/Notes
name *	Text	A label for the tenant
tenantGroup	ID	If specified, associate this tenant with the supplied tenant group; the IDs of coSpaces in tenants within the same tenant group must be unique.
callLegProfile	ID	If specified, associates the specified call leg profile with this tenant
callProfile	ID	If specified, associates the specified call profile with this tenant
dtmfProfile	ID	If specified, associates the specified DTMF profile with this tenant
ivrBrandingProfile	ID	If specified, associates the specified IVR branding profile with this tenant
callBrandingProfile	ID	If specified, associates the specified call branding profile with this tenant
participantLimit	Number	If specified, sets a limit on the number of participants associated with this tenant that can be simultaneously active; new participants beyond this limit will not be permitted.
userProfile	ID	If supplied, a user profile to associate with this tenant; unless otherwise overridden, all users associated with this tenant will use this user profile

Parameters	Type/Value	Description/Notes
lobbyProfile	ID	If specified, associates the specified lobby profile with this tenant.
dialInSecurityProfile	ID	If specified, associates the specified dial-in security profile with this tenant (3.0 onwards)
webBridgeProfile	ID	If specified, associates the specified web bridge profile with this tenant (3.0 onwards)

12.1.3 Retrieving Detailed Information about an Individual Tenant

GET method performed on a "/tenants/<tenant ID>" node – if the tenant ID supplied is valid, a "200 OK" response is received.

GET on **`/tenants/<tenant id>`** gives the following response values:

Parameter	Type/Value	Description/Notes
name *	Text	A label for the tenant
tenantGroup	ID	If specified, the tenant group this tenant is associated with; the IDs of coSpaces in tenants within the same tenant group must be unique.
callLegProfile	ID	If specified, the specified call leg profile associated with this tenant
callProfile	ID	If specified, the specified call profile associated with this tenant
dtmfProfile	ID	If specified, the specified DTMF profile associated with this tenant
ivrBrandingProfile	ID	If specified, the specified IVR branding profile associated with this tenant
callBrandingProfile	ID	If specified, the specified call branding profile associated with this tenant
participantLimit	Number	If specified, the limit on the number of participants associated with this tenant that can be simultaneously active; new participants beyond this limit will not be permitted.
userProfile	ID	If supplied, a user profile associated with this tenant; unless otherwise overridden, all users associated with this tenant will use this user profile
lobbyProfile	ID	If specified, associates the specified lobby profile with this tenant.
dialInSecurityProfile	ID	If specified, the specified dial-in security profile associated with this tenant (3.0 onwards)
webBridgeProfile	ID	If specified, the specified web bridge profile associated with this tenant (3.0 onwards)

12.2 Tenant group operations

12.2.1 Retrieving Tenant Groups

GET method performed on a "/tenantGroups" node.

Parameters	Type/Value	Description/Notes
offset limit	Number Number	An " offset" and " limit" can be supplied to retrieve elements other than the first " page" in the notional list (see Section 4.2.2). Response is structured as a top-level <tenantGroups total="N"> tag with potentially multiple <tenantGroup> elements within it.
tenantGroups	Number	Number of tenant groups
tenantGroup id	ID	Id for each tenant group

12.2.2 Creating and Modifying a Tenant Group

- Creating: POST method to the "/tenantGroups" node. If the tenant group is created successfully, an ID for the new tenant group is returned in the "Location" field of the response header
- Modifying: PUT method performed on a "/tenantGroups/<tenant group id>" node

12.2.3 Retrieving Detailed Information about an Individual Tenant Group

GET method performed on a "/tenantGroups/<tenant group id>" node. If the tenant ID supplied is valid, a "200 OK" response is received.

13 Query Methods

13.1 accessQuery Method

The accessQuery method finds details of how a given URI or call ID (for example, one that could be associated with a coSpace) might be reached. One use is an external system discovering that a coSpace with URI " sales.meeting" would be reached via the SIP URI " sales.meeting@example.com" .

POST performed on the /api/v1/accessQuery node.

Parameters	Type/Value	Description/Notes
uri	Text	The URI " user part" is the part before any '@' character in a full URI. It can have a maximum length of 200 characters.
callId	Number	A numeric ID (typically 9 digits long)
tenant	ID	If supplied, limits the search to the specific tenant

None of the parameters above are mandatory but the query is only meaningful if the uri or callId is supplied.

Response format:

Response elements	Type/Value	Description/Notes
		The response includes one or more of the elements on the left within an " <accessQuery>" tag
uri	String	The full URI corresponding to the uri supplied in the request
webAddress	String	An HTTPS URI for web access to the callId supplied in the request (deprecated from version 3.1)
ivr	String	A telephone number to reach an IVR that can be supplied with the callId supplied in the request (deprecated from version 3.1)

13.2 conversationIdQuery Method

The conversationIdQuery method finds whether a conversation with a specified ID has been found.

GET performed on the /api/v1/conversationIdQuery node.

Parameters	Type/Value	Description/Notes
conversationId	String	The conversation id to search for

Response format:

Response elements	Type/Value	Description/Notes
conversationId	String	The conversation id searched for
found	true|false	Whether or not a conversation with the specified id has been found.

13.3 uriUsageQuery Method

The uriUsageQuery method finds the coSpace, user and/or IVR using the specified URI within the specified tenant.

POST performed on the /api/v1/uriUsageQuery node.

Parameters	Type/Value	Description/Notes
uri	String	The " user part" of the URI; that is, the part before any '@' character in a full URI. It can have a maximum length of 200 characters.
tenant	ID	If supplied, only those coSpaces, users and IVRs within the specified tenant will be returned. If omitted only entities without a tenant will be returned.

Response format:

Response elements	Type/Value	Description/Notes
coSpaceId	ID	the id of the coSpace that uses, or whose access methods use, the specified URI
userId	ID	the id of the user that uses the specified URI
ivrId	ID	the id of the IVR that uses the specified URI

14 Dial-in security profile methods

14.1 General information

Version 3.0 introduces the minimum passcode length feature which is configurable by an administrator in order to improve security and adhere to an individual company's security policies. The minimum passcode length can be applied to all the different methods of dialing in, for example IVRs, direct SIP dial, and web app.

The minimum passcode length is defined in the new API object **`/dialInSecurityProfiles`**. The newly defined security profile can then be assigned to the top level (global) profile, tenants, coSpaces, or accessMethods. The profile can also be assigned to coSpaceTemplates and /coSpaceTemplates/<coSpace template id>/accessMethodTemplates.

There is a hierarchy of profiles – values in the profiles lower in the hierarchy override those set above, and if a parameter is unset or no dial-in security profile is set then it inherits from the next profile up within the hierarchy.

The hierarchy for dialInSecurityProfile is:

- Top level (global) profile (**`/system/profiles`**)
- Tenants (**`/tenants/<tenant id>`**)
- coSpaces (**`/coSpaces/<cospace id>`**)
- accessMethod (**`/coSpaces/<cospace id>/accessMethods/<access method id>`**)

Dial-in security profiles can also be applied to cospace templates and cospace access method templates as follows:

- coSpaceTemplates (**`/coSpaceTemplates/<coSpace template id>`**)
- accessMethodTemplates (**`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method template id>`**)

When coSpaces and their associated access methods get instantiated from templates, the dial-in security profiles from the templates get assigned to the corresponding instantiated objects.

Note: If you use TMS versions earlier than 15.12.0 for scheduled meetings – CUCM ad hoc conferencing calls – do not set a security profile at the system or tenant level.

Note: If the parameter **`minPasscodeLength`** is set to **`0`**, it will result in no passcode length enforcement.

14.1.1 Using dial-in security profiles to implement minimum passcode length

All **`dialInSecurityProfile`** parameters are optional. If these parameters are not specified at any level, the default settings will be minPassCodeLength=0 and allowOutOfPolicy=true. The default setting in dialInSecurityProfiles is <unset>.

14.2 Retrieving dial-in security profiles

GET on individual profiles with **`/dialInSecurityProfiles/<dial in security profile id>`** gives the following response:

Response elements	Type/Value	Description/Notes
name	String	The human-readable name associated with this dial-in security profile. It can have a maximum length of 200 characters.
minPasscodeLength	Number	Minimum allowed password length, can be between 0 and 200 (inclusive)
allowOutOfPolicy	**true**|false	Whether or not users are allowed to join a call using an old passcode that was set before the dial-in security profile was applied and which is not compliant with the newly defined passcode length. If this parameter is not supplied in a create (POST) operation, it defaults to " true" .

Enumeration of **`/dialInSecurityProfiles`** accepts the following URI parameters:

URI parameters	Type/Value	Description/Notes
offset		an offset and limit can be supplied to retrieve dial-in security profiles other than the first page in the notional list
limit		
usageFilter	unreferenced |referenced	Supply " usageFilter=unreferenced" in the request to retrieve only those dial-in security profiles that are not referenced by global settings or any other object. This is a useful check before deleting the profile. To retrieve just those dial-in security profiles which are referenced in at least one place, you can supply " usageFilter=referenced"

Response is structured as a top-level <dialInSecurityProfiles total=" N" > tag with potentially multiple <dialInSecurityProfile> elements within it.

Each <dialInSecurityProfile> tag may include the following elements:

Response elements	Type/Value	Description/Notes
name	String	The human-readable name associated with this dial-in security profile. It can have a maximum length of 200 characters.
minPasscodeLength	Number	Minimum allowed password length, can be between 0 and 200 (inclusive)

Response elements	Type/Value	Description/Notes
allowOutOfPolicy	**true**|false	Whether or not users are allowed to join a call using an old passcode that was set before the dial-in security profile was applied and which is not compliant with the newly defined passcode length. If this parameter is not supplied in a create (POST) operation, it defaults to " true" .

14.3 Creating and modifying dial-in security profiles

The new **`/dialInSecurityProfiles`** object is used to implement dial-in security profiles. This API node supports the following operations:

- POST to **`/dialInSecurityProfiles`** to create a new dial-in security profile
- PUT on individual profiles with **`/dialInSecurityProfiles/<dial in security profile id>`**

with the following request parameters:

Parameters	Type/Value	Description/Notes
name	String	The human-readable name associated with this dial in security profile. It can have a maximum length of 200 characters.
minPasscodeLength	Number	Minimum allowed passcode length, can be between 0 and 200 (inclusive)
allowOutOfPolicy	**true**|false	Whether or not users are allowed to join a call using an old passcode that was set before the dial-in security profile was applied and which is not compliant with the newly defined passcode length. If this parameter is not supplied in a create (POST) operation, it defaults to " true" .

15 Scheduler APIs

Scheduler introduces new API nodes in version 3.3. The scheduler has its own HTTPS interface which when enabled, can be used to configure scheduler meetings using the scheduler APIs.

Meeting Server management API interface does not support Scheduler APIs. Third part tools like Postman can be used for all requests, browser can also be used for GET APIs. See Third Party Tools to Use.

New API nodes are available at the address specified in the new command **`scheduler https listen <interface> <port>`**, and are prefixed with **`https://hostname:port/api/v1/scheduler`**.

For example,

`https://hostname:port/api/v1/scheduler/health` or
`https://hostname:port/api/v1/scheduler/meetings`

The Scheduler API nodes are:

- **`/health`**
- **`/meetings`**
- **`/meetings/<meeting id>`**
- **`/meetings/<meeting id>/participants`**
- **`/meetings/<meeting id>/recurrences/<recurrence id>`**
- **`/meetings/<meeting id>/recurrences/<recurrence id>/participants`**
- **`/meetings/<meeting ID>/refreshEmails`**
- **`/meetings/<meeting ID>/recurrences/<recurrence id>/refreshEmails`**
- **`/timedLogging`**

15.1 Configuring the Scheduler's HTTPS interface

The scheduler has its own HTTPS interface which if enabled, can be used to configure scheduler meetings using the scheduler APIs. The Web Bridge however, does not communicate with the scheduler using the management API. Though it is not mandatory to enable the HTTPS server, it is recommended that you do so because it provides some diagnostic and troubleshooting functionality.

Configure the HTTPS server listen interface using the command:

`scheduler https listen <interface> <port>`

For example:

`scheduler https listen a 8443`

Configure a certificate key pair for the server using the command:

scheduler https certs <key-file> <crt-fullchain-file>

For example:

scheduler https certs scheduler_https.key scheduler_https.cer

15.2 Scheduler API message format

For its API operations, the Scheduler message body uses the JSON format.

For example,

POST to **http://localhost:8082/api/v1/scheduler/meetings** with:

```
{
"coSpace": "9ed57ad2-26aa-469f-85d1-e62bbe75cdd6",
"summary": "Test meeting",
"dtStart": "2021-07-20T22:00:00.000",
"dtEnd": "2021-07-20T23:00:00.000",
"rrule":
"FREQ=YEARLY;BYMONTH=7;BYMONTHDAY=9;UNTIL=20220710T235900Z",
"timeZone": "Europe/Budapest",
"participants": [
{
"accessMethod": "527f439b-9527-4718-bc1c-d2511856c268",
"userName": "testuser",
"email": "testuser@example.com"
}
],
"organizerUserName": "testorganizer",
"organizerEmail": "testorganizer@example.com",
"organizerDisplayName": "Test Organizer"
}
```

15.3 Content header format

```
Header key: Content-Type
Header value: application/json
```

15.4 Response codes

Response Codes	Description
200	The request is successful.

Response Codes	Description
400	Bad request, verification failures.
403	Forbidden access to API. Typically occurs if the user tries to create a meeting in a spacein which they are not a member.
404	Meeting ID not found.
405	Method not allowed. For example, if PUT is used instead of POST.
415	Unsupported content type. Occurs if the user tries to send the body in a content type other than JSON.
500	Internal server error.
503	Service unavailable. Typically occurs when the Scheduler is running but it does not have a connection to the database.

15.5 Custom Error Codes

Following is a list of Error IDs and the corresponding error messages that the Scheduler sends in case of any user errors. The Error ID represents unique id assigned to the error messages and can be used to map the error messages with the scheduler interfaces.

Error ID	Error Message
1003	Meeting required field is missing
1004	Invalid meeting start date/time format
1005	Invalid meeting end date/time format
1006	Invalid meeting start and end date/time order
1007	Invalid recurrence rule value while creating/updating the meeting
1008	Invalid participant field values while creating the meeting
1009	Invalid meeting organizer email
1010	Invalid meeting organizer email display name
1013	No meeting end date/time and no all day meeting
1018	Time zone is not valid
1019	This meeting overlaps with an existing meeting

15.6 Example request and response

This is an example for retrieving scheduled meetings in a specified time period.

REQUEST:

```
GET /api/v1/scheduler/meetings?fromTime=2021-06-
27T22%3A00%3A00.000Z&untilTime=2021-07-
04T21%3A59%3A59.999Z&maxMeetings=50 HTTP/1.1
Host: localhost:8082
```

RESPONSE:

```
HTTP/1.1 200 OK Content-Type: application/json

[
{
"coSpace": "9ed57ad2-26aa-469f-85d1-e62bbe75cdd6",
"summary": "Test 1",
"dtStart": "2022-07-09T22:00:00",
"dtEnd": "2022-07-09T23:00:00",
"timeZone": "Europe/Budapest",
"meeting": "1aa0cf29-ea53-4fa8-b63a-7675aba85e36",
"isSchedulerSpaceOwner": false,
"isFullDayMeeting": null,
"rrule":
"FREQ=YEARLY;BYMONTH=7;BYMONTHDAY=9;UNTIL=20220710T235900
Z",
"organizerEmail": "amilicev@example.com",
"organizerUserName": "amilicev@example.com",
"participantCount": 1,
"isInvited": null,
"recurrence": "2022-07-09T22:00:00"
}
]
```

This is an example for creating a meeting with wrong start time format.

REQUEST:

```
POST to http://localhost:8082/api/v1/scheduler/meetings
HTTP/1.1 with:

{
"coSpace": "0002c44c-9abb-4ff2-a522-29e67082fd56",
"summary": "Test Meet",
"dtStart": "2022-09-08T12:00:00.000ZZ",
"dtEnd": "2022-09-08T13:00:00.000",
"timeZone": "Asia/Calcutta",
"isSchedulerSpaceOwner": false,
"participants": [
{
"accessMethod": "18579b6f-072d-4e93-b8bc-8d7a6d5853ce",
"userName": "test@example.com",
"email": "test@example.com"
```

```
}
],
"organizerUserName": "test@example.com",
"organizerEmail": "test@example.com",
"organizerDisplayName": "Test"
}
```

Response:

HTTP/1.1 400 Bad Request Content-Type: application/json

```
{
"code": 400,
"errorId": 1004,
"message": "Invalid meeting start date/time format."
}
```

15.7 Scheduler related methods

Scheduler APIs support scheduling, modifying or deleting meetings with single or multiple occurrences, adding or removing participants, retrieving health information, and sending or refreshing email notifications.

15.7.1 Retrieving service health status

Retrieving service health status is supported by the API node **`/health`** using the method:

- GET on **`/health`**

Request element	Type/Value	Description/ Notes
status	String	Overall service health status

<table>
<tr><th>Request element</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>component</td><td>Array</td><td>
<table>
<tr><th>Response element</th><th>Type/Value</th><th>Description</th></tr>
<tr><td>db:</td><td></td><td></td></tr>
<tr><td>status</td><td>String</td><td>UP or DOWN</td></tr>
<tr><td>mail:</td><td></td><td></td></tr>
<tr><td>status</td><td>String</td><td>UP or DOWN</td></tr>
<tr><td>details</td><td>Object</td><td>Server address and port.</td></tr>
<tr><td>cmsWebScheduler:</td><td></td><td></td></tr>
<tr><td>status</td><td>String</td><td>UP or DOWN</td></tr>
<tr><td>details</td><td>Object</td><td>guid used as the Scheduler's identification on c2w connections.</td></tr>
<tr><td>ping:</td><td></td><td></td></tr>
<tr><td>status</td><td>String</td><td>UP or DOWN</td></tr>
</table>
</td></tr>
</table>

15.7.2 Creating, modifying, and deleting meetings in a coSpace

- POST to **`/scheduler/meetings`**
- GET on **`/scheduler/meetings`**
- DELETE on **`/scheduler/meetings`**

Creating a meeting is supported by POST to **`/scheduler/meetings`** with the following parameters:

Parameters	Type/Value	Description/ Notes
coSpace*	ID	coSpace ID.
organizerEmail	string	Meeting organizer's email.
organizerUserName	string	Meeting organizer's userName.
organizerDisplayName (optional)	String	Meeting organizer’s name to be displayed in the email.
summary	string	Meeting summary, max length 50.

Parameters	Type/Value	Description/ Notes
isSchedulerSpaceOwner	true | false	true if the scheduler component is responsible for managing the lifetime of the coSpace. **Note:** When set to true, the scheduler will delete the coSpace approximately 24 hours after the end of the meeting / meeting series.
isFullDayMeeting	true | false	true if Meeting is a full day meeting. Either dtEnd or isFullDayMeeting is required.
dtEnd	string	Meeting end date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000 Either dtEnd or isFullDayMeeting is required.
dtStart*	string	Meeting start date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000
rrule	string	Frequency of the recurring meeting. Example: FREQ=YEARLY; BYMONTH=10; BYDAY=-1SU;UNTIL=20301105T082754Z
participants	Array	List of participants objects.

Parameters	Type/Value	Description/ Notes
email	string	User's email with email verification.
userName	string	Meeting participant's userName.
accessMethod	ID	accessMethod ID. If not supplied, then the cospace level join information will be used for invites.

timeZone	string	The region or location corresponding to the dtStart and dtEnd elements. Example: " Europe/London" If not supplied, defaults to " UTC" .

Retrieving meeting occurrences is supported by GET on **`/scheduler/meetings`** with the following request parameters.

Parameters	Type/Value	Description/ Notes
coSpace	ID	coSpace ID.
maxMeetings	numeric	The maximum number of meetings to retrieve. Returns the first n meetings (after fromTime), ordered in ascending order of startDate. The minimum value is 1 minimum and default is 5.
fromTime	string	Calculated against meeting end date in ISO 8601 full-time format. Example: 2021-01-30T08:30:00Z. If unset, it defaults to the current time.
untilTime	string	Calculated against meeting start date in ISO 8601 full-time format. Example: 2021-01-30T08:30:00Z. If unset, then there is no upper limit on untilTime.
userName	string	Either organizerUserName or participant's userName.

Response element for GET on **`/scheduler/meetings`** is a list of meetings, each of which can have the following parameters:

Response elements	Type/Value	Description/ Notes
coSpace	ID	coSpace ID.
summary	string	Meeting title.
dtStart	string	Meeting start date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000
dtEnd	string	Meeting end date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000 Either dtEnd or isFullDayMeeting is returned.
meeting	ID	Meeting ID.
isSchedulerSpaceOwner	true | false	true if the scheduler component is responsible for managing the lifetime of the coSpace. **Note:** When set to true, the scheduler will delete the coSpace approximately 24 hours after the end of the meeting / meeting series.
isFullDayMeeting	true | false	true if Meeting is full day meeting. Either dtEnd or isFullDayMeeting is returned.

Response elements	Type/Value	Description/ Notes
rrule	string	Frequency of the recurring meeting. Example: FREQ=YEARLY; BYMONTH=10; BYDAY=-1SU; UNTIL=20301105T082754Z
organizerEmail	string	Meeting organizer's email.
organizerUserName	string	Meeting organizer's userName
participantCount	numeric	Number of participants invited to the meeting.
recurrence	string	Start time of the first occurrence of the recurrent meeting in ISO 8601 full-time format.
timeZone	string	The region or location corresponding to the dtStart and dtEnd elements. Example: " Europe/London" If not supplied, defaults to " UTC"

Deleting meetings in a coSpace is supported by DELETE on **`/scheduler/meetings`** with the **`coSpace`** parameter.

Request element	Type/Value	Description/ Notes
coSpace*	string	coSpace ID, with guid verification, length 36

15.7.3 Retrieving, modifying, or deleting individual meetings

Retreiving information on a individual meeting is supported by GET on **`/scheduler/meetings/<meeting id>`** and can return the following parameters:

Response elements	Type/Value	Description/ Notes
coSpace	ID	coSpace ID.
summary	string	Meeting title.
dtStart	string	Meeting start date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000
dtEnd	string	Meeting end date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000 Either dtEnd or isFullDayMeeting is returned.
meeting	ID	Meeting ID.

Response elements	Type/Value	Description/ Notes
isSchedulerSpaceOwner	true | false	true if the scheduler component is responsible for managing the lifetime of the coSpace. **Note:** When set to true, the scheduler will delete the coSpace approximately 24 hours after the end of the meeting / meeting series.
isFullDayMeeting	true | false	true if Meeting is full day meeting. Either dtEnd or isFullDayMeeting is returned.
rrule	string	Frequency of the recurring meeting. Example: FREQ=YEARLY; BYMONTH=10; BYDAY=-1SU; UNTIL=20301105T082754Z
organizerEmail	string	Meeting organizer's email.
organizerUserName	string	Meeting organizer's userName
participantCount	numeric	Number of participants invited to the meeting.
recurrence	string	Start time of the first occurrence of the recurrent meeting in ISO 8601 full-time format.
timeZone	string	The region or location corresponding to the dtStart and dtEnd elements. Example: " Europe/London" If not supplied, defaults to " UTC"

Modifying an individual meeting is supported by PUT to **`/scheduler/meetings/<meetingId>`** with the following parameters:

Parameters	Type/Value	Description/ Notes
coSpace	ID	coSpace ID.
organizerEmail	string	Meeting organizer's email.
organizerUserName	string	Meeting organizer's userName.
summary	string	Meeting title.
isSchedulerSpaceOwner	true | false	true if the scheduler component is responsible for managing the lifetime of the coSpace. **Note:** When set to true, the scheduler will delete the coSpace approximately 24 hours after the end of the meeting / meeting series.

Parameters	Type/Value	Description/ Notes
isFullDayMeeting	true | false	true if Meeting is full day meeting. Either dtEnd or isFullDayMeeting is returned.
dtStart	string	Meeting start date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000
dtEnd	string	Meeting end date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000 Either dtEnd or isFullDayMeeting is returned.
timeZone	string	The region or location corresponding to the dtStart and dtEnd elements. Example: " Europe/London" If not supplied, defaults to " UTC" .
rrule	string	Frequency of the recurring meeting. Example: FREQ=YEARLY; BYMONTH=10; BYDAY=-1SU; UNTIL=20301105T082754Z
participants	Array	List of participants objects.

Parameters	Type/Value	Description/ Notes
email	string	User's email with email verification.
userName	string	Meeting participant's userName.
accessMethod	ID	accessMethod ID. If not supplied, then the cospace level join information will be used for invites.

Deleting meeting instances is supported by DELETE on **`/scheduler/meetings/<meeting Id>`**

Parameters	Type/Value	Description/ Notes
meetingId	ID	Meeting ID of the meeting to be deleted.

15.7.4 Modifying or retrieving the list of participants in a meeting

Retrieving participants of a meeting is supported by GET on **`/scheduler/meetings/<meetingId>/participants`** and each participant can include the

following objects:

<table>
<tr><th>Request element</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>participants</td><td>Array</td><td>List of participants objects.
<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>email</td><td>string</td><td>User's email with email verification.</td></tr>
<tr><td>userName</td><td>string</td><td>Meeting participant's userName.</td></tr>
<tr><td>accessMethod</td><td>ID</td><td>accessMethod ID. If not supplied, then the cospace level join information will be used for invites.</td></tr>
</table></td></tr>
</table>

Modifying the participants in a meeting or meeting series is supported by PUT to **`/scheduler/meetings/<meetingId>/participants`** with the following parameters:

<table>
<tr><th>Parameter</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>newParticipants</td><td>Array</td><td>Participants to be added to the invite list.
<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>email</td><td>string</td><td>User's email with email verification.</td></tr>
<tr><td>userName</td><td>string</td><td>Meeting participant's userName.</td></tr>
<tr><td>accessMethod</td><td>ID</td><td>accessMethod ID. If not supplied, then the cospace level join information will be used for invites.</td></tr>
</table></td></tr>
<tr><td>deletedEmailParticipants</td><td>array or strings</td><td>Participants with these email addresses will be removed from the invited list.</td></tr>
<tr><td>deletedUserParticipants</td><td>array or strings</td><td>Participants with these userNames will be removed from the invited list.</td></tr>
</table>

15.7.5 Creating, modifying, and deleting meeting occurrences

Retrieving meeting occurrences is supported by GET on **`/scheduler/meetings/<meeting id>/recurrences/<recurrence id>`** and returns the following values:

Response elements	Type/Value	Description/ Notes
coSpace	ID	coSpace ID.
summary	string	Meeting title.
dtStart	string	Meeting start date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000
dtEnd	string	Meeting end date in LocalDateTime full-time format. Example: 2021-01-30T08:30:00.000 Either dtEnd or isFullDayMeeting is returned.
timeZone	string	The region or location corresponding to the dtStart and dtEnd elements. Example: " Europe/London" If not supplied, defaults to " UTC.
meeting	ID	Meeting ID.
isSchedulerSpaceOwner	true | false	true if the scheduler component is responsible for managing the lifetime of the coSpace. **Note:** When set to true, the scheduler will delete the coSpace approximately 24 hours after the end of the meeting / meeting series.
isFullDayMeeting	true | false	true if Meeting is full day meeting. Either dtEnd or isFullDayMeeting is returned.
rrule	string	Frequency of the recurring meeting. Example: FREQ=YEARLY; BYMONTH=10; BYDAY=-1SU; UNTIL=20301105T082754Z
organizerEmail	string	Meeting organizer's email.
organizerUserName	string	Meeting organizer's userName.
participantCount	integer	Number of participants invited to the meeting.
isCancelled	true | false	true if the meeting is canceled.

Modifying a meeting occurrence is supported by PUT to **`/scheduler/meetings/<meeting id>/recurrences/<recurrence id>`** with the following parameters:

Request parameters	Type/Value	Description/ Notes
summary	string	Meeting title.

<table>
<tr><th>Request parameters</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>isFullDayMeeting</td><td>true | false</td><td>true if Meeting is full day meeting.
Either dtEnd or isFullDayMeeting is required.</td></tr>
<tr><td>dtStart</td><td>string</td><td>Meeting start date in LocalDateTime full-time format.
Example: 2021-01-30T08:30:00.000</td></tr>
<tr><td>dtEnd</td><td>string</td><td>Meeting end date in LocalDateTime full-time format.
Example: 2021-01-30T08:30:00.000
Either dtEnd or isFullDayMeeting is required.</td></tr>
<tr><td>timeZone</td><td>string</td><td>The region or location corresponding to the dtStart and dtEnd elements.
Example: " Europe/London"
If not supplied, defaults to " UTC" .</td></tr>
<tr><td>participants</td><td>Array</td><td>List of participants objects.
<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>email</td><td>string</td><td>User's email with email verification.</td></tr>
<tr><td>userName</td><td>string</td><td>Meeting participant's user-Name.</td></tr>
<tr><td>accessMethod</td><td>ID</td><td>accessMethod ID. If not supplied, then the cospace level join information will be used for invites.</td></tr>
</table></td></tr>
<tr><td>isCancelled</td><td>true | false</td><td>Indicates if the meeting is cancelled. If set to true, it cancels the meeting instance by recurrence ID.</td></tr>
</table>

15.7.6 Retrieving or modifying the participants in a meeting occurrence

Retrieving a list of participants in a meeting occurrence is supported by GET on **`/scheduler/meetings/<meeting id>/recurrences/<recurrence id>/participant`** with the following parameters:

<table>
<tr><th>Request element</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>participants</td><td>Array</td><td>List of participants objects.
<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>email</td><td>string</td><td>User's email with email verification.</td></tr>
<tr><td>userName</td><td>string</td><td>Meeting participant's userName.</td></tr>
<tr><td>accessMethod</td><td>ID</td><td>accessMethod ID. If not supplied, then the cospace level join information will be used for invites.</td></tr>
</table></td></tr>
</table>

Modifying the participants in a meeting occurance is supported by PUT to **`/scheduler/meetings/<meeting id>/recurrences/<recurrence id>/participants`** with the following parameters:

<table>
<tr><th>Parameter</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>newParticipants</td><td>Array</td><td>Participants to be added to the invite list.
<table>
<tr><th>Parameters</th><th>Type/Value</th><th>Description/ Notes</th></tr>
<tr><td>email</td><td>string</td><td>User's email with email verification.</td></tr>
<tr><td>userName</td><td>string</td><td>Meeting participant's userName.</td></tr>
<tr><td>accessMethod</td><td>ID</td><td>accessMethod ID. If not supplied, then the cospace level join information will be used for invites.</td></tr>
</table></td></tr>
<tr><td>deletedEmailParticipants</td><td>array or strings</td><td>Participants with these email addresses will be removed from the invited list.</td></tr>
<tr><td>deletedUserParticipants</td><td>array or strings</td><td>Participants with these userNames will be removed from the invited list.</td></tr>
</table>

15.7.7 Refreshing emails for a coSpace

Refreshing emails for a coSpace is supported by:

POST on **`/scheduler/meetings/refresh?coSpace=<coSpace id>`**

Parameters	Type/Value	Description/ Notes
meeting*	ID	Meeting id.

15.7.8 Refreshing emails for single meeting and whole meeting series

Refreshing emails for a meeting is supported by:

PUT to **`/scheduler/meetings/<meeting ID>/refreshEmails`**

Parameters	Type/Value	Description/ Notes
meeting*	ID	Meeting id.

Refreshing emails for one meeting recurrence from the meeting series is supported by:

PUT to**`/scheduler/meetings/<meeting ID>/recurrences/<recurrence id>/refreshEmails`**

Parameters	Type/Value	Description/ Notes
meeting*	ID	Meeting id.
reccurence*	string	Id of single instance in LocalDateTime full-time format. Example: 2021-01-30T08:30:00Z

15.7.9 Modifying and retrieving timed logging information

This feature introduces a new API node **`/scheduler/timedLogging`** to support the following operations:

- PUT to **`/scheduler/timedLogging`**
- GET on **`/scheduler/timedLogging`**

Parameters	Type/Value	Description/ Notes
webBridge	numeric	time remaining (in seconds) for which detailed Web Bridge logging should be enabled.
api	numeric	Time remaining for HTTPS side logging (in seconds).
email	numeric	Time remaining for email logging (in seconds).

16 Using Profiles

There are a number of profiles you can use:

- /system/profiles (this is the top-level profile)
- /callProfiles
- /callLegProfiles
- /callBrandingProfiles
- /dtmfProfiles
- /ivrBrandingProfiles
- /userProfiles
- /compatibilityProfiles
- /dialInSecurityProfiles
- /webBridgeProfiles
- /lobbyProfiles

The top-level profile can include all the other profiles.

Equally, all the other profiles can be set, modified or retrieved for a tenant. For example, on a per-tenant basis you can set a different callBrandingProfile to the top-level system profile so that calls for tenant A have different branding from calls to users who do not belong to a tenant and from calls for tenant B.

coSpaces, access methods, users and IVRs can be created with some of the profiles, see the following figures. The values set in the profiles in these definitions would override any values set at the top level or tenant level. For example, the values in the callLegProfile set for a coSpace would be used for call legs in that coSpace — overriding any values set in the callLegProfile specified for the tenant that the coSpace is associated with, or the top-level profile.

Finally, you can specify a value for one of the parameters that also appears in a profile in each individual call leg. Then you can use a profile for most parameters but override just one such as the defaultLayout for a call leg.

Using profiles is optional at all levels.

The following figures show the relationship, inheritance and overrides between profiles.

Figure 5: Value inheritance with profiles

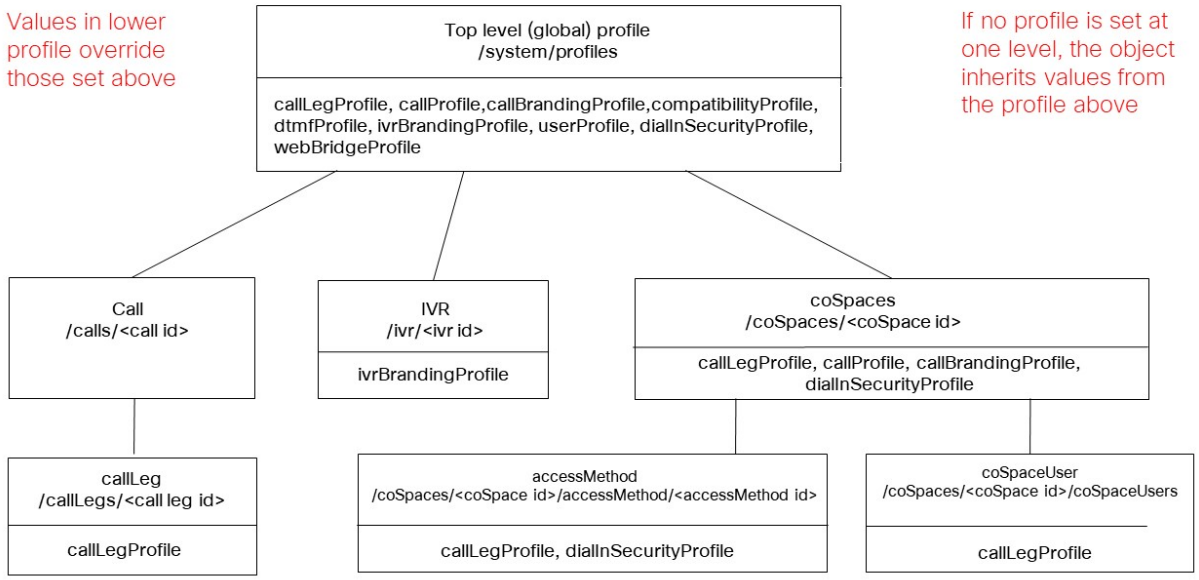
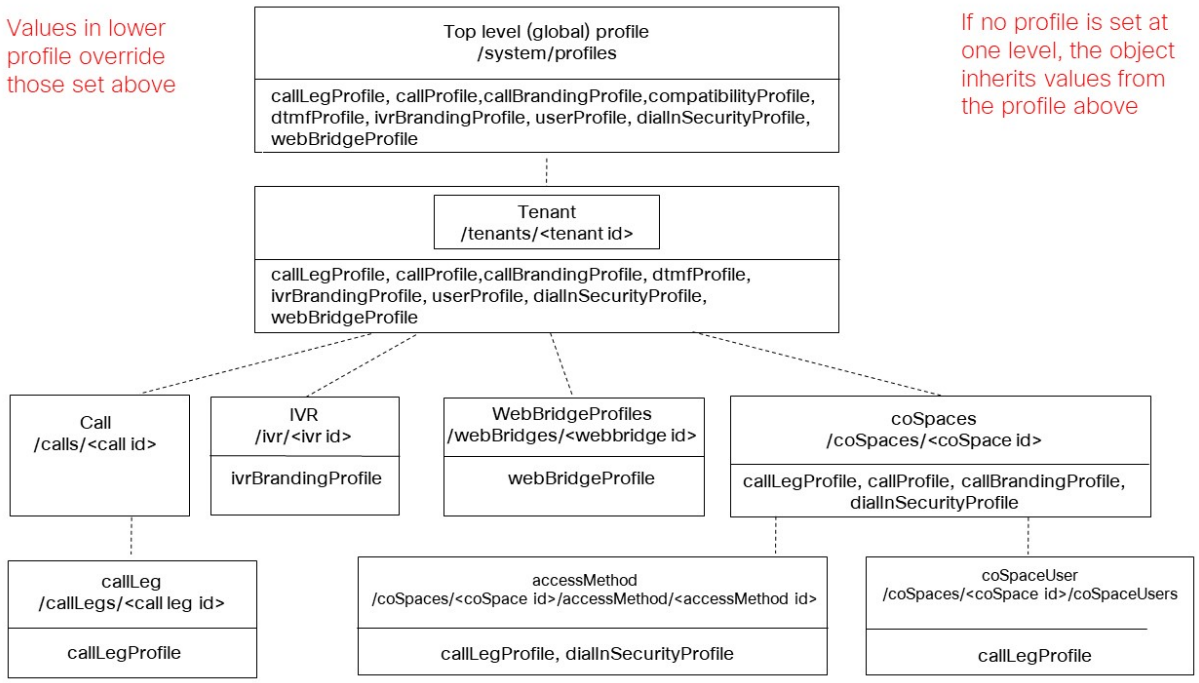


Figure 6: Value inheritance with profiles when using tenants



Appendix A mall Change participant roles in a web app meeting

Meeting Server does not have pre-configured roles for the meeting participants. Admins define the role names and their associated permissions while provisioning Meeting Server using Meeting Management (see Meeting Management User Guide for Administrators for more information). Users will be assigned these roles based on the access method they use to join the meeting. Meeting participants connected via the Cisco Meeting server web app can change the role of other participants. This feature is also supported via the Web Admin API for coSpace calls.

Note: This feature is supported for participants joining from web app and SIP/CE endpoints or clients only. Changing the role for participants who have joined from other clients like Lync or Skype is not supported.

Note: Cisco Endpoints with ActiveControl enabled will not change their video or video+presentation permissions with a mid call role change. This could be changing video or video+presentation from disabled to enabled or vice-versa. For example, a participant joined as a Guest where video permission is disabled. During the call, if their role is changed to Host where video permission is enabled, video still remains disabled on the endpoint.

The **`changeRoleAllowed`** parameter supported on callLegProfiles, callLegs, and participants determines whether a participant is allowed to change the role of other participants in-call. The effective value of **`changeRoleAllowed`** is computed based on the existing rules for the hierarchy of call leg profiles and, if undefined at all levels of the hierarchy, it defaults to false.

A.1 Available roles

In the web app

The initiator (who changes the role of a participant) has certain roles available to assign based on their:

- Access method Scope
- coSpace membership / ownership
- Initial role of the initiator participant
- Current role of the target participant (whose role is being changed)

The initiator participant has access to:

- all access methods of the coSpace regardless of their scope, if the initiator participant is the owner of the coSpace.
- all public and directory access methods of the coSpace.
- access methods with member scope only if they are the coSpace owner or they are a member of the coSpace.
- access methods with private scope if they are the coSpace owner or if they joined using that particular access method as a coSpace member.

If the initiator participant has access to an access method, they can assign it to another participant provided that they are able to revert the role back to what it was. The exception to this rule is the role “Space default”: the initiator can not assign a “Space default” role to a participant who has another role. Refer to Space default role for more information on " Space default" role.

From the Web Admin

An Admin user can change the access method of a participant or a call leg object by specifying a value for accessMethod when doing PUT on /callLegs/<call leg id>. The Admin user has access to all access methods configured on the coSpace and in addition can unset the access method, i.e. provide a value “”, which also removes the access methods’s call leg profile from the call leg profile hierarchy of the object and unsets the access method’s importance value.

A.2 Special roles

The role of a participant is displayed as " Space default" or " Custom" in the following scenarios:

A.2.1 Space default role

If meetings are provisioned by Meeting Management admin (using the procedure provided in the Meeting Management User Guide for Administrators) then all users or access methods will be assigned a named role. If the spaces were created using older methods then it is possible that the role will be shown as " Space default" .

- If the coSpace has a callId/uri on the coSpace object, a virtual access method " Role 1" is created on the web appspace portal for coSpace members using GUID 00000000-0000-0000-0000-000000000001. When coSpace members join a call using this role, the in-call role is shown as " Space default" .
- In the absence of this virtual access method, if a coSpace member has an In-call role assigned as " Space default" from the web app space portal, the in-call role is shown as " Space default" .

- During a call, the Change Role menu does not display " Space default" as one of the options. If a participant's role is changed from " Space default" to any other role, their role can not be changed back to " Space default" .

A.2.2 Custom role

If the callLegprofile set for a coSpaceUser does not match the call leg profile of any of the access methods or the coSpace object, the role will be shown as " Custom" in-call on web app. This role cannot be changed in web app.

A.3 Assigning roles and coSpace membership

A.3.1 Members

A role by any name such as Host or Guest corresponds to the assigned settings or permissions. For coSpace members, the role is changed by changing the coSpace user call leg profile. Since the access method is not changed, the result of a role change will not be reflected in GET on /callLegs/<call leg id> and on /participants/<participant id>. The importance value is also not affected.

If there are multiple access methods in a space and Role1 with GUID 00000000-0000-0000-0000-000000000001 is one of them, then while creating a meeting, the user will not be able to add Role1 as a role for a non-member. Role1 will be available only for members.

If there is only one access method in a space which is Role1 with GUID 00000000-0000-0000-0000-000000000001, then while creating a meeting, the webapp user will be able to add only members in the attendees list. If a non-member is added, then a message " Only space members can be added." is displayed.

A.3.2 Other participants

For non-coSpace members, the role is changed by changing the accessMethod. The value of the parameter accessMethod when doing GET on /callLegs/<call leg id> and on /participants/<participant id> reflects the change. In addition, the access method's call leg profile is plugged in the call leg profile hierarchy and the access method's importance value is applied to the participant as well.

Appendix B Additional Call Leg Information

B.1 Call Leg Information

When information is retrieved on a specific individual call leg, its structure follows the form:

```
<callLeg id="386621ab-927b-4624-a77d-0288913c92ac">
```

call leg response values (see Section 8.3.6)

```
  <configuration>
```

call leg configuration (see below)

```
  </configuration>
  <status>
```

call leg status (see below)

```
  </status>
</callLeg>
```

B.1.1 Call leg configuration

The configuration section returned includes the same values as those that can be modified in a PUT method on that call leg.

An example configuration section is shown below:

```
<configuration>
  <ownerId>2671a77d-4bd5-4bf2-8ed6-f14afd80c2ac</ownerId>
  <chosenLayout></chosenLayout>
  <needsActivation>false</needsActivation>
  <defaultLayout>speakerOnly</defaultLayout>
  <participantLabels>false</participantLabels>
  <presentationDisplayMode>dualStream</presentationDisplayMode>
  <presentationContributionAllowed>false</presentationContributionAllowed>
  <presentationViewingAllowed>true</presentationViewingAllowed>
  <endCallAllowed>true</endCallAllowed>
  <muteOthersAllowed>true</muteOthersAllowed>
  <videoMuteOthersAllowed>true</videoMuteOthersAllowed>
  <muteSelfAllowed>true</muteSelfAllowed>
  <videoMuteSelfAllowed>true</videoMuteSelfAllowed>
  <changeLayoutAllowed>true</changeLayoutAllowed>
  <joinToneParticipantThreshold>0</joinToneParticipantThreshold>
  <leaveToneParticipantThreshold>0</leaveToneParticipantThreshold>
  <videoMode>false</videoMode>
  <rxAudioMute>false</rxAudioMute>
  <txAudioMute>false</txAudioMute>
  <rxVideoMute>false</rxVideoMute>
  <txVideoMute>false</txVideoMute>
```

```
  <sipMediaEncryption>allowed</sipMediaEncryption>
  <audioPacketSizeMs>20</audioPacketSizeMs>
  <deactivationMode>disconnect</deactivationMode>
  <deactivationModeTime>1</deactivationModeTime>
  <telepresenceCallsAllowed>false</telepresenceCallsAllowed>
  <sipPresentationChannelEnabled>true</sipPresentationChannelEnabled>
  <bfcpMode>1</bfcpMode>
</configuration>
```

B.1.2 Call leg status

The status information returned contains live values relating to the call leg's active state. Media information is contained within one or more rxAudio, txAudio, rxVideo and txVideo sub-sections; multiple video or audio streams may be identified and distinguished via their "role" attribute, which may be either "main" or "presentation".

Each media section includes packet loss percentage and jitter.

An example set of status data is shown below:

```
<status>
  <state>connected</state>
  <durationSeconds>349</durationSeconds>
  <direction>incoming</direction>
  <sipCallId>6ff025ec40843df0b5153cab4a8601ee</sipCallId>
  <groupId>8b29e92d-27c2-421a-8d73-47daf09d7fe7</groupId>
  <recording>true</recording>
  <streaming>false</streaming>
  <deactivated>false</deactivated>
  <encryptedMedia>true</encryptedMedia>
  <unencryptedMedia>false</unencryptedMedia>
  <layout>telepresence</layout>
  <activeLayout></activeLayout>
  <availableVideoStreams></availableVideoStreams>
  <rxAudio>
     <codec>aac</codec>
     <packetLossPercentage>0.0</packetLossPercentage>
     <jitter>2</jitter>
     <bitRate>64000</bitRate>
  </rxAudio>
  <txAudio>
     <codec>aac</codec>
     <packetLossPercentage>0.0</packetLossPercentage>
     <jitter>0</jitter>
     <bitRate>64000</bitRate>
     <roundTripTime>66</roundTripTime>
  </txAudio>
  <rxVideo role="main">
     <codec>h264</codec>
     <width>768</width>
     <height>448</height>
     <frameRate>29.7</frameRate>
     <bitRate>544603</bitRate>
```

```
      <packetLossPercentage>0.0</packetLossPercentage>
      <jitter>2</jitter>
   </rxVideo>
   <txVideo role="main">
      <codec>h264</codec>
      <width>1280</width>
      <height>720</height>
      <frameRate>29.9</frameRate>
      <bitRate>1098095</bitRate>
      <packetLossPercentage>0.0</packetLossPercentage>
      <jitter>11</jitter>
      <roundTripTime>60</roundTripTime>
   </txVideo>
   <rxVideo role="presentation">
      <codec>h264</codec>
      <width>1280</width>
      <height>768</height>
      <frameRate>15.0</frameRate>
      <bitRate>3433</bitRate>
      <packetLossPercentage>0.0</packetLossPercentage>
      <jitter>2</jitter>
   </rxVideo>
   <activeControl>
      <encrypted>true</encrypted>
      <remoteSubscriptions>
         <capabilities />
         <conferenceInfo />
         <selfInfo />
         <speakerInfo />
      </remoteSubscriptions>
      <localSubscriptions>
         <capabilities />
         <conferenceInfo />
      </localSubscriptions>
   </activeControl>
</status>
```

Additionally, a “<sipCallId>” value may be included; this will be present if the call leg corresponds to a SIP connection, and will be the global unique “Call-ID” value from the SIP protocol headers.

If there is an active video stream from the Call Bridge to the remote party, there will be a “layout” value showing the actual layout currently in use for that call leg, either because of a specific choice on the part of the user (for web app call legs) or because of a coSpace default or call leg override.

Appendix C Additional Multiparty Licensing Information

C.1 /system/licensing Information

When information is retrieved from the /system/licensing node, its structure follows the form:

```
<licensing>
  <features>
    <callBridge>
      <status>activated</status>
      <expiry>2100-Jan-01</expiry>
    </callBridge>
    <webBridge>
      <status>activated</status>
      <expiry>2100-Jan-01</expiry>
    </webBridge>
    <turn>
      <status>activated</status>
      <expiry>2100-Jan-01</expiry>
    </turn>
    <ldap>
      <status>activated</status>
      <expiry>2100-Jan-01</expiry>
    </ldap>
    <branding>
      <status>activated</status>
      <expiry>2100-Jan-01</expiry>
      <level>whiteLabel</level>
    </branding>
    <recording>
      <status>activated</status>
      <expiry>2100-Jan-01</expiry>
      <limit>30</limit>
    </recording>
    <personal>
      <status>noLicense</status>
    </personal>
    <shared>
      <status>noLicense</status>
    </shared>
    <capacityUnits>
      <status>noLicense</status>
    </capacityUnits>
  </features>
</licensing>
```

C.2 /system/multipartyLicensing Information

When information is retrieved from the /system/multipartyLicensing node, its structure follows the form:

```
<multipartyLicensing>
   <timestamp>2016-07-20T14:22:17Z</timestamp>
   <personalLicenseLimit>0</personalLicenseLimit>
   <sharedLicenseLimit>0</sharedLicenseLimit>
   <capacityUnitLimit>0</capacityUnitLimit>
   <users>545</users>
   <personalLicenses>0</personalLicenses>
   <participantsActive>0</participantsActive>
   <callsActive>0</callsActive>
   <weightedCallsActive>0.000</weightedCallsActive>
   <callsWithoutPersonalLicense>0</callsWithoutPersonalLicense>
   <weightedCallsWithoutPersonalLicense>0.000</weighted
                       CallsWithout>PersonalLicense>
   <capacityUnitUsage>0.000</capacityUnitUsage>
   <capacityUnitUsageWithoutPersonalLicense>0.000</capacityUnit
                       UsageWithoutPersonalLicense>
</multipartyLicensing>
```

Appendix D Installing and using RESTer with the API

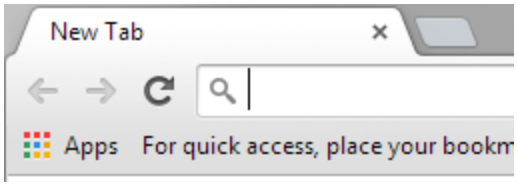
This appendix provides an example of installing and using the API tool, RESTer. Other tools are available, see Section 3.4.

Note: From version 2.9, you can use the API without the need of a third-party application. You can simply perform API configuration via the Web Admin user interface. For more information, see Section 3.3.

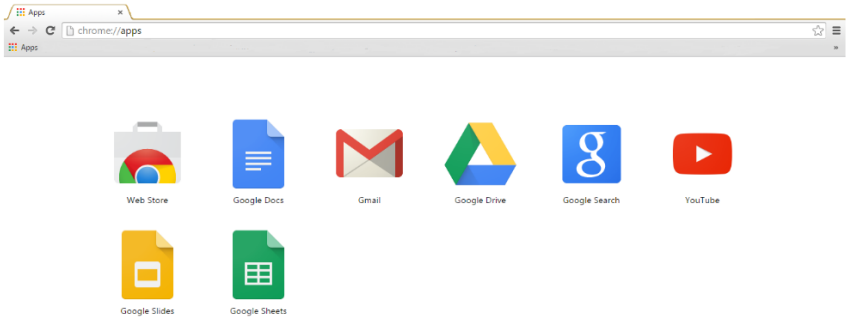
Ensure you have an account with API access as described earlier.

D.1 Installing RESTer

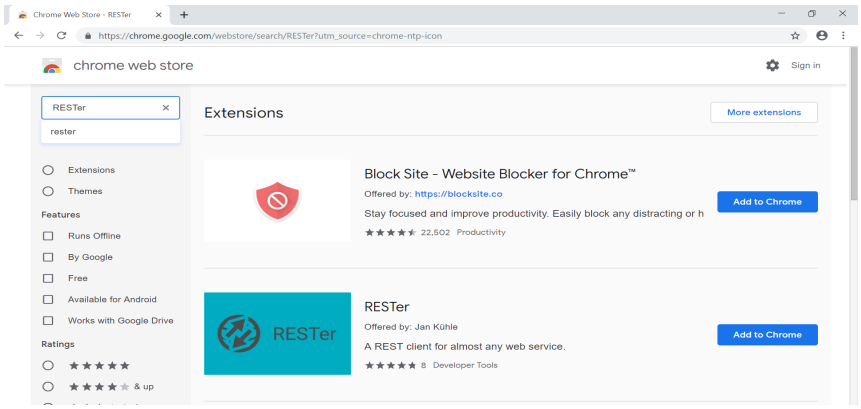
1. Open Google Chrome as your browser and in the top left-hand corner of the browser window select the Apps icon. Or click: https://chrome.google.com/webstore/category/extensions in a Chome browser window to skip to step 3.



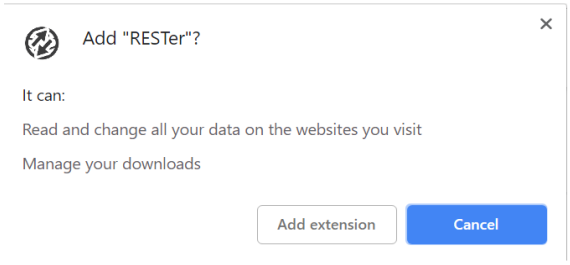
2. Click on Web Store to open the Chrome App Store and select **Extensions**.



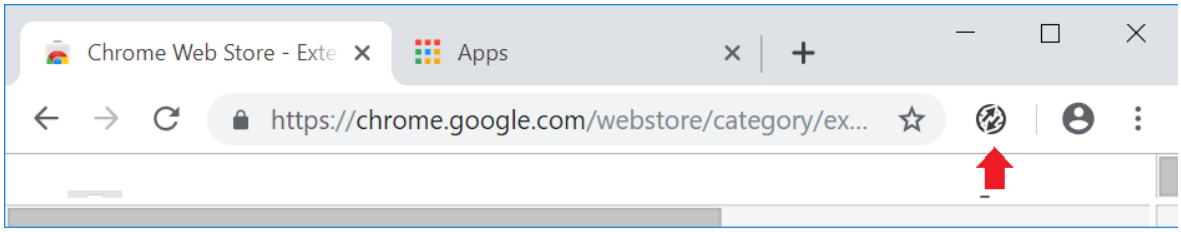
3. Enter RESTer in the Search field and press Enter, then locate the RESTer extension in the results and click **Add to Chrome**.



4. Click **Add extension** to continue with the installation.



5. Once installed, a confirmation dialog displays and the RESTer icon appears in the menu bar.



D.1.1 Add/Confirm SSL Exception

RESTer will follow the same SSL certificate verification and exception rules that your browser is configured for. If the URL for your webadmin site will not pass certificate validation or does not have an exception saved, RESTer will reject all requests. To check or add an exception:

1. In a Chrome window, open the URL to your Meeting Server webadmin interface, making sure you use HTTPS and the Port number (if listening on non-standard port). Example: **`https://cms.lab:445`**
2. If Chrome loads the Meeting Server web page without error you are OK to proceed.

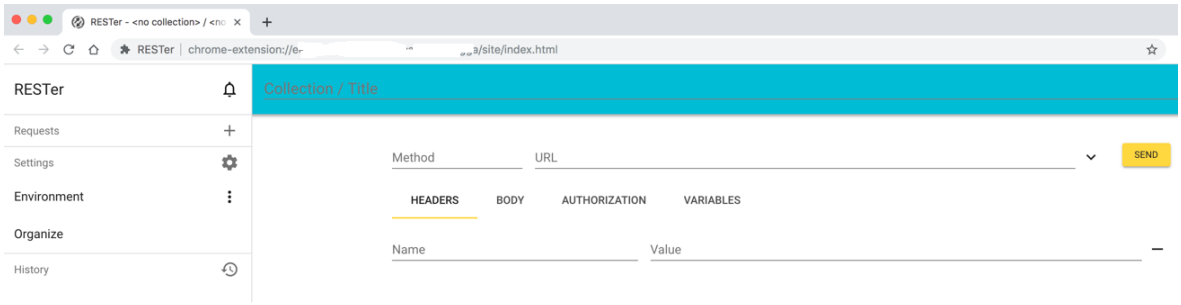
   or

3. If Chrome shows a Privacy Error, click **Advanced** and then click the link to proceed to the address.

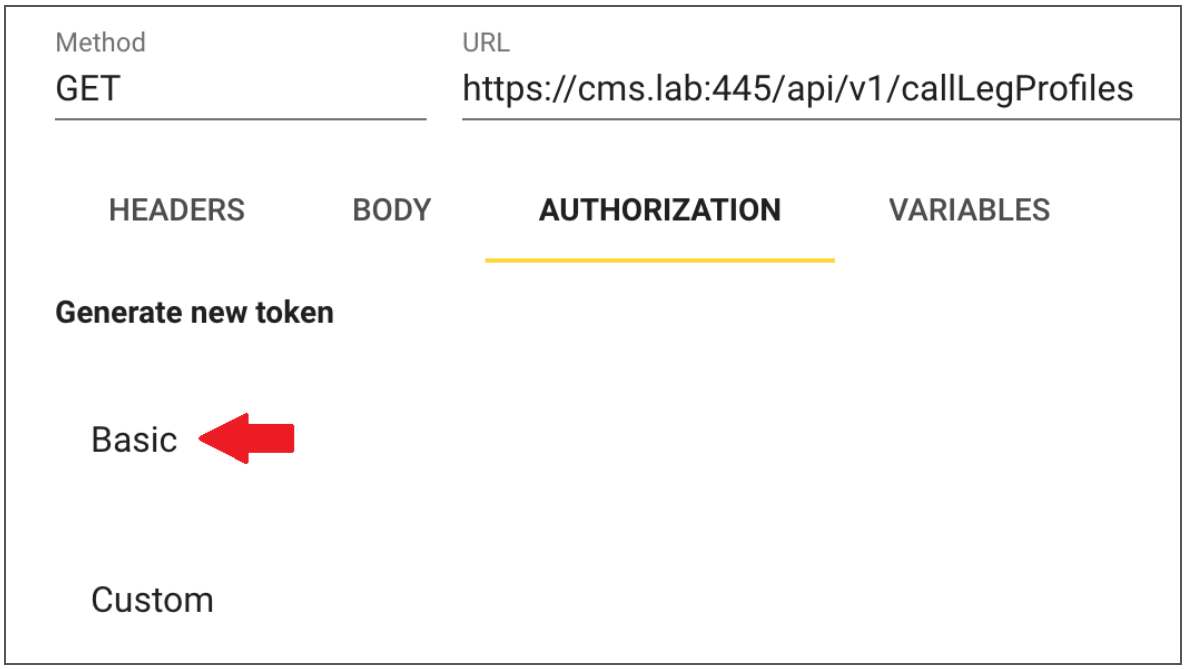
D.1.2 Generate a Basic Auth header

Your requests will need to authenticate to the server. RESTer can generate and save an authorization header so this step does not have to be repeated for each command.

1. If not already open, click on the RESTer icon in the browser menu bar to open the tool.



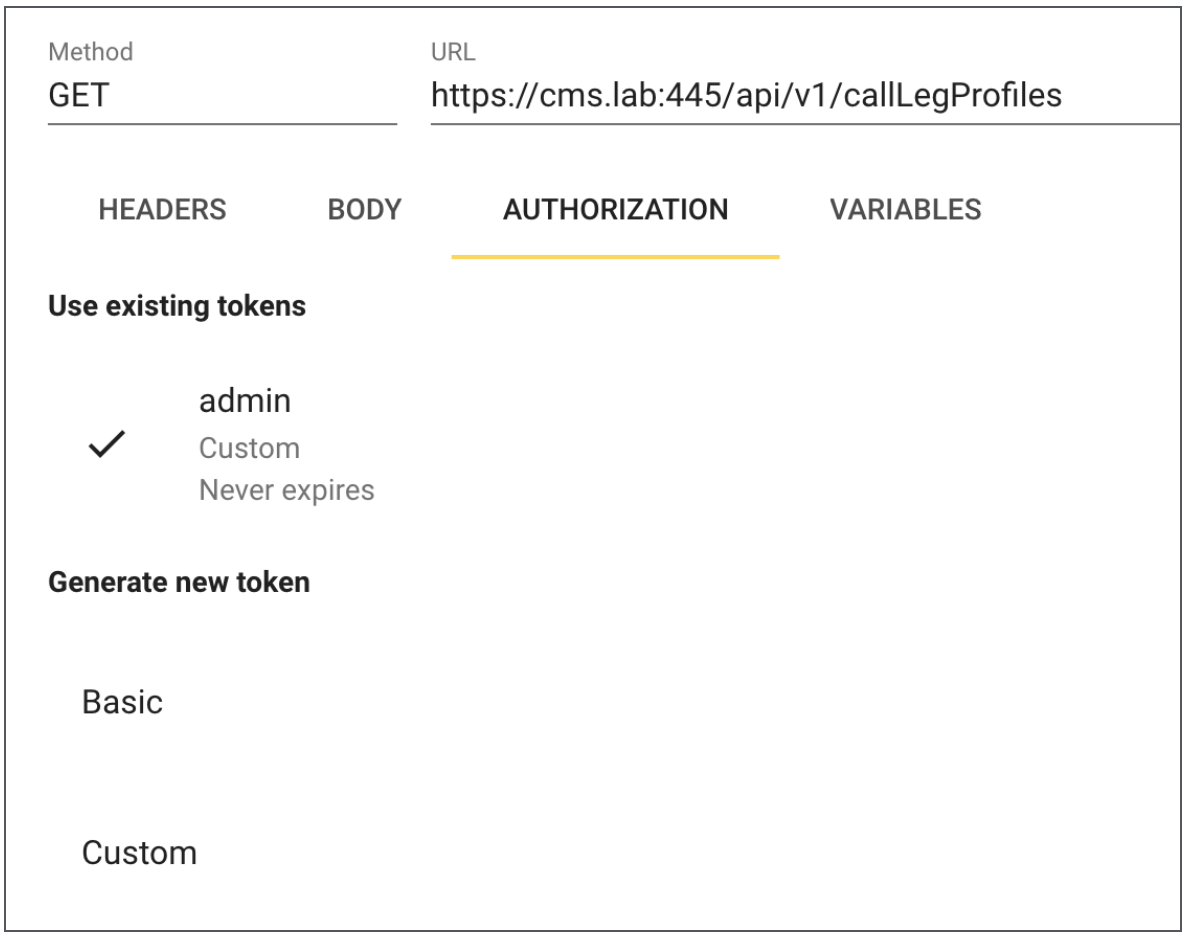
2. Click the **Authorization** tab, then click **Basic**.



3. Enter an appropriate set of credentials for your server in the pop-up that displays, and click **Save**.

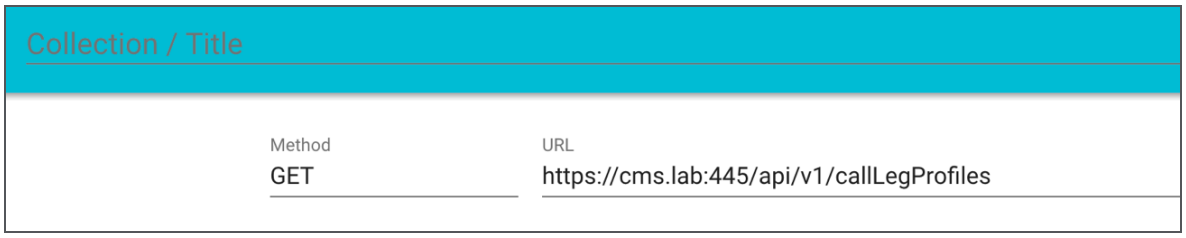


4. The tab will update to show the **Authorization** tab is now using the saved credential.



D.1.3 GET command example

1. Open a RESTer tab, if not already open. If this is a new session, follow the steps in Section D.1.2 to set an Authorization header. Confirm by clicking on the **Authentication** header and ensure it is set to use the existing credentials. (See screen in Step 4 above.)
2. In the RESTer tab, click **Method** and select **GET** from the drop-down list.
3. Click the URL field and enter the full URL to the API object you wish to query including your webadmin address, port, and the API path, in this example we will retrieve information on callLegProfiles:



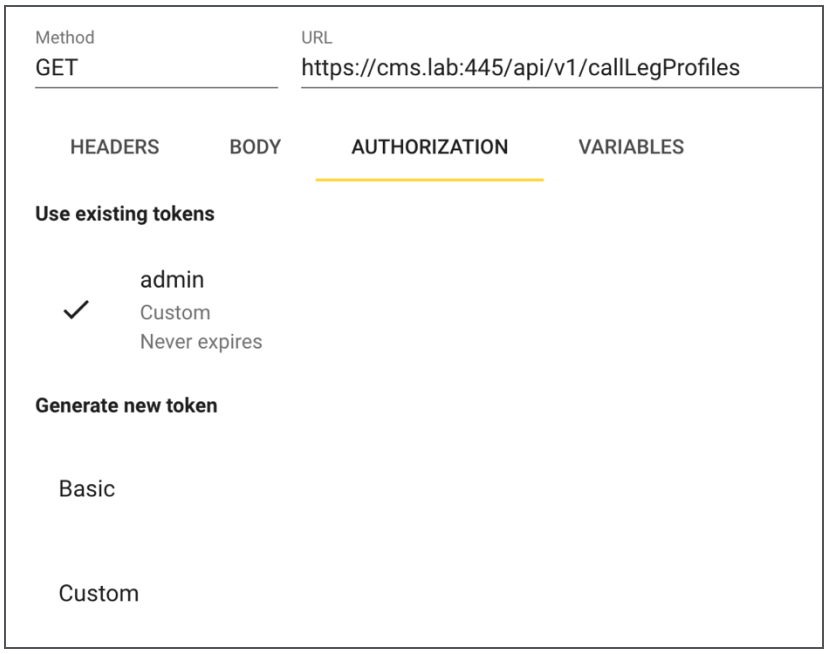
4. Click **Send** to send your request, and the response will appear at the bottom of the page.



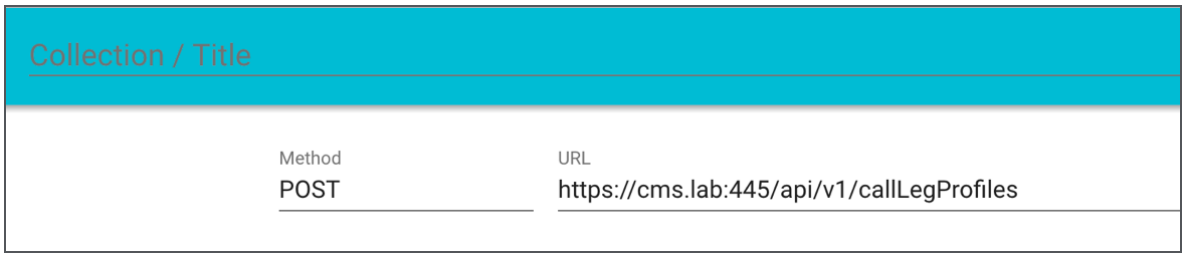
5. The tool automatically detects and formats the response as XML.

D.1.4 POST command example

1. Open a RESTer tab, if not already open. If this is a new session, follow the steps in Section D.1.2 to set an Authorization header. Confirm by clicking on the **Authentication** header and ensure it is set to use the existing credentials.

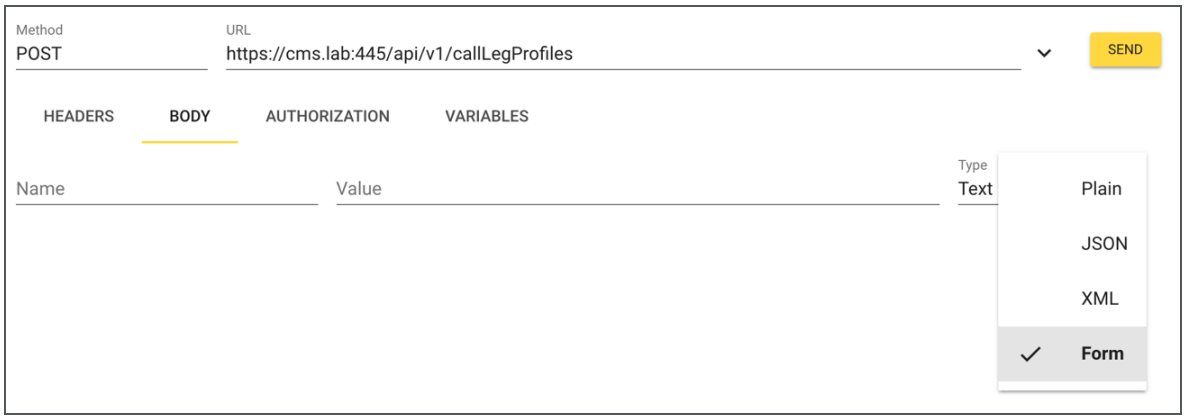


2. In the RESTer tab, click **Method** and select **POST** from the drop-down list.
3. Click the URL field and enter the full URL to the API object you wish to query including your webadmin address, port, and the API path, in this example we will create a new callLegProfile:

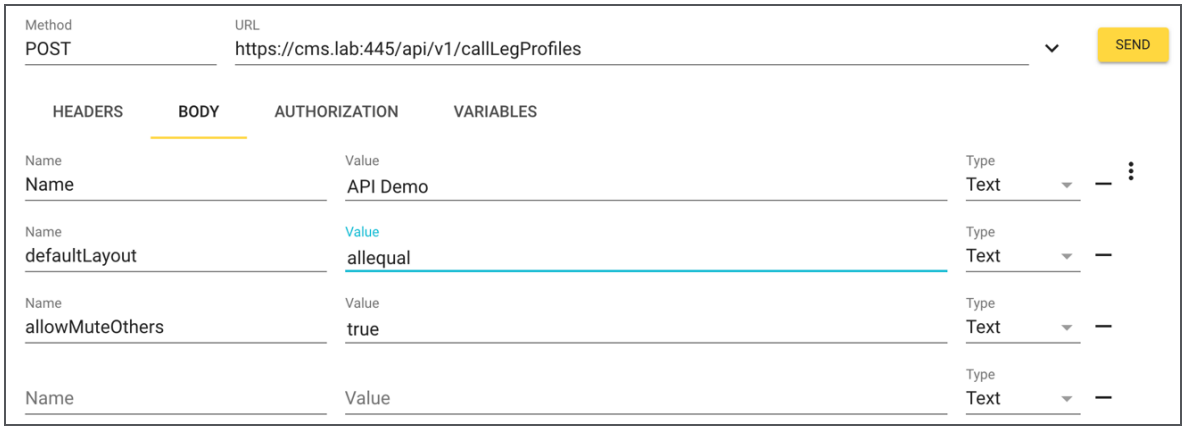


4. To send data with a POST request (optional), you must populate the **Body** tab with the data you wish to send with the request.

   Click **Body** and click the three dot drop-down on the right-hand side of the window. Select **Form** from the drop-down list:



5. Fill in the **Name** and **Value** fields for the properties you are setting with this request. See Section 8.4.3 for available settings.

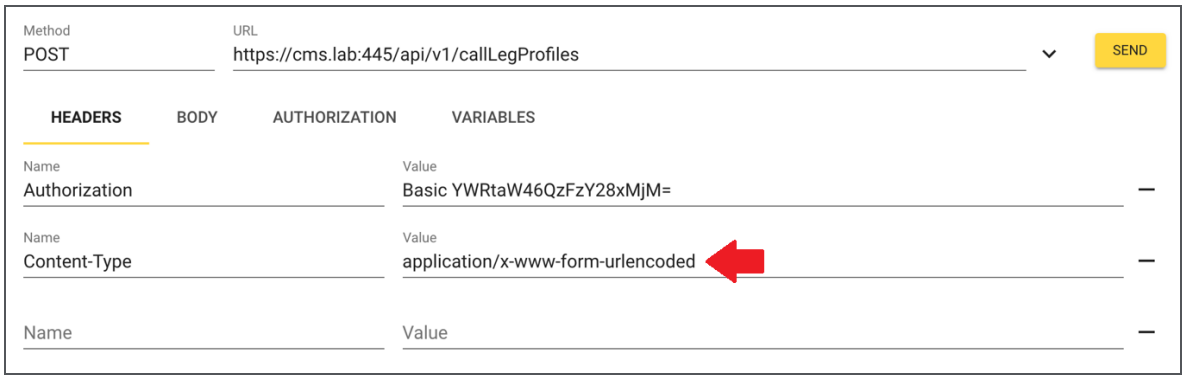


6. The request must include a header for the correct content type. To do this, click the **Headers** Tab and enter a new header as **Content-Type : application/x-www-form-urlencoded**.

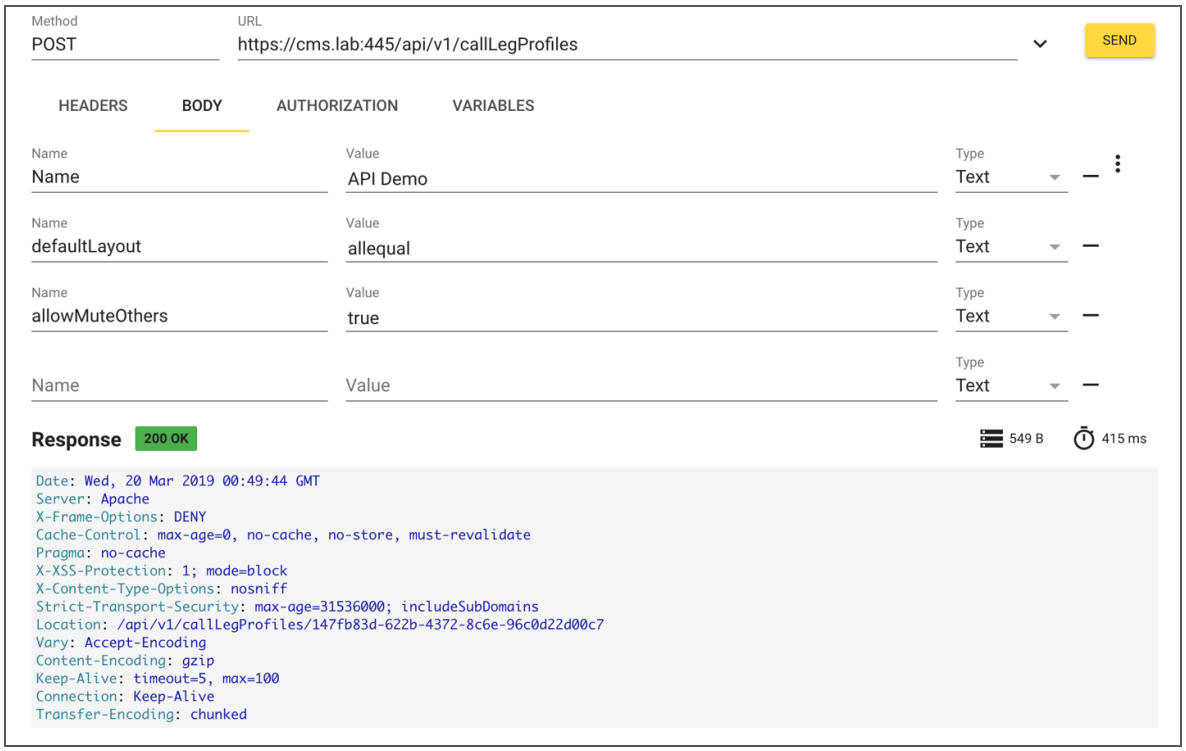
   Failure to do so will result in an HTTP 502 error when you send your request. However, RESTer will detect what you are trying to do and offer to set the header for you for future requests. Click on the highlighted warning banner shown and RESTer will automatically add the required Content Header to your request.



7. The correct Content Header is added to your request:



8. Click **Send** to send your request. The response headers and body are shown at the bottom of the page. Note the **Location** header includes the path and ID to the object created by the request.



D.1.5 PUT command example

Completing a PUT command in RESTer follows the same steps shown for a POST command except with a different HTTP method. Follow the POST example in the previous section and use the PUT option instead of POST when setting the Method setting.

Appendix E API methods presented by Web Bridge 3

Version 2.9 introduces new API methods to retrieve information specifically for the Web Bridge 3. These new API methods are not found on the usual Meeting Server API; they are supported on the API presented by Web Bridge 3. This API is used by the web app that runs in the browser to communicate with the Web Bridge 3. These methods are intended for use by administrators for diagnostic purposes.

For example, if Web Bridge 3 is running on: **`join.meeting.space`**, these API methods are on **`https://join.meeting.space/api/bridge/info`**

The new methods are:

- GET on **`/api/bridge/info`** returns an identifier for this Web Bridge 3.
- GET on **`/api/v1/load`** returns an identifier for this Web Bridge 3 (for legacy use).
- GET on **`/api/bridge/callbridges`** returns information on the current Call Bridge connections to this Web Bridge 3.
- GET on **`/api/bridge/connections`** returns information on current HTTP connections being served by this Web Bridge 3.*
- GET on **`/api/bridge/websockets`** returns information on current websockets being served by this Web Bridge 3.*
- GET on **`/api/configuration`** returns the languages available for custom email invites and other non-confidential configuration information.

* requires authentication (the same authentication that a web app user would use to log in).

Cisco Legal Information

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

© 2023 Cisco Systems, Inc. All rights reserved.

Cisco Trademark

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)