

Funciones básicas de la API de Cisco Meeting Server

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Solicitud y respuesta de API](#)

[Configurar](#)

[CMS 2.9 y posteriores](#)

[Configurar un objeto API](#)

[Modificar un objeto API que ya se ha creado](#)

[Eliminar un objeto API del menú API](#)

[Solicitudes de API explicadas \(mediante herramientas de terceros\)](#)

[HTTP POST](#)

[HTTP GET](#)

[HTTP PUT](#)

[HTTP DELETE](#)

[Clientes de API](#)

[POSTER](#)

[POSTMAN](#)

[Verificación](#)

[Troubleshoot](#)

Introducción

En este documento se describen las cuatro funciones básicas de la interfaz de programación de aplicaciones (API): GET, POST, PUT, DELETE, que se utilizan en CMS (Cisco Meeting Server). A partir de CMS 2.9, la GUI de administración web tiene un menú API disponible en el menú Configuración. Este documento revisa ese nuevo menú y también describe dos herramientas API diferentes: Poster y Postman, y cómo se utilizan para la configuración de CMS.

Prerequisites

Requirements

No hay requisitos específicos para este documento.

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de

hardware.

La información de este documento se puede utilizar con CMS 2.9 y posteriores o con diferentes clientes de API como Postman o Poster. Estas herramientas de terceros se describen en la sección Clientes de API del documento.

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.

Antecedentes

La API de CMS ofrece un medio sumamente flexible para configurar muchas de sus funciones. Existen muchísimas funciones de API para memorizar o abordar aquí, por lo tanto, no deje de consultar la documentación de referencia de la API actual. A partir del momento en que se redactó este documento, las guías de referencia de la API actuales están disponibles [aquí](#).

Solicitud y respuesta de API

La comunicación de API es una relación de solicitud y respuesta entre clientes y servidores. El cliente realiza una solicitud al servidor. Después de gestionar la solicitud (realizar una acción o negarse a hacerla) se arroja una respuesta.

API Request and Response



Las cuatro solicitudes descritas en este artículo son:

1. GET - Recupera información existente
2. POST - Crea nueva información
3. PUT: modifica la información existente
4. ELIMINAR: elimina la información existente.

Estas son las solicitudes API básicas utilizadas para configurar CMS.

La respuesta más común es 200 OK. Otras respuestas son 4xx y 5xx, que son respuestas de error.

Configurar

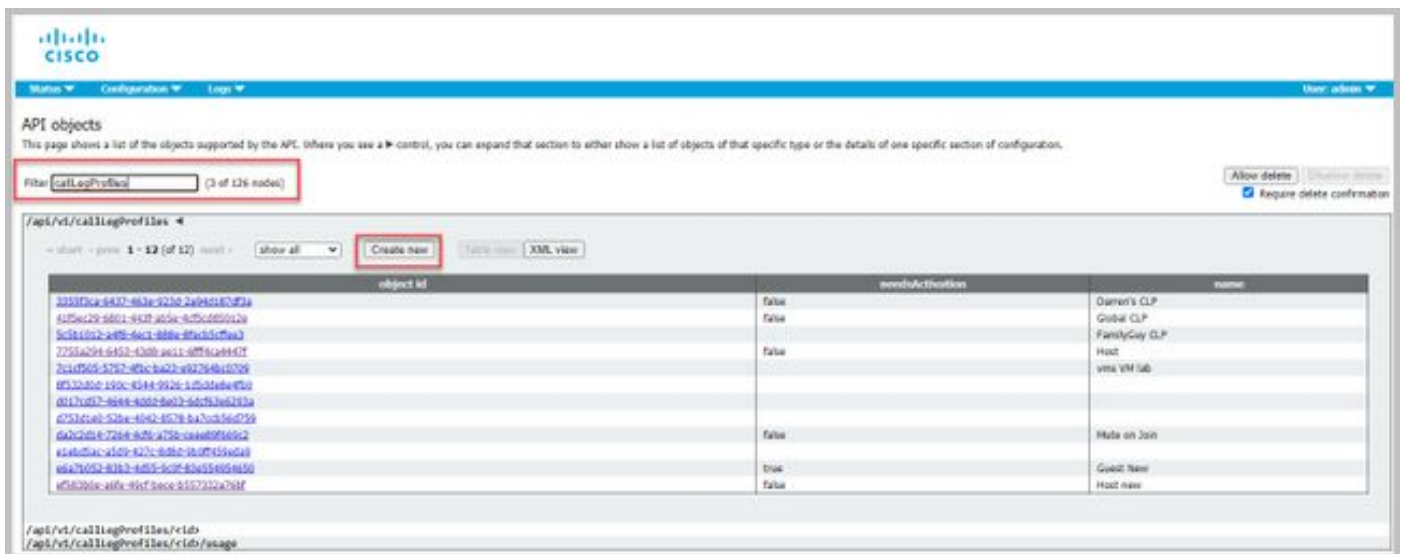
CMS 2.9 y posteriores

CMS 2.9 introdujo un nuevo menú de API que facilita a los administradores la modificación de la configuración y el ajuste preciso en CMS. Al utilizar el menú, se muestran todos los parámetros disponibles, lo que facilita y agiliza el cambio de los parámetros y la habilitación de nuevas funciones.

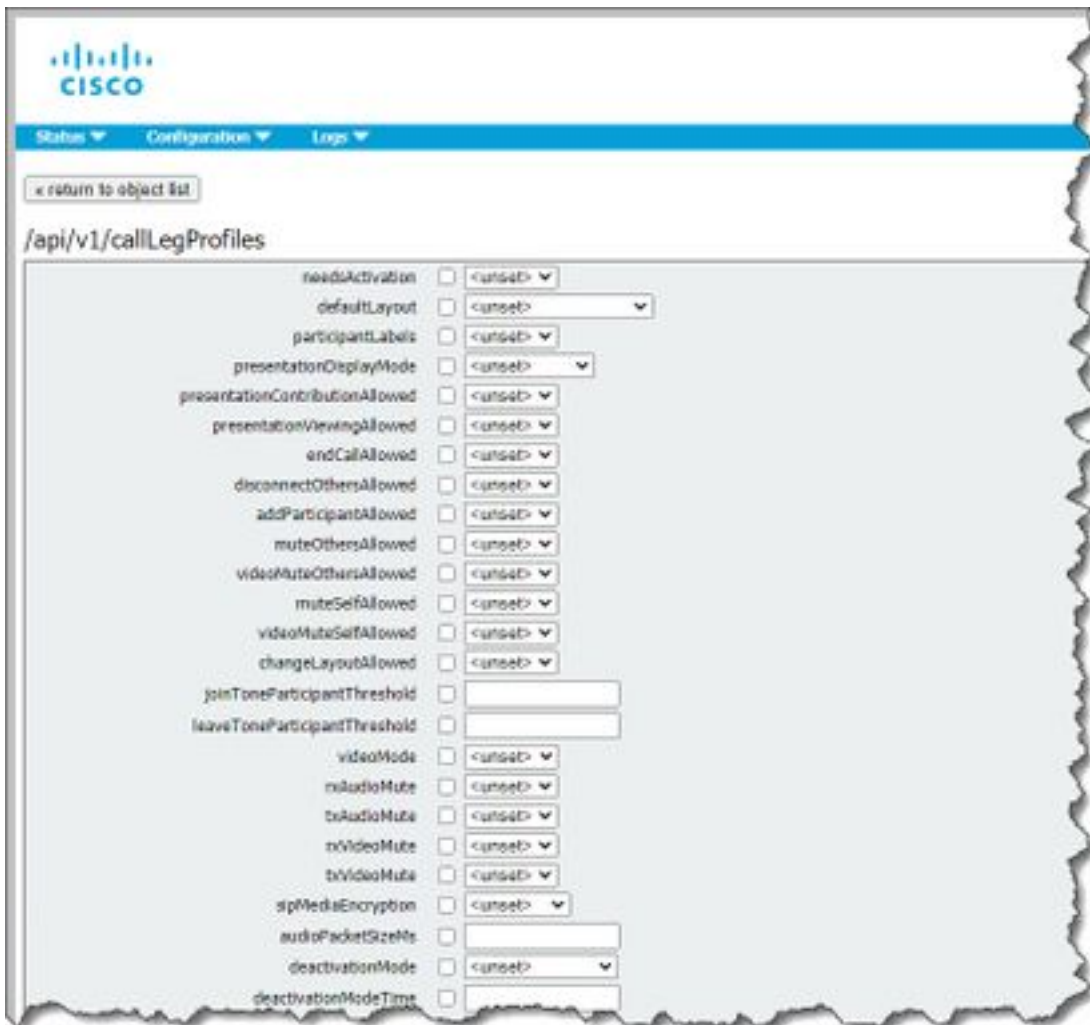


Configurar un objeto API

Una vez en el menú API, puede filtrar los objetos API a lo que desea editar/crear y, a continuación, hacer clic en la flecha negra situada junto al objeto para realizar esos cambios. Este ejemplo muestra cómo buscar callLegProfiles y crear un nuevo callLegProfile.



Al hacer clic en **Crear nuevo**, aparece esta pantalla que muestra todos los parámetros disponibles para CallLegProfiles. Al pasar el cursor sobre un parámetro concreto, se mostrará una ventana emergente con el propósito de cada opción.



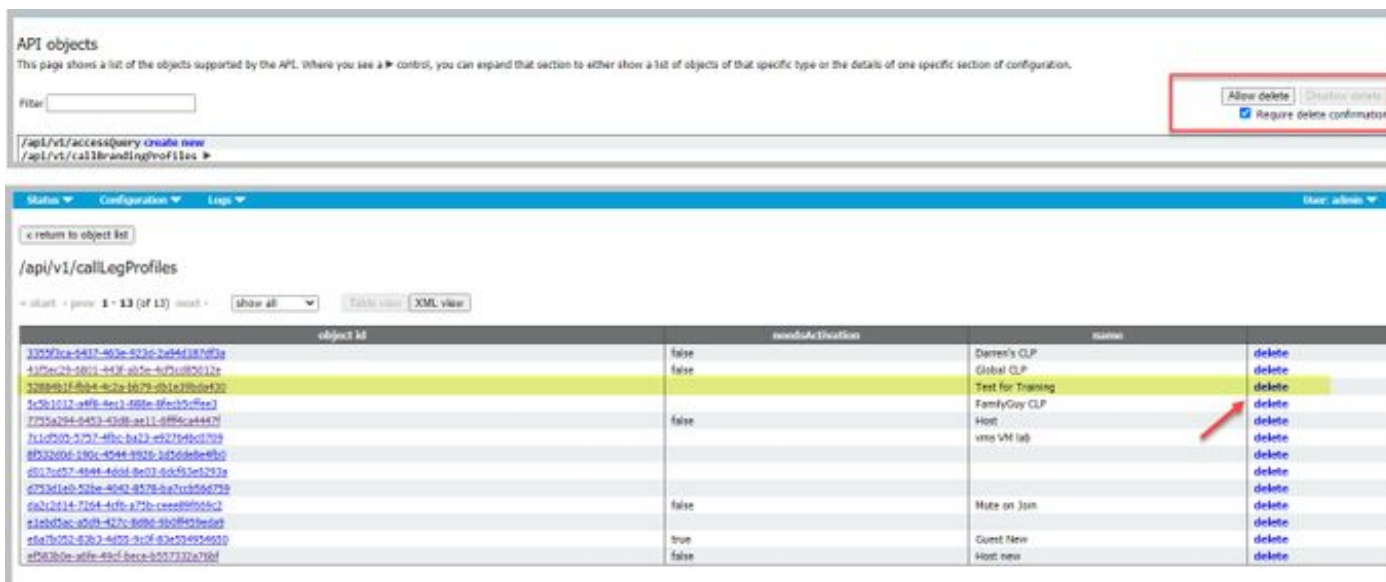
Modificar un objeto API que ya se ha creado

Cuando cambie la configuración de un objeto, verá el botón Modificar en la parte inferior. Esto es lo mismo que una PUT de herramientas de terceros.



Eliminar un objeto API del menú API

Para eliminar un objeto, en la página principal Lista de objetos, puede habilitar la posibilidad de eliminar elementos. Haga clic en **Permitir eliminar** para habilitar la opción de eliminación, como se muestra en este ejemplo:



Solicitudes de API explicadas (mediante herramientas de terceros)

Estas cuatro solicitudes básicas se explican a través de un ejemplo de configuración.

HTTP POST

Paso 1. Use **POST** para crear un objeto.

En este ejemplo, se crea un espacio CMS con esta solicitud. Para crear el espacio a través de la API, consulte la documentación de la API. Para este ejemplo utilicé la guía de la API CMS 2.4, pero debería utilizar las guías de la API más recientes, que se encuentran [aquí](#)

La sección 6.2 contiene información sobre cómo crear y modificar un espacio en común.

6.2 Creating and Modifying a coSpace	37
6.2.1 Secondary coSpace URIs	40
6.2.2 Auto-generation of coSpace callId	40

La primera oración indica que para crear un espacio, debe enviar una solicitud POST a /coSpaces. Luego, indica que el ID del espacio estará en el encabezado Location (Ubicación) de 200 OK. ¡Perfecto! Ahora sabe cómo crear un espacio. Solo debe enviar una solicitud **POST** a <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

Especifique los parámetros para la solicitud **POST**.

En la sección 6.2 de la documentación verá una tabla que enumera todos los parámetros que puede utilizar.

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)

Por ejemplo: Crear un espacio con el nombre **APITest** y una parte de usuario del URI de **APITestURI**.

El tipo de contenido es **application/x-www-form-urlencoded** y el contenido es **name=APITest&uri=APITestURI**.

La solicitud está completa después de que se agregan estos parámetros, como se muestra en la imagen.

```
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
```

La imagen muestra una respuesta a la solicitud anterior.

```
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
```

Observe el encabezado Location (Ubicación) en la respuesta.

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

70ca0ed7-4e50-428c-b9ab-4e68faeb86ae es el ID del espacio nuevo creado. La ID es útil cuando necesita realizar futuras solicitudes de API que tengan como destino el mismo espacio.

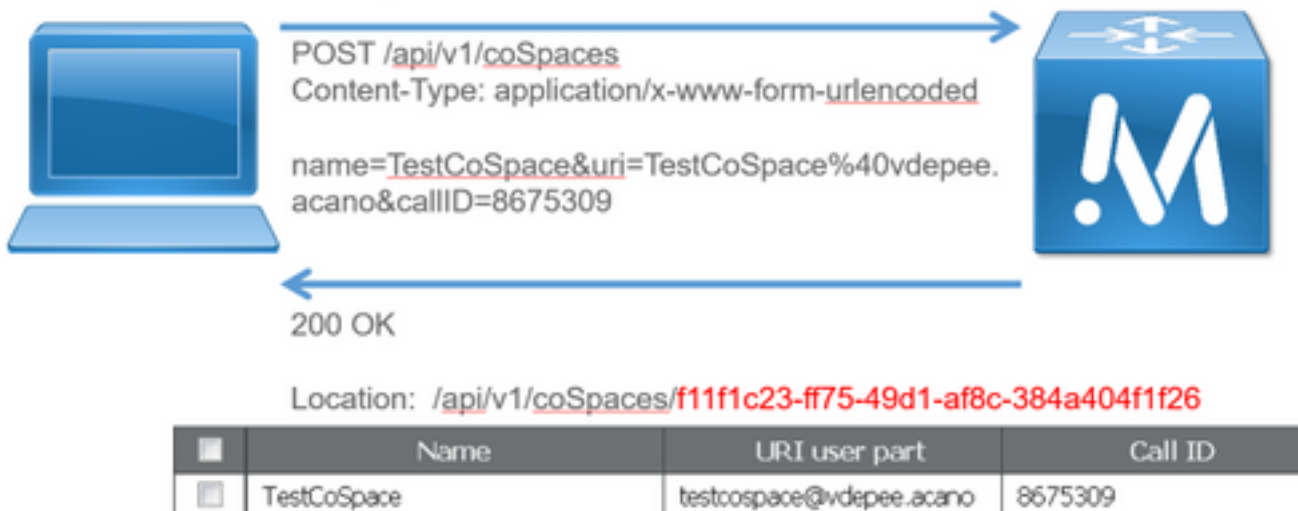
El espacio puede verse en WebAdmin de CMS. Vaya a **Configuration > Spaces (Configuración > Espacios)**.

	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout
<input type="checkbox"/>	APITest	apitesturi					not set

La imagen muestra un resumen de la solicitud **POST**.

HTTP POST

- Creates new object



HTTP GET

Paso 2. Una vez que se ha creado el espacio, obtenga la configuración para él.

Utilice el método HTTP GET con este fin.

Utilice el ID para el espacio creado en el encabezado Location (Ubicación). La dirección URL es <https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae>. Realice una solicitud **GET en esta página**.

Ejemplo de solicitud GET:

```
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
```

Respuesta a la solicitud GET:

```
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></coSpace>
```

Nota: La respuesta es una configuración codificada XML del espacio.

La imagen resume la solicitud **GET**.

HTTP GET

- Retrieves existing information
- No Content in Body



HTTP PUT

Paso 3. Realice un cambio en el espacio (si fuera necesario).

En este ejemplo se muestra cómo modificar el espacio creado. Imagínese que una parte de usuario secundaria debe agregarse al espacio.

Consulte el documento de API. Indica el parámetro que se debe utilizar: **secondaryUri**.

Agregue un URI de asdf. Escriba una solicitud con una apariencia similar a la solicitud que se crea para la solicitud POST.

Ejemplo de solicitud PUT:

```
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

Respuesta a la solicitud PUT:

```
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
```

Los cambios pueden verse en WebAdmin de CMS. Vaya a **Configuration > Spaces (Configuración > Espacios)**.

	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout
<input type="checkbox"/>	APITest	apitesturi	asdf				not set

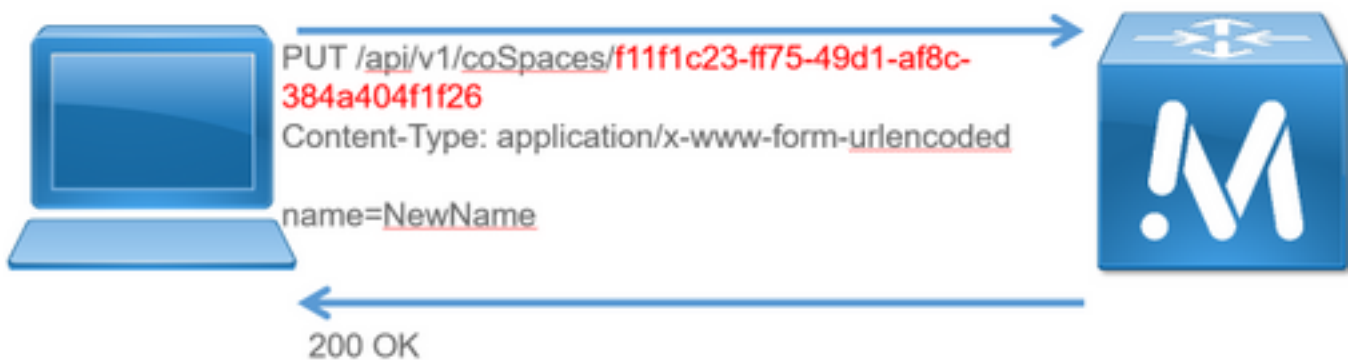
También a través de **GET**:

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

La imagen resume la solicitud **PUT**.

HTTP PUT

- Modifies existing object



	Name	URI user part	Call ID
<input type="checkbox"/>	NewName	testcospace@vdepee.acano	8675309

HTTP DELETE

Paso 4. Elimine el espacio (si es necesario).

El método **DELETE** es similar al método **GET**.

Ejemplo de solicitud **DELETE**:

```
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
```

Respuesta a la solicitud DELETE:

```
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
```

Los cambios pueden verse en WebAdmin de CMS. Vaya a **Configuration > Spaces (Configuración > Espacios)**.

Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout	
						not set	<input type="button" value="Add New"/> <input type="button" value="Reset"/>

También a través de **GET**:

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

La imagen resume la solicitud **DELETE**.

HTTP DELETE

- Destroys an object



Name	URI user part	Call ID

Cientes de API

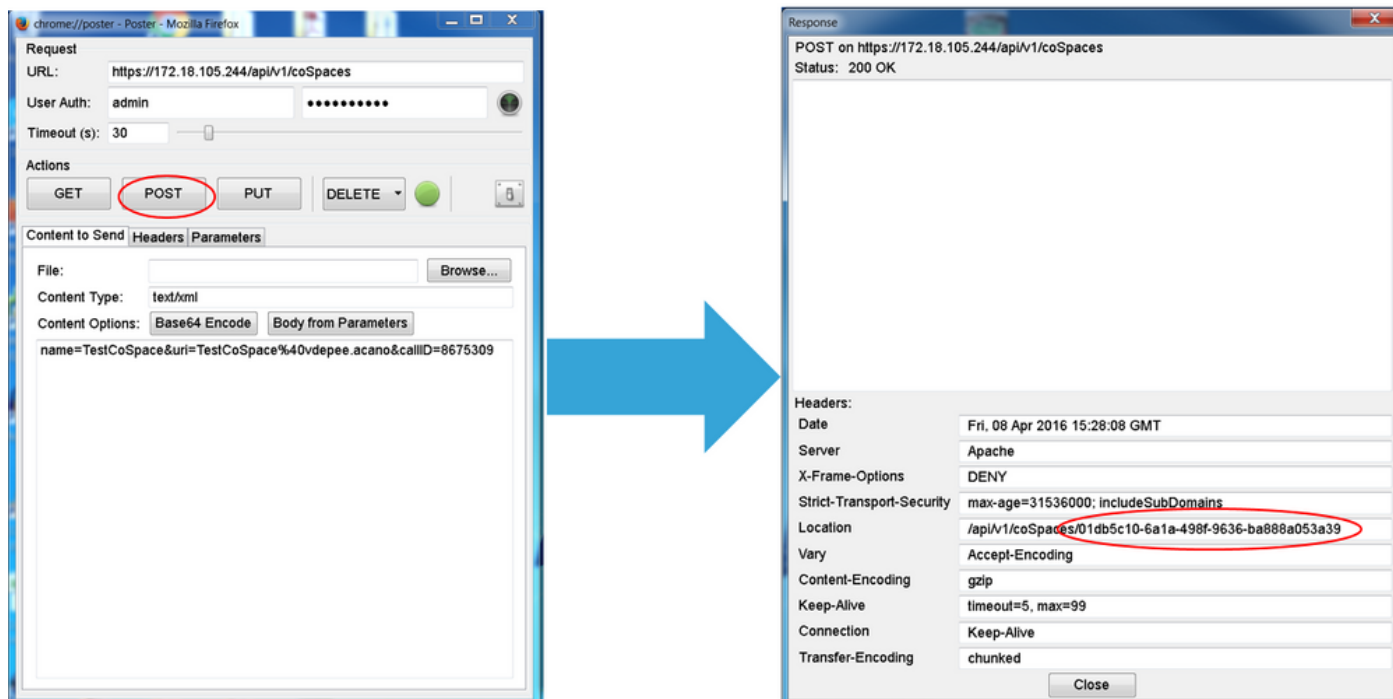
POSTER

El cuadro superior en Poster es donde introduce la dirección URL para las solicitudes.

En los campos User Auth (Autenticación del usuario) introduce el nombre de usuario y la contraseña (en ese orden). A continuación, si está realizando una operación **GET** o **DELETE**, elija

los botones correspondientes. Por ejemplo: haga clic en **GET** y aparecerá una ventana emergente con su respuesta. En el caso de **DELETE**, asegúrese de que la opción **DELETE** esté seleccionada y haga clic en el botón verde.

Poster (Firefox)



En el caso de las solicitudes **POST** y **PUT**, es necesario agregar contenido. Seleccione la ficha Parameters (Parámetros) y agregue los nombres y los valores para los parámetros. Luego, vuelva al botón Content to send y elija **Body from Parameters**.

Envíe su solicitud POST o PUT.

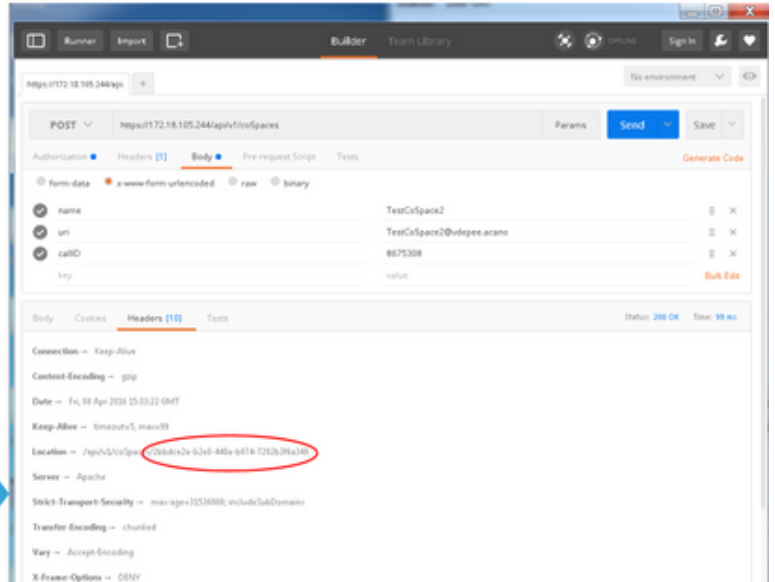
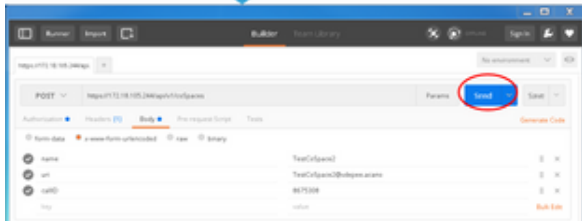
POSTMAN

En Postman, en la parte superior izquierda, elija el método que desea utilizar en el cuadro desplegable e introduzca la URL de solicitud.

Para Autorización, elija **Basic-Auth** e ingrese su nombre de usuario y contraseña. A continuación, elija **Update Request**. En la ficha Headers (Encabezados), verá el encabezado Authorization (Autorización).

Si su solicitud es POST/PUT, navegue a la pestaña Body, elija **x-www-form-urlencoded** e introduzca sus parámetros y valores. Cuando haya terminado, elija **Enviar**.

Postman (Chrome)



Verificación

El método de verificación se explica en cada solicitud.

Troubleshoot

Actualmente, no hay información específica de troubleshooting disponible para esta configuración.