Cisco Meeting Server Basic API Functions

Contents

Introduction

This document describes the four basic API (Application Program Interface) functions GET, POST, PUT, DELETE used on CMS (Cisco Meeting Server). As of CMS 2.9, the web admin GUI has an API menu available under the Configuration menu. This document reviews that new menu and also describes two different API tools: Poster and Postman and how to use them for CMS configuration.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document can be used with CMS 2.9 and later or with different API clients like Postman or Poster. These third-party tools are described in the API clients section of the document.

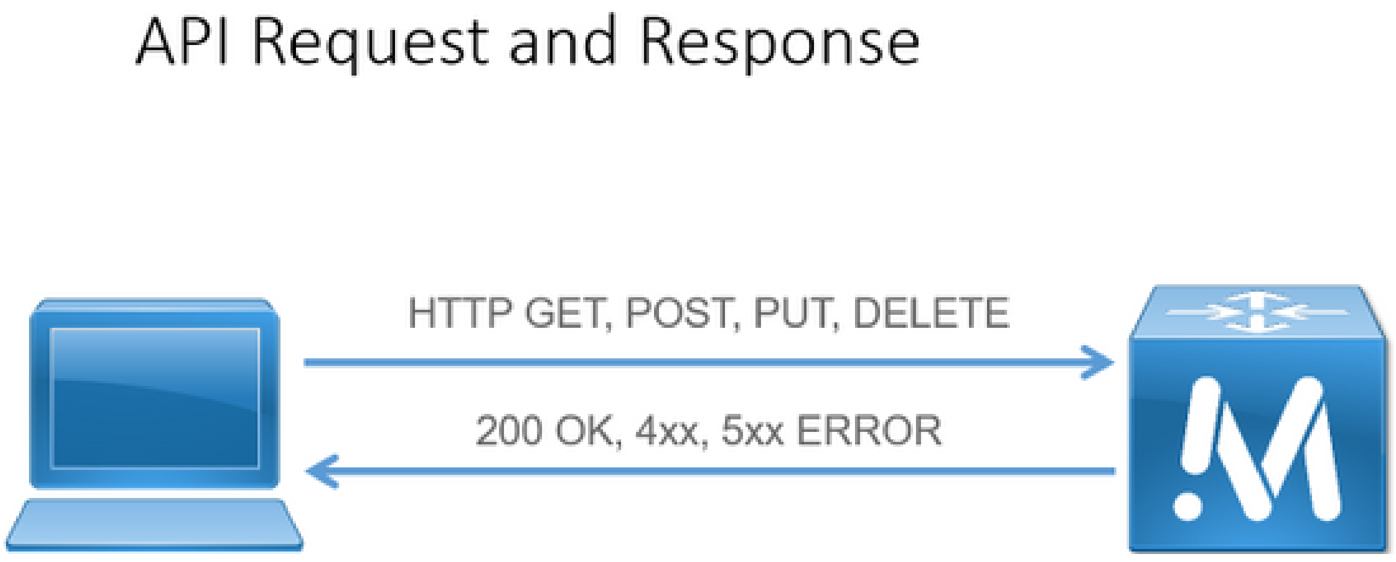
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

The API for the CMS is an extremely flexible way to configure many of its features. There are far too many API features to memorize or cover here, so be sure to reference the current API Reference documentation. As of the time of this writing, the current API Reference guides are available here.

API Request and Response

API communication is a request-response relationship between clients and servers. The client makes a request of the server. After handling the request (completing an action, or refusing to do so) a response is returned.



The four requests described in this article are:

1. GET - Retrieves existing information
2. POST - Creates new information
3. PUT - Modifies existing information
4. DELETE - Deletes existing information.

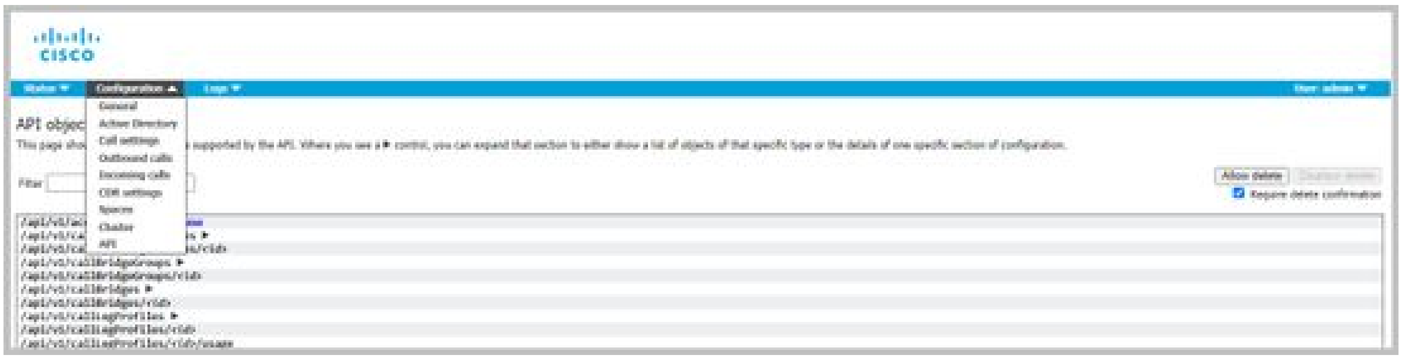
These are the basic API requests used to configure CMS.

The most common response is a 200 OK. Other responses are 4xx and 5xx, which are error responses.

Configure

CMS 2.9 and Later

CMS 2.9 introduced a new API menu that makes it much easier for administrators to modify settings and fine tune settings in CMS. When using the menu, all available parameters are displayed, which makes it quick and easy to change settings and enable new features.



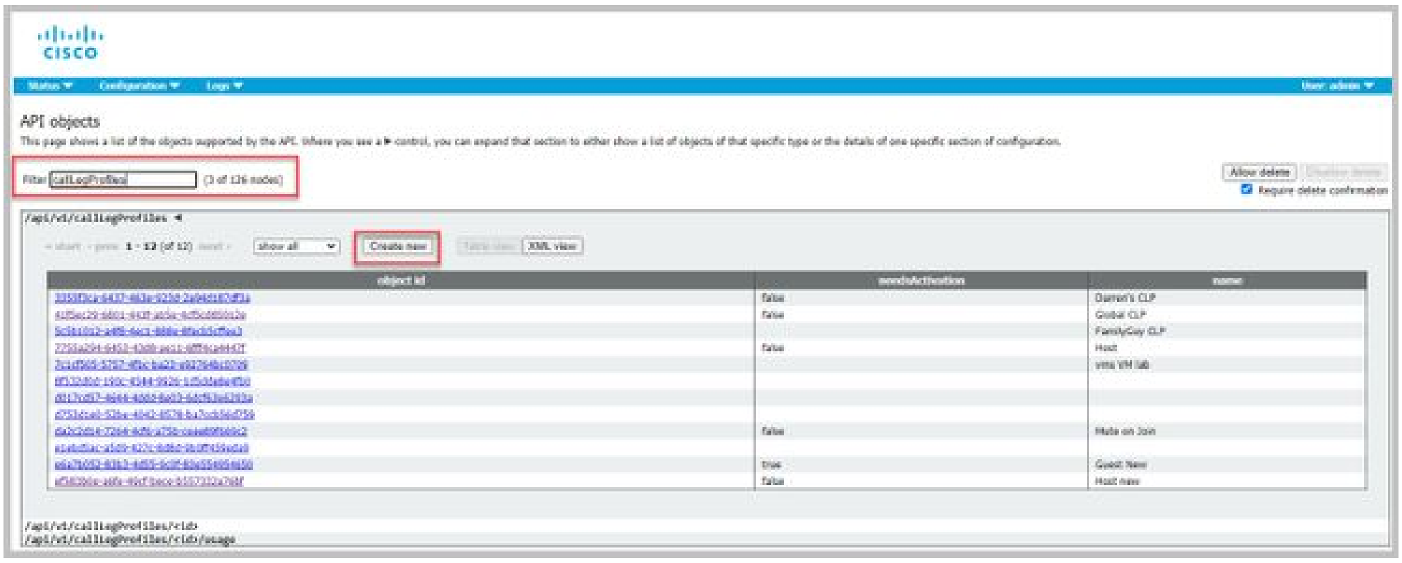
API objec

This page sho… supported by the API. Where you see a ▶ control, you can expand that section to either show a list of objects of that specific type or the details of one specific section of configuration.

```
/api/v1/ac…
/api/v1/ca… ▶
/api/v1/ca…s/<id>
/api/v1/callBridgeGroups ▶
/api/v1/callBridgeGroups/<id>
/api/v1/callBridges ▶
/api/v1/callBridges/<id>
/api/v1/callLegProfiles ▶
/api/v1/callLegProfiles/<id>
/api/v1/callLegProfiles/<id>/usage
```

Configure an API Object

Once in the API menu, you can filter the API objects to what you are looking to edit/create and then click the black arrow next to the object to make those changes. This example shows how to look up callLegProfiles and create a new callLegProfile.



API objects

This page shows a list of the objects supported by the API. Where you see a ▶ control, you can expand that section to either show a list of objects of that specific type or the details of one specific section of configuration.

Filter: callLegProfiles (3 of 126 nodes)

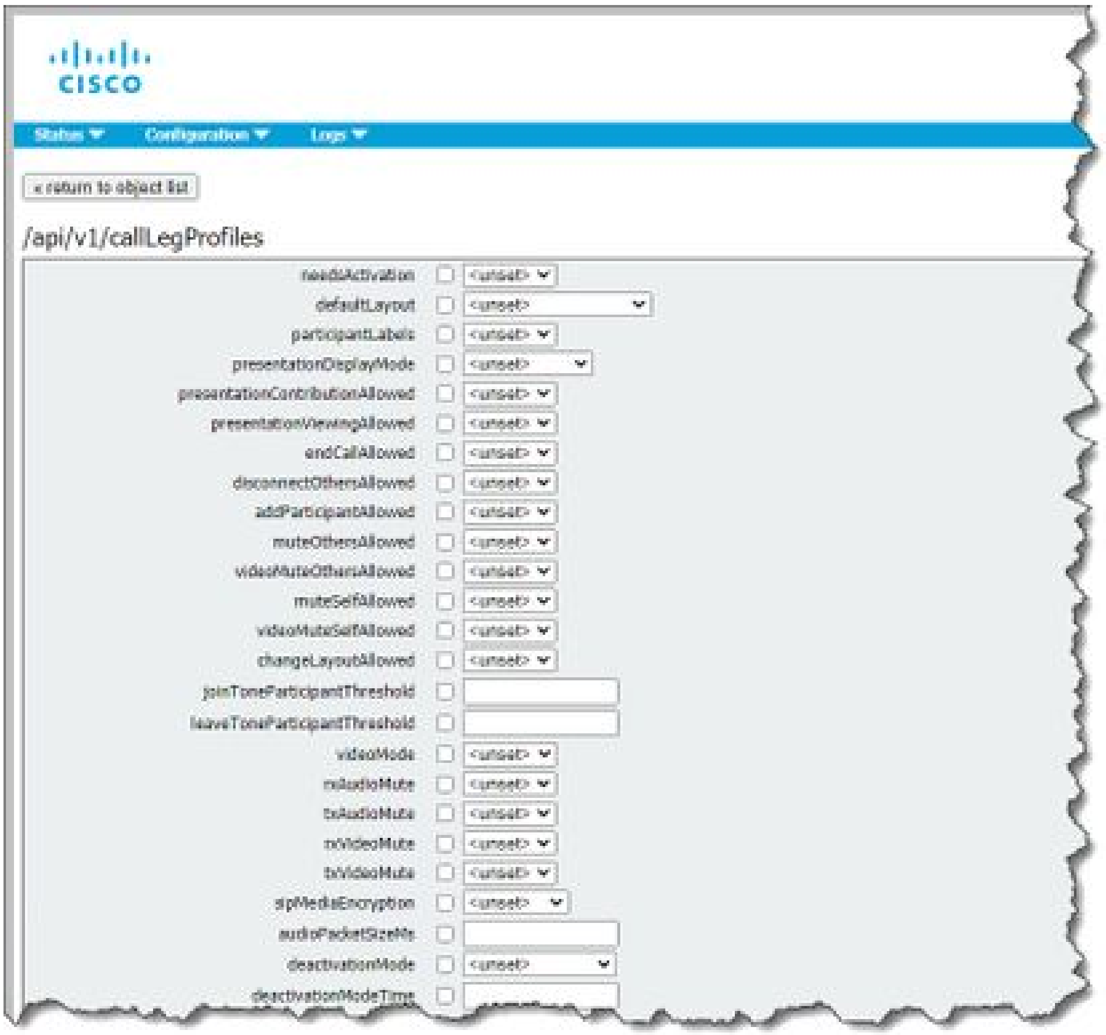
/api/v1/callLegProfiles ◀

1 - 12 (of 12)

object id	needsActivation	name
3359f3ca-6437-463a-9250-2a94d187df1a	false	Darren's CLP
41f5ec29-b601-443f-ab0e-4cf5cd85012e	false	Global CLP
5c5b1012-a4f8-4ec1-888e-8fac65cf9ea3		FamilyGuy CLP
7755a204-6453-43db-ac11-6fff6ca4447f	false	Host
7c1df565-5757-4fbc-ba23-a93764b10709		vms VM lab
8f532d0d-190c-4544-9926-1f5dda8e4f0d		
d017cd57-4644-4002-8a03-6dcf63e6293a		
d751dca0-52be-4042-8578-ba7ccb56d759		
da2c2d14-7264-4df6-a750-ceaa69f9b0c2	false	Mute on Join
e1a6d5ac-a509-427c-8d60-9b0ff459ada8		
e6a7b052-83b3-4d55-9c0f-83e55495465b	true	Guest New
ef5839de-a6fa-49cf-bece-b557332a768f	false	Host new

```
/api/v1/callLegProfiles/<id>
/api/v1/callLegProfiles/<id>/usage
```

When you click **Create New**, you are presented with this screen which displays all available parameters for CallLegProfiles. When you hover over a particular parameter, you will get a popup that shows the purpose of each option.

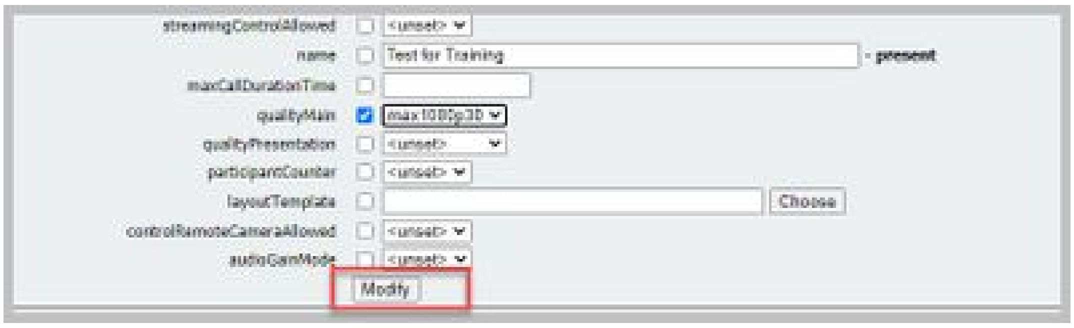


/api/v1/callLegProfiles

needsActivation	☐	<unset>
defaultLayout	☐	<unset>
participantLabels	☐	<unset>
presentationDisplayMode	☐	<unset>
presentationContributionAllowed	☐	<unset>
presentationViewingAllowed	☐	<unset>
endCallAllowed	☐	<unset>
disconnectOthersAllowed	☐	<unset>
addParticipantAllowed	☐	<unset>
muteOthersAllowed	☐	<unset>
videoMuteOthersAllowed	☐	<unset>
muteSelfAllowed	☐	<unset>
videoMuteSelfAllowed	☐	<unset>
changeLayoutAllowed	☐	<unset>
joinToneParticipantThreshold	☐	
leaveToneParticipantThreshold	☐	
videoMode	☐	<unset>
rxAudioMute	☐	<unset>
txAudioMute	☐	<unset>
rxVideoMute	☐	<unset>
txVideoMute	☐	<unset>
sipMediaEncryption	☐	<unset>
audioPacketSizeMs	☐	
deactivationMode	☐	<unset>
deactivationModeTime	☐	

Modify an API Object That is Already Created

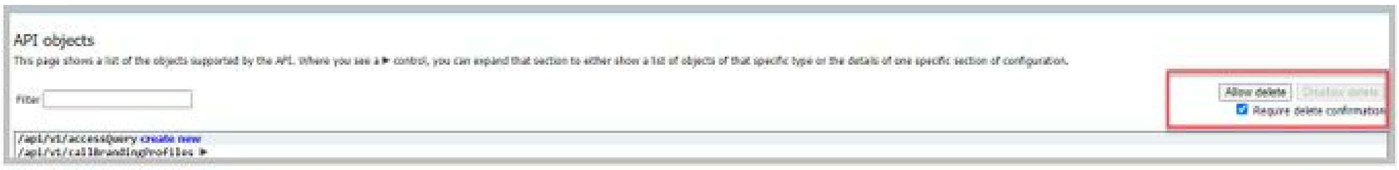
When you change settings in an object, you will see the Modify button at the bottom. This is the same as a PUT from the third-party tools.



streamingControlAllowed	☐	<unset>
name	☐	Test for Training - present
maxCallDurationTime	☐	
qualityMain	☑	max1080p30
qualityPresentation	☐	<unset>
participantCounter	☐	<unset>
layoutTemplate	☐	Choose
controlRemoteCameraAllowed	☐	<unset>
audioGainMode	☐	<unset>

Delete an API Object from the API Menu

In order to delete an object, on the main Object List page, you can enable the ability to delete items. Click **Allow delete** in order to enable the option to delete, as shown in this example:



API objects

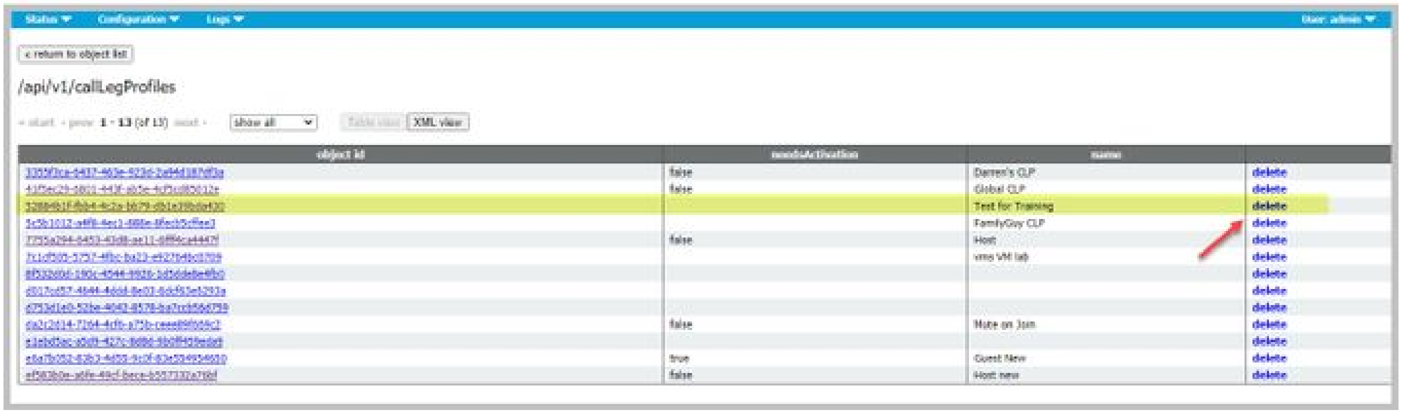
This page shows a list of the objects supported by the API. Where you see a ► control, you can expand that section to either show a list of objects of that specific type or the details of one specific section of configuration.

Filter

Allow delete | Disallow delete

☑ Require delete confirmation

```
/api/v1/accessQuery create new
/api/v1/callBrandingProfiles ►
```



/api/v1/callLegProfiles

« start ‹ prev 1 - 13 (of 13) next › show all | Table view | XML view

object id	needsActivation	name	
3355f3ca-6437-463e-923d-2a94d187df3a	false	Darren's CLP	delete
41f5ec29-6801-443f-ab5e-4cf5cd85012e	false	Global CLP	delete
3288b61f-fbb4-4c2a-bb79-db1e39bda630		Test for Training	delete
5c5b1012-a4f8-4ec1-888e-8fecb5cffee3		FamilyGuy CLP	delete
7755a294-0453-43d8-ae11-0ff6ca4447f	false	Host	delete
7c1df505-5757-4fbc-ba23-e427b4bc0709		vms VM lab	delete
8f532d0d-180c-4544-9926-1d56de8e4fb0			delete
d017cd57-4b44-4ddd-8e03-6dcf61eb293a			delete
d753d1e0-52be-4042-8578-ba7ccb56d759			delete
da2c2d14-7264-4cfb-a75b-ceee89fb09c2	false	Mute on Join	delete
e1abd5ac-a5d9-427c-8d8d-9b0ff459eda9			delete
e6a7b052-83b3-4d55-9c0f-83e594954650	true	Guest New	delete
ef583b0e-a6fe-49cf-bece-b5577332a716f	false	Host new	delete

API Requests Explained (using 3rd party tools)

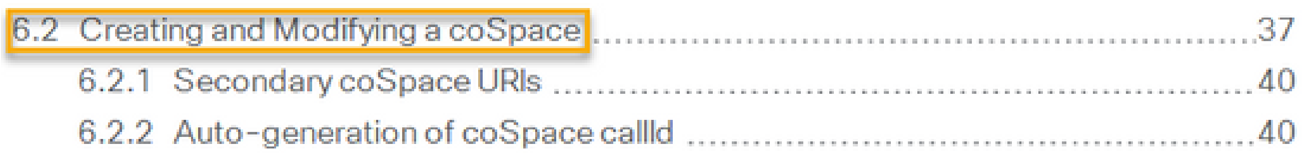
The four basic requests are explained through a configuration example.

HTTP POST

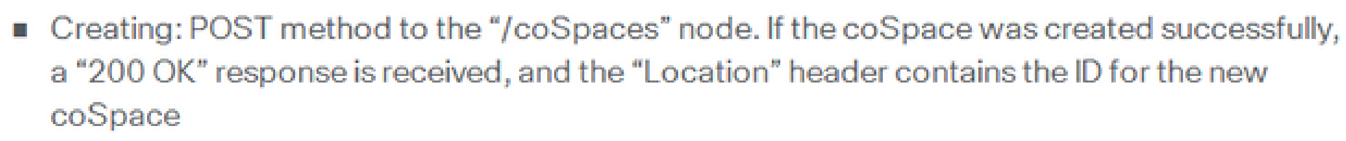
Step 1. Use **POST** to create an object.

In this example, a CMS Space is created using this request. In order to create the Space via API, consult the API documentation. For this example I used the CMS 2.4 API guide, but you should use the latest API guides, found here

Section 6.2 has information on how to create and modify a cospace.



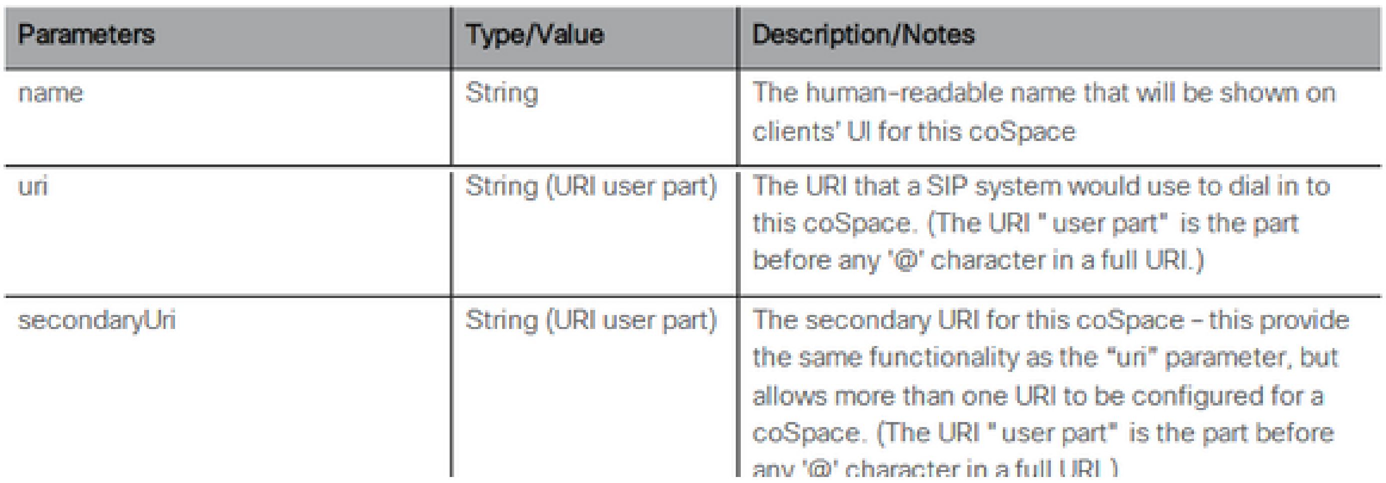
The first sentence says that to create a Space, you need to send a post to /coSpaces. Then, it says the ID of the Space will be in the Location header of the 200 OK. Great, you now know how to create a Space. You just send a **POST** to **https://<WebAdminIP>/api/v1/coSpaces**.



- 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

Specify the parameters for the **POST**.

In section 6.2 of the documentation you see a table that lists all of the parameters you can use.



Parameters	Type/Value	Description/Notes
name	String	The human-readable name that will be shown on clients' UI for this coSpace
uri	String (URI user part)	The URI that a SIP system would use to dial in to this coSpace. (The URI " user part" is the part before any '@' character in a full URI.)
secondaryUri	String (URI user part)	The secondary URI for this coSpace – this provide the same functionality as the "uri" parameter, but allows more than one URI to be configured for a coSpace. (The URI " user part" is the part before any '@' character in a full URI.)

For example: Create a Space with the name **APITest** and a URI user part of **APITestURI**

The content type is **application/x-www-form-urlencoded** and the content is **name=APITest&uri=APITestURI**

When you add this parameters the request is complete, as shown in the image.

```
POST https://<WebAdminIP>/api/v1/coSpaces HTTP/1.1
Host: <WebAdminIP>
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive

name=APITest&uri=APITestURI
```

The image shows a response to the previous request.

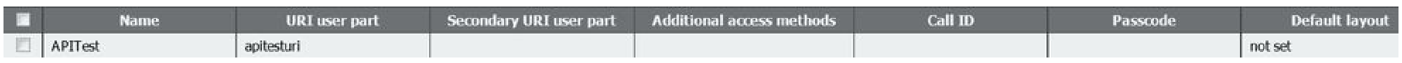
```
HTTP/1.1 200 OK
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Location: /api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae
Vary: Accept-Encoding
Content-Encoding: gzip
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
```

Notice the Location header in the response.

```
Location: /api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae
```

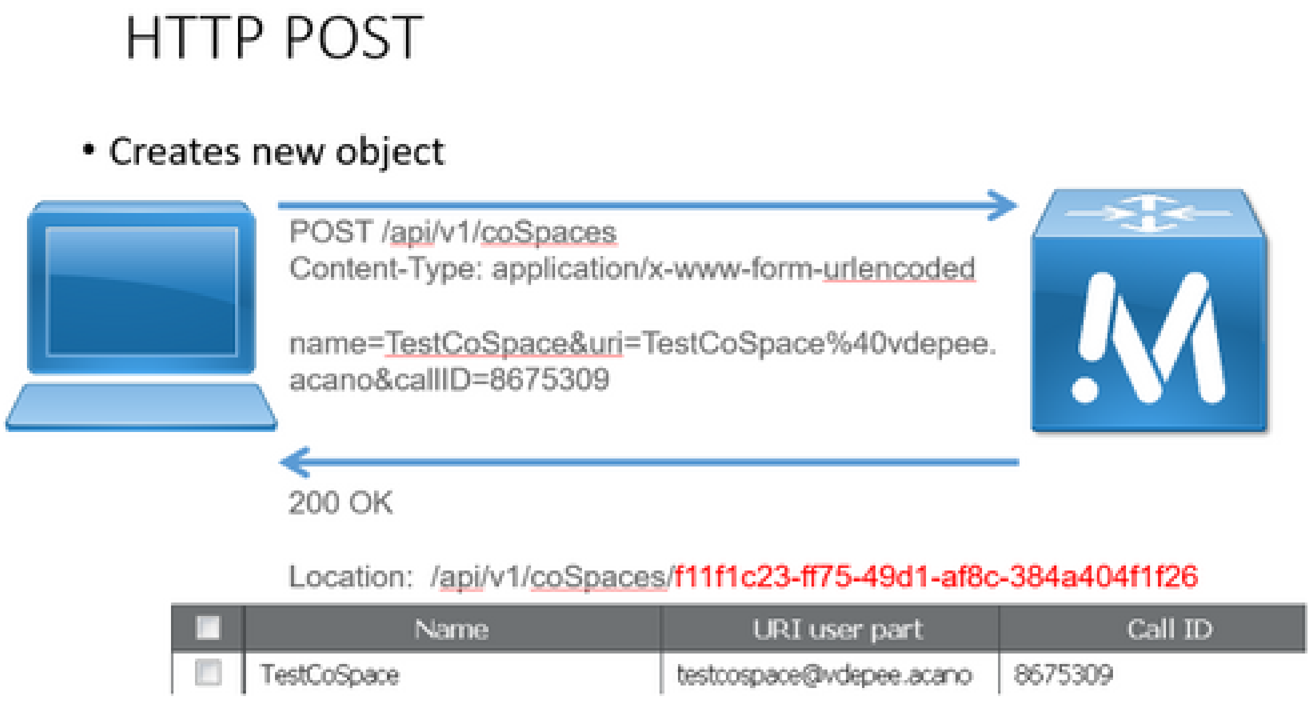
70ca0ed7-4e50-428c-b9ab-4e68faeb86ae is the ID of the new created Space. The ID is useful when you need to make future API requests that target the same Space.

The Space can be seen in the WebAdmin of CMS. Navigate to **Configuration > Spaces.**



☐	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout
☐	APITest	apitesturi					not set

The image summarizes the request **POST**.



HTTP POST

- Creates new object

```
POST /api/v1/coSpaces
Content-Type: application/x-www-form-urlencoded

name=TestCoSpace&uri=TestCoSpace%40vdepee.
acano&callID=8675309
```

```
200 OK

Location:  /api/v1/coSpaces/f11f1c23-ff75-49d1-af8c-384a404f1f26
```

☐	Name	URI user part	Call ID
☐	TestCoSpace	testcospace@vdepee.acano	8675309

HTTP GET

Step 2. After the Space has been created, pull the configuration for it.

Use the HTTP GET method for this purpose.

Use the ID for the Space created from the Location header. The URL is https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae. Perform a **GET** on this page.

Example GET request:

```
GET https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae HTTP/1.1
Host: <WebAdminIP>
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Cookie: session=logout
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive
```

Response for the GET request:

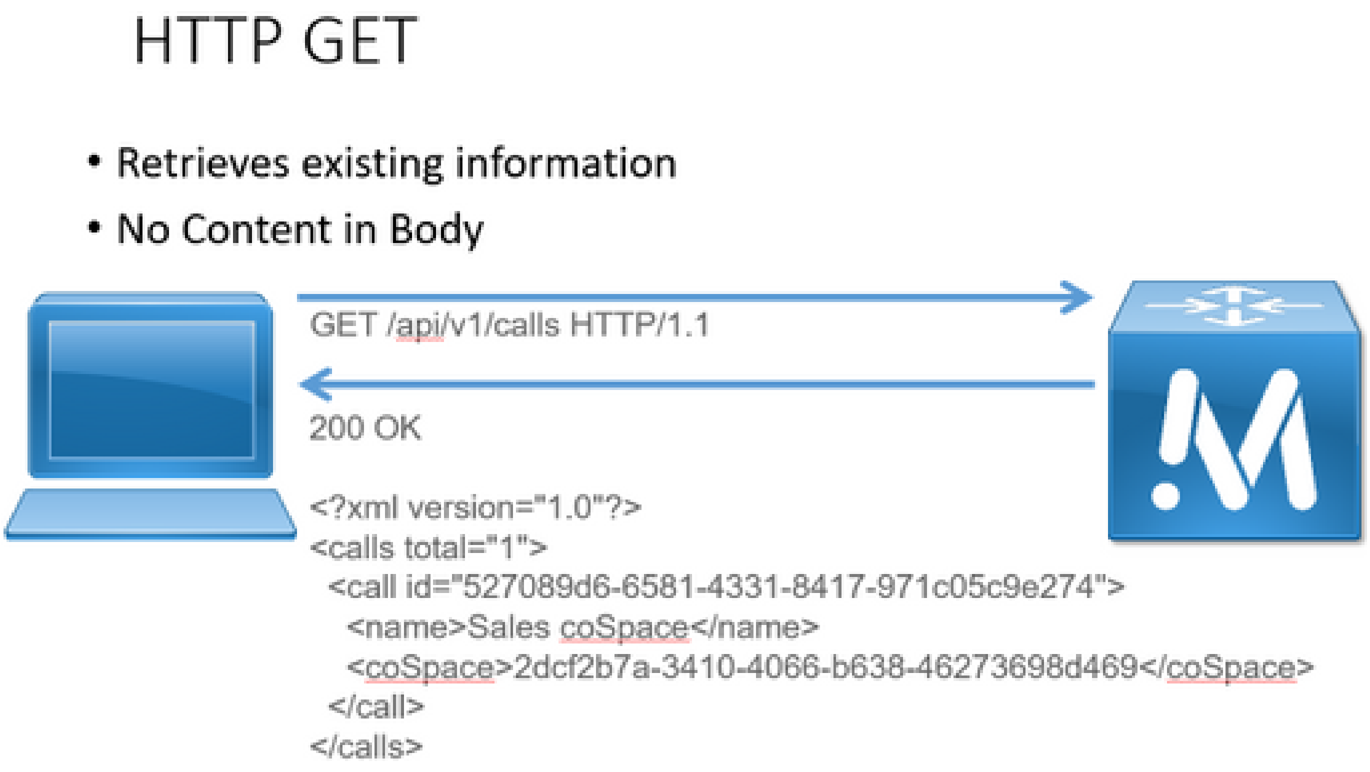
```
HTTP/1.1 200 OK
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Type: text/xml
Vary: Accept-Encoding
Content-Length: 159
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

<?xml version="1.0"?><coSpace id="70ca0ed7-4e50-428c-b9ab-
4e68faeb86ae"><name>APITest</name><autoGenerated>false</autoGenerated><uri>apitesturi</uri></coS
```

```
pace>
```

Note: The response is an XML encoded configuration of the Space.

The image summarizes the request **GET**.



HTTP GET

- Retrieves existing information
- No Content in Body

GET /api/v1/calls HTTP/1.1

200 OK

```
<?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>
```

HTTP PUT

Step 3. Make a change to the Space (if needed).

This example shows how to modify the space created. Imagine a secondary User portion needs to be added to the Space.

Refer to the API document. It tells the parameter needed to use that is: **secondaryUri**.

Add a URI of asdf. Write a request that looks similar to the request created for POST.

Example PUT request:

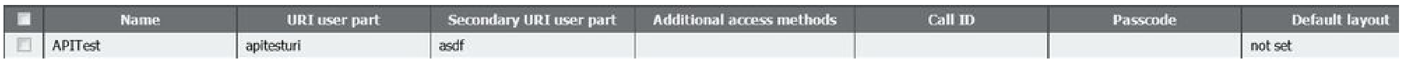
```
PUT https://172.18.105.244/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae HTTP/1.1
Host: 172.18.105.244
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 17
Cookie: session=b810c447daaeab6cdc6e019c
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive

secondaryUri=asdf
```

Response for the PUT request:

```
HTTP/1.1 200 OK
Date: Tue, 12 Apr 2016 19:11:02 GMT
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Length: 0
```

The changes can be seen in the WebAdmin of CMS. Navigate to **Configuration > Spaces.**

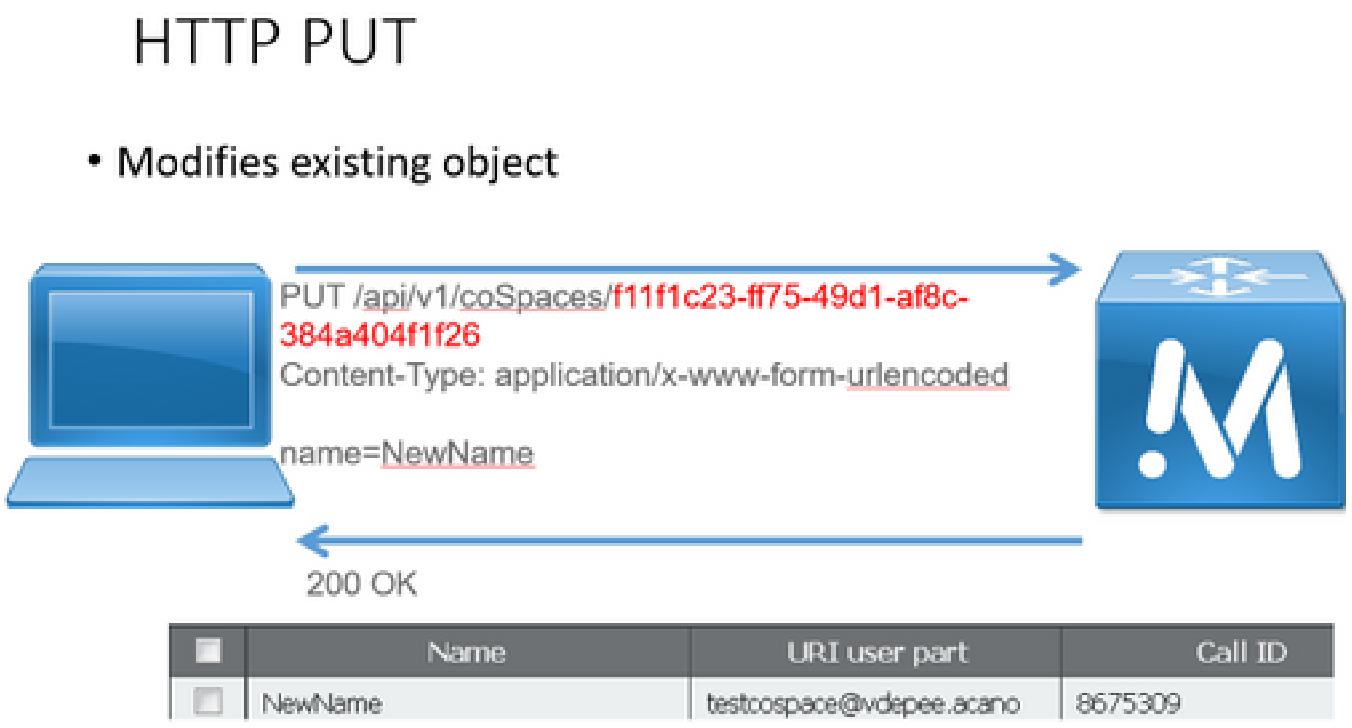


■	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout
☐	APITest	apitesturi	asdf				not set

and via a **GET**:

```
<?xml version="1.0"?><coSpace id="70ca0ed7-4e50-428c-b9ab-
4e68faeb86ae"><name>APITest</name><autoGenerated>false</autoGenerated><uri>apitesturi</uri><seco
ndaryUri>asdf</secondaryUri></coSpace>
```

The image summarizes the request **PUT**.



HTTP PUT

• Modifies existing object

```
PUT /api/v1/coSpaces/f11f1c23-ff75-49d1-af8c-
384a404f1f26
Content-Type: application/x-www-form-urlencoded

name=NewName
```

200 OK

■	Name	URI user part	Call ID
☐	NewName	testcospace@vdepee.acano	8675309

HTTP DELETE

Step 4. Delete the Space (if needed).

The **DELETE** method is similar to the **GET** method.

Example DELETE request:

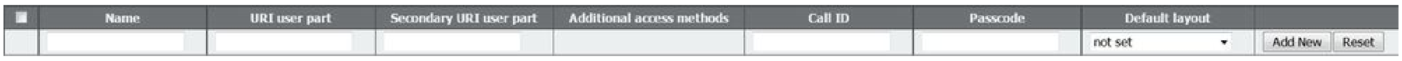
```
DELETE https://172.18.105.244/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae HTTP/1.1
Host: 172.18.105.244
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Cookie: session=4d13c7ebe739b662dc6e019c
Authorization: Basic YWRtaW46QzFzYzBDMXNjMA==
Connection: keep-alive
```

Response for the DELETErequest:

```
HTTP/1.1 200 OK
Date: Tue, 12 Apr 2016 19:16:37 GMT
Server: Apache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubDomains
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Length: 0
```

The changes can be seen in the WebAdmin of CMS. Navigate to **Configuration > Spaces.**

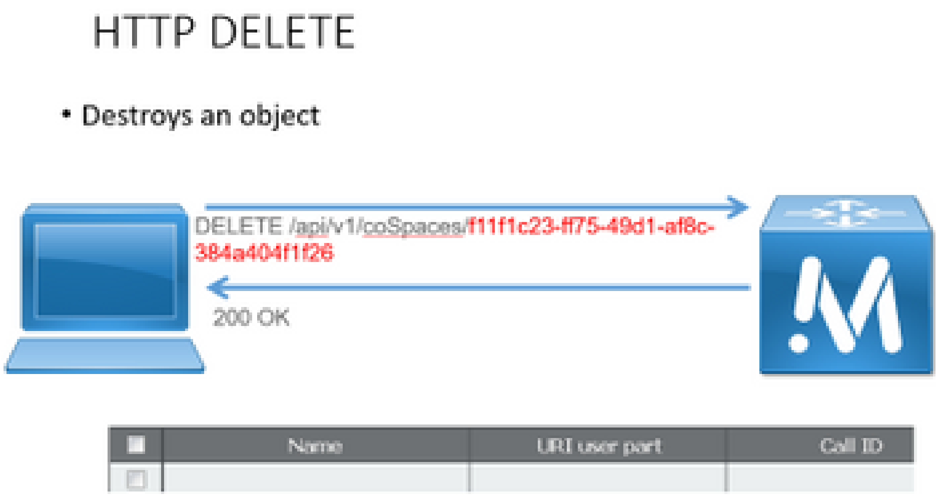


■	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout	
							not set	Add New Reset

and via a **GET**:

```
<?xml version="1.0"?><failureDetails><coSpaceDoesNotExist /></failureDetails>
```

The image summarizes the request **DELETE**.



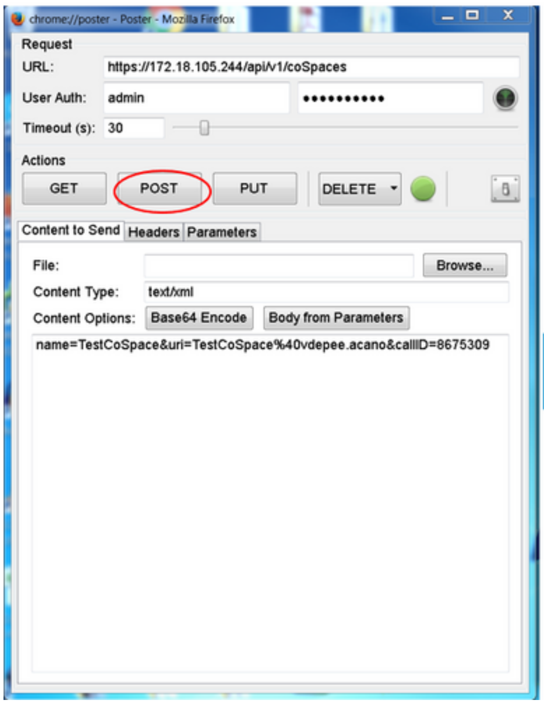
API Clients

POSTER

The top box in Poster is where you enter the URL for the requests.

The User Auth fields are where you enter the username and password in that order. Then, if you are doing a **GET** or a **DELETE**, choose the respective buttons. For example: click **GET** and a popup appears with your response. For **DELETE**, ensure **DELETE** is selected and click the green button.

Poster (Firefox)



chrome://poster - Poster - Mozilla Firefox

Request

URL: https://172.18.105.244/api/v1/coSpaces

User Auth: admin ••••••••••

Timeout (s): 30

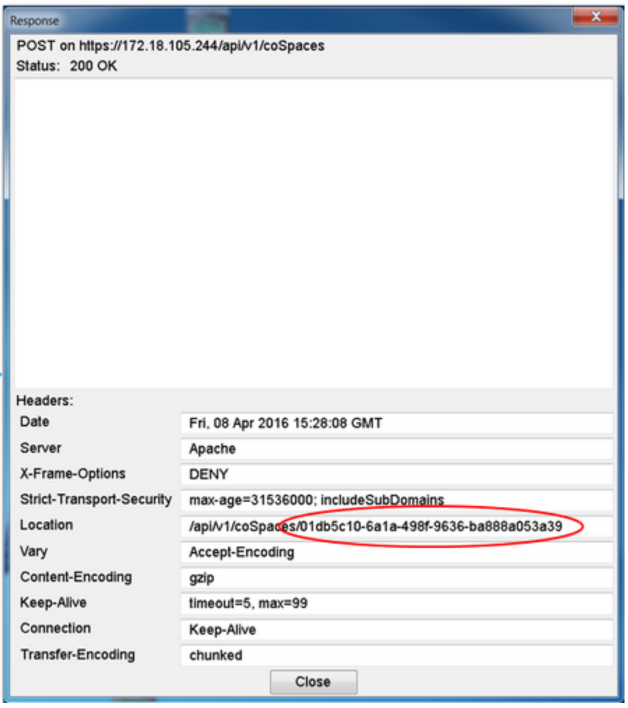
Actions

File:

Content Type: text/xml

Content Options:

```
name=TestCoSpace&uri=TestCoSpace%40vdepee.acano&callID=8675309
```



Response

POST on https://172.18.105.244/api/v1/coSpaces

Status: 200 OK

Headers:

Date	Fri, 08 Apr 2016 15:28:08 GMT
Server	Apache
X-Frame-Options	DENY
Strict-Transport-Security	max-age=31536000; includeSubDomains
Location	/api/v1/coSpaces/01db5c10-6a1a-498f-9636-ba888a053a39
Vary	Accept-Encoding
Content-Encoding	gzip
Keep-Alive	timeout=5, max=99
Connection	Keep-Alive
Transfer-Encoding	chunked

For **POST** and **PUT**, content needs to be added. Select the Parameters tab and add the names and values for your parameters. Then, go back to the Content to send button and choose **Body from Parameters**.

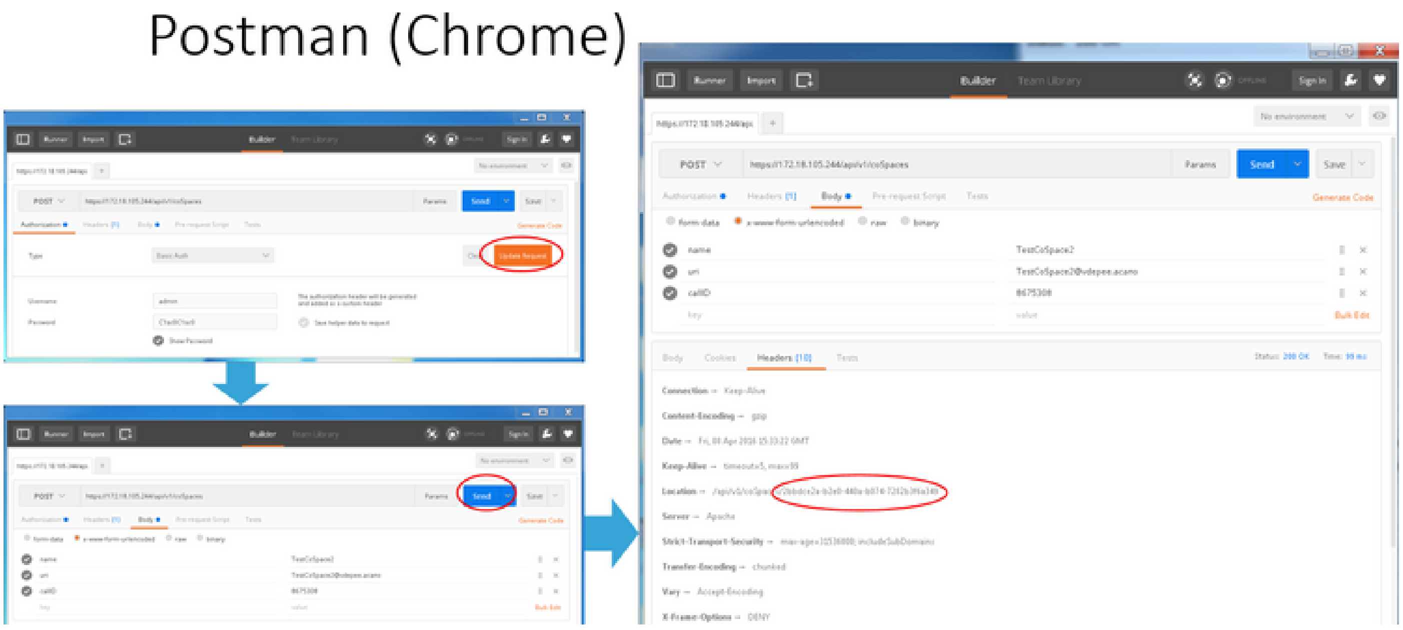
Send your POST and/or PUT.

POSTMAN

In Postman, in the top-left, choose the Method you would like to use from the drop-down box and enter the request URL.

For Authorization, choose **Basic-Auth** and enter your username and password. Then, choose **Update Request**. In the Headers tab you see an Authorization Header.

If your request is a POST/PUT, navigate to the Body tab, choose **x-www-form-urlencoded** and enter your parameters and values. When you are finished, choose **Send**.



POST https://172.18.105.244/api/v1/coSpaces

Authorization

Type: Basic Auth

Username: admin

Password: [illegible]

Show Password

Body: x-www-form-urlencoded

key	value
name	TestCoSpace2
uri	TestCoSpace2@vdepee.acano
callID	8675308

Headers (10) — Status: 200 OK

```
Connection → Keep-Alive
Content-Encoding → gzip
Date → Fri, 01 Apr 2016 15:33:22 GMT
Keep-Alive → timeout=5, max=99
Location → /api/v1/coSpaces/[illegible]
Server → Apache
Strict-Transport-Security → max-age=31536000; includeSubDomains
Transfer-Encoding → chunked
Vary → Accept-Encoding
X-Frame-Options → DENY
```

Verify

The verification method is explained in each request.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.