

Funções básicas da API do Cisco Meeting Server

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Solicitação de API e resposta](#)

[Configurar](#)

[CMS 2.9 e posterior](#)

[Configurar um objeto API](#)

[Modificar um objeto API já criado](#)

[Excluir um objeto API do menu API](#)

[Solicitações de API explicadas \(usando ferramentas de terceiros\)](#)

[HTTP POST](#)

[HTTP GET](#)

[HTTP PUT](#)

[HTTP DELETE](#)

[Clientes API](#)

[Poster](#)

[POSTMAN](#)

[Verificar](#)

[Troubleshoot](#)

Introduction

Este documento descreve as quatro funções de API (Application Programa Interface) básicas GET, POST, PUT, DELETE usadas no CMS (Cisco Meeting Server). A partir do CMS 2.9, a GUI do administrador da Web tem um menu de API disponível no menu Configuração. Este documento analisa esse novo menu e também descreve duas ferramentas de API diferentes: Poster e Postman e como usá-las para a configuração do CMS.

Prerequisites

Requirements

Não existem requisitos específicos para este documento.

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento podem ser usadas com CMS 2.9 e posterior ou com clientes API diferentes, como Postman ou Poster. Essas ferramentas de terceiros são descritas na seção clientes API do 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.

Informações de Apoio

A API do CMS é uma maneira extremamente flexível de configurar muitos de seus recursos. Existem muitos recursos da API para memorizar ou abordar aqui, portanto, não deixe de consultar a documentação de referência da API atual. Na elaboração deste documento, os guias de referência da API atuais foram disponibilizados [aqui](#).

Solicitação de API e resposta

A comunicação de API é um relacionamento de solicitação/resposta entre clientes e servidores. O cliente faz uma solicitação do servidor. Após tratar a solicitação (concluir uma ação ou recusar fazê-la) uma resposta é retornada.

API Request and Response



As quatro solicitações descritas neste artigo são:

1. GET - Recupera informações existentes
2. POST - Cria novas informações
3. PUT - Modifica as informações existentes
4. DELETE - Exclui informações existentes.

Essas são as solicitações básicas de API usadas para configurar o CMS.

A resposta mais comum é um 200 OK. Outras respostas são 4xx e 5xx, que são respostas a erros.

Configurar

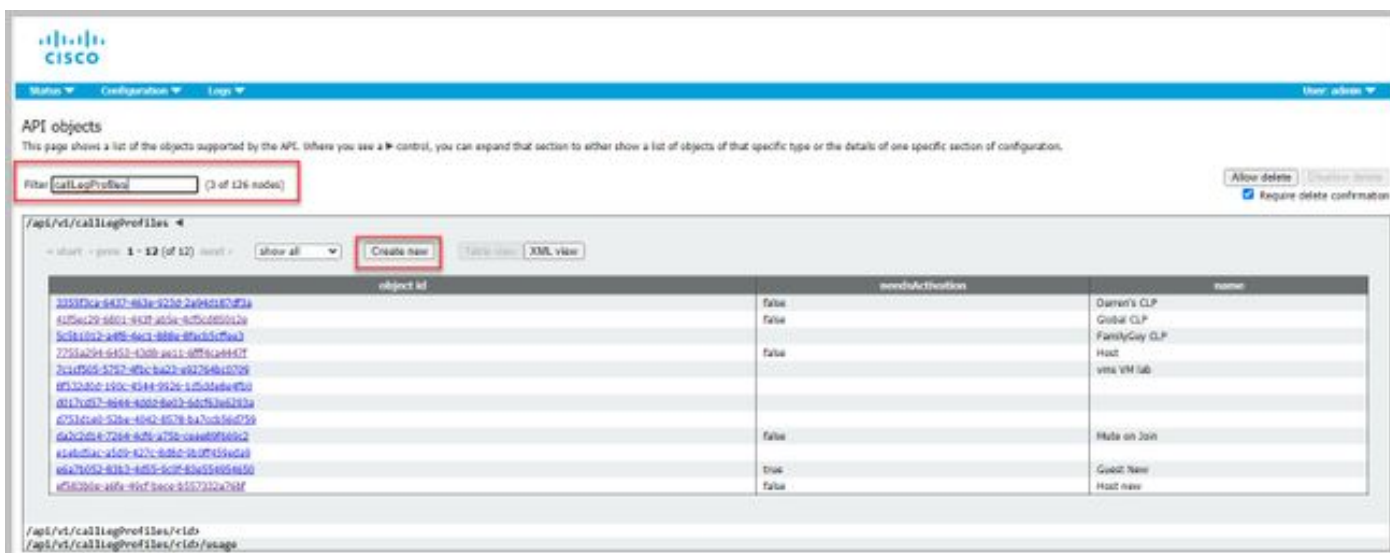
CMS 2.9 e posterior

O CMS 2.9 introduziu um novo menu de API que torna muito mais fácil para os administradores modificar as configurações e ajustar as configurações no CMS. Ao usar o menu, todos os parâmetros disponíveis são exibidos, o que facilita e agiliza a alteração de configurações e habilita novos recursos.

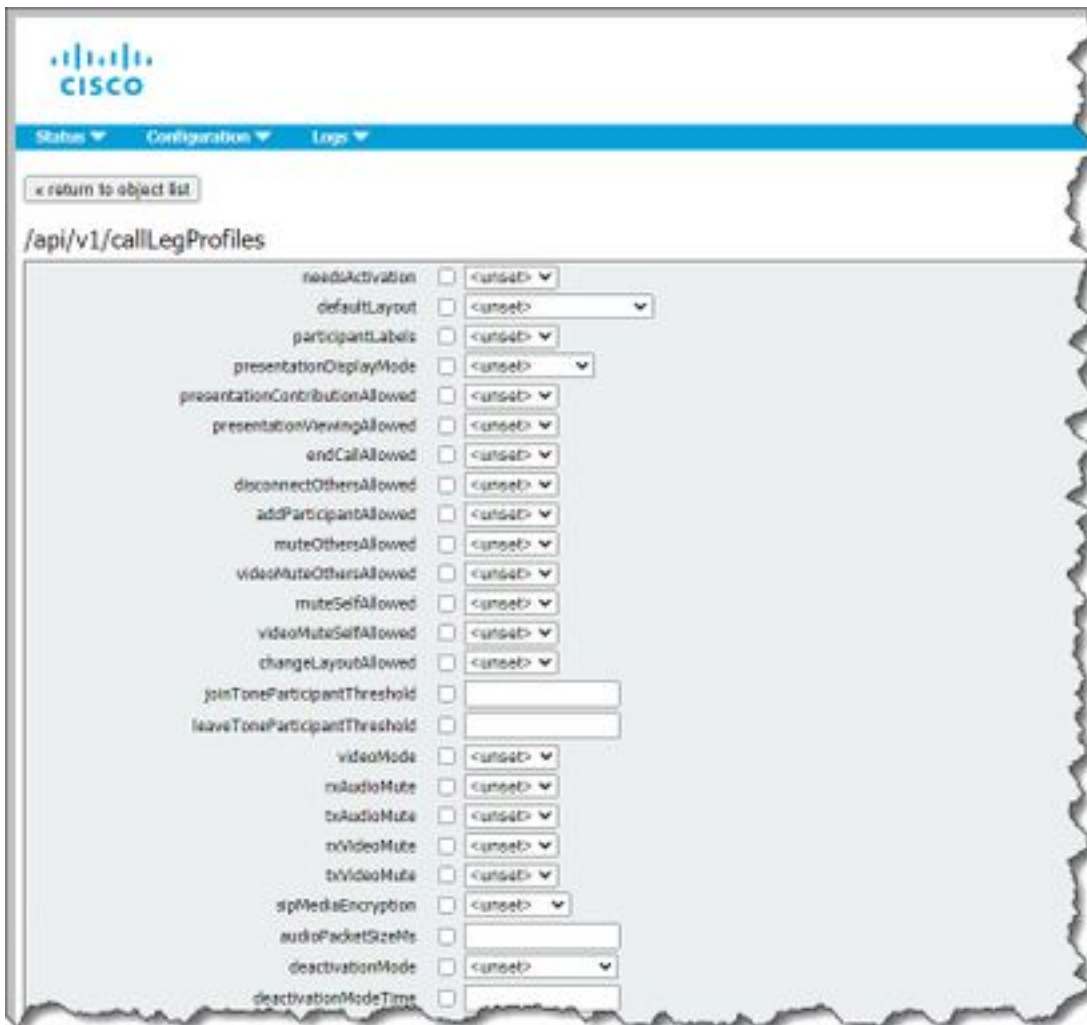


Configurar um objeto API

No menu da API, você pode filtrar os objetos da API para o que deseja editar/criar e, em seguida, clicar na seta preta ao lado do objeto para fazer essas alterações. Este exemplo mostra como pesquisar callLegProfiles e criar um novo callLegProfile.



Quando você clica em **Criar novo**, esta tela é exibida para você, mostrando todos os parâmetros disponíveis para CallLegProfiles. Ao passar o mouse sobre um parâmetro específico, você receberá um pop-up que mostra o objetivo de cada opção.



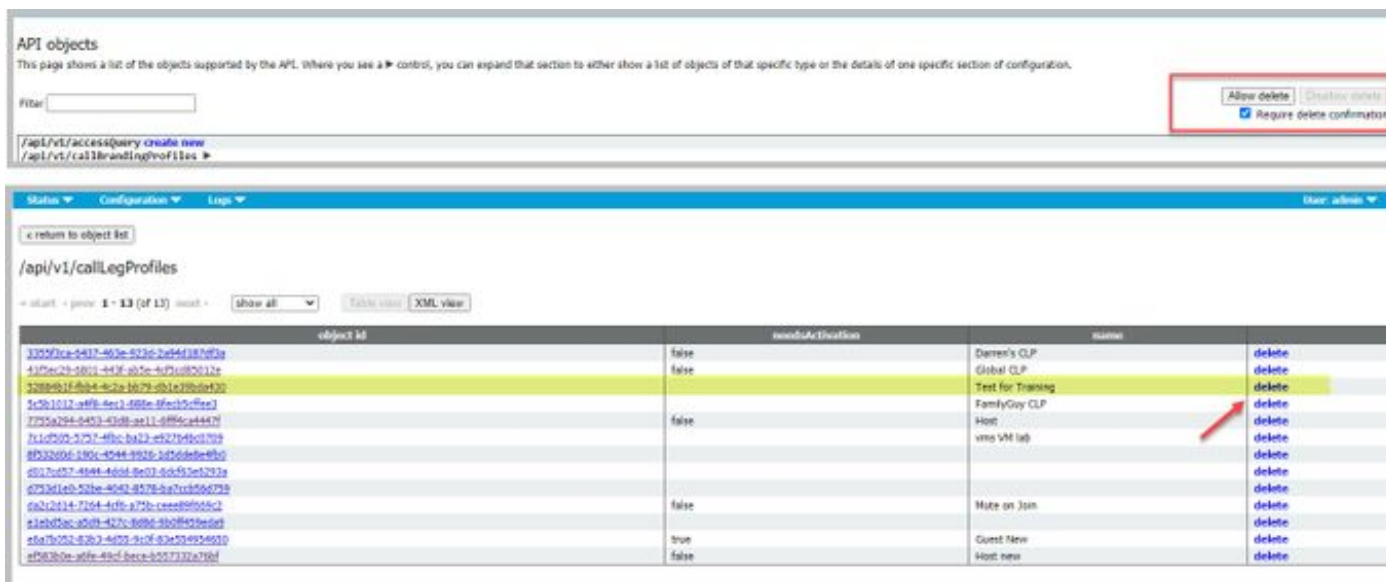
Modificar um objeto API já criado

Quando você altera as configurações em um objeto, verá o botão Modificar na parte inferior. Isso é o mesmo que uma PUT de ferramentas de terceiros.



Excluir um objeto API do menu API

Para excluir um objeto, na página principal Lista de objetos, você pode habilitar a capacidade de excluir itens. Clique em **Permitir exclusão** para habilitar a opção de exclusão, como mostrado neste exemplo:



Solicitações de API explicadas (usando ferramentas de terceiros)

As quatro solicitações básicas são explicadas por meio de um exemplo de configuração.

HTTP POST

Etapa 1. Use o **POST** para criar um objeto.

Neste exemplo, um espaço CMS é criado usando esta solicitação. Para criar o Espaço via API, consulte a documentação da API. Para este exemplo, usei o guia de API do CMS 2.4, mas você deve usar os guias de API mais recentes, encontrados [aqui](#)

A seção 6.2 tem informações sobre como criar e modificar um espaço em grupo.

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

A primeira frase diz que, para criar um espaço, você precisa enviar uma postagem para /coSpaces. Em seguida, diz que a ID do espaço estará no cabeçalho do local do 200 OK. Ótimo, agora você sabe como criar um espaço. Você acabou de enviar um **POST** para **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 os parâmetros para o **POST**.

Na seção 6.2 da documentação, você vê uma tabela que lista todos os parâmetros que você pode usar.

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 exemplo: Crie um espaço com o nome **APITest** e uma parte do usuário de URI de **APITestURI**

O tipo de conteúdo é **aplicativo/x-www-form-urlencoded** e o conteúdo é **name=APITest&uri=APITestURI**

Quando você adiciona esses parâmetros, a solicitação é concluída, como mostrado na imagem.

```
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
```

A imagem mostra uma resposta à solicitação 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 o cabeçalho do local na resposta.

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

70ca0ed7-4e50-428c-b9ab-4e68faeb86ae é a ID do novo espaço. A ID é útil quando você precisa fazer futuras solicitações de API que visem o mesmo espaço.

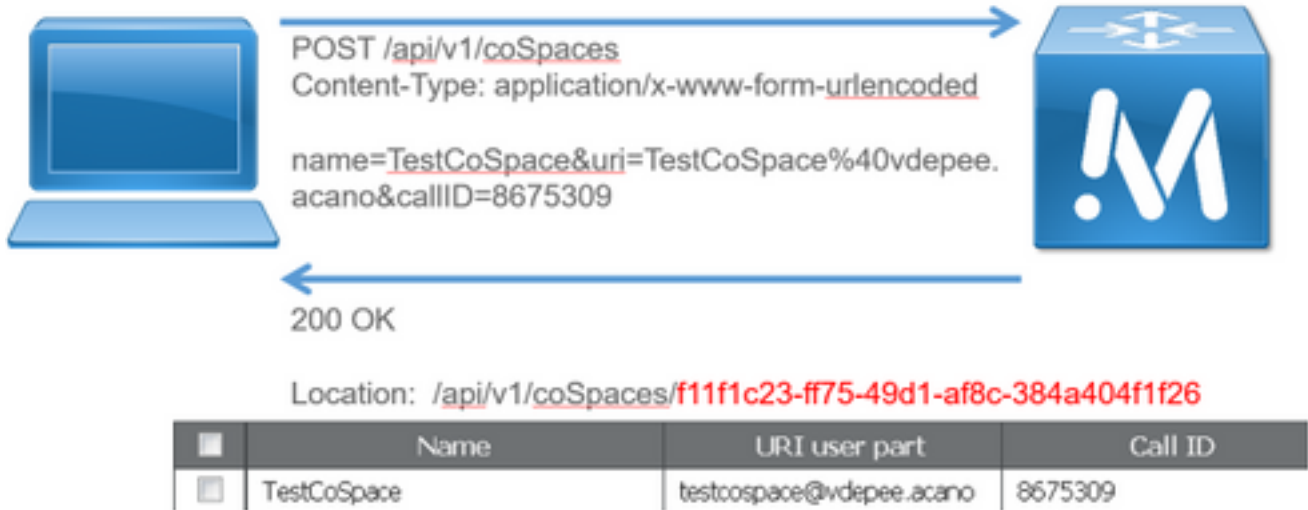
O Espaço pode ser visto no WebAdmin do CMS. Navegue até **Configuração > Espaços**.

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

A imagem resume a solicitação **POST**.

HTTP POST

- Creates new object



HTTP GET

Etapa 2. Depois que o Espaço tiver sido criado, obtenha a configuração para ele.

Use o método HTTP GET para essa finalidade.

Use a ID para o Espaço criado no cabeçalho do local. A URL é <https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae>. Execute um **GET** nesta página.

Exemplo de solicitação 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
```

Resposta para a solicitação 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></coS
```

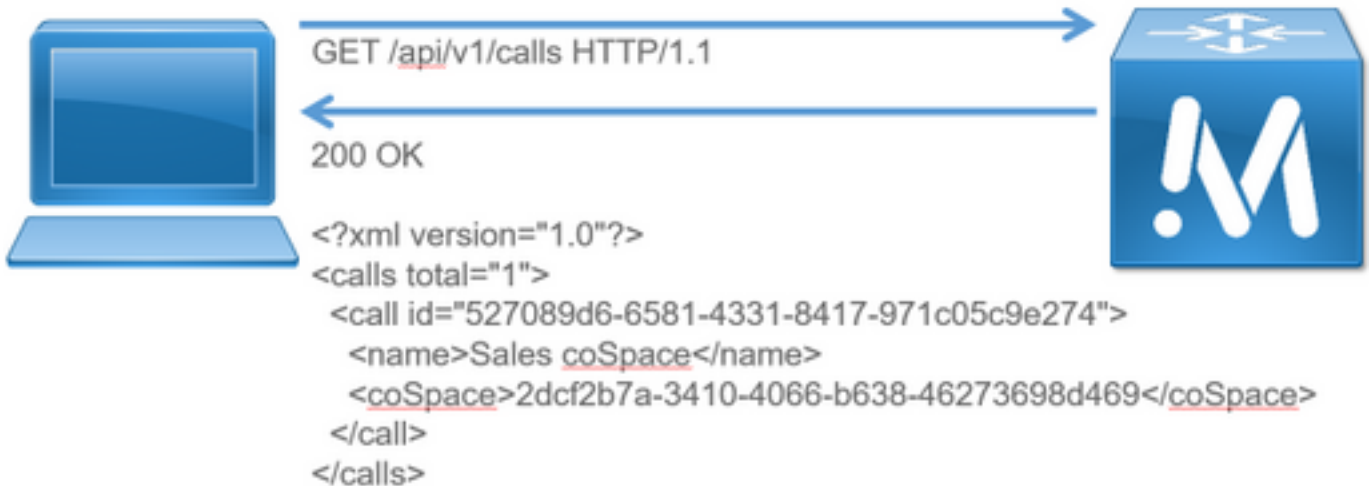
pace>

Note: A resposta é uma configuração codificada em XML do Espaço.

A imagem resume a solicitação GET.

HTTP GET

- Retrieves existing information
- No Content in Body



HTTP PUT

Etapa 3. Faça uma alteração no Espaço (se necessário).

Este exemplo mostra como modificar o espaço criado. Imagine que uma parte do Usuário secundário precisa ser adicionada ao Espaço.

Consulte o documento da API. Ele informa o parâmetro necessário a ser usado, isto é: **secondaryUri**.

Adicione um URI de asdf. Escreva uma solicitação que seja similar à solicitação criada para o POST.

Exemplo de solicitação 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

Resposta para a solicitação 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
```

As alterações podem ser vistas no WebAdmin do CMS. Navegue até **Configuração > Espaços**.

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

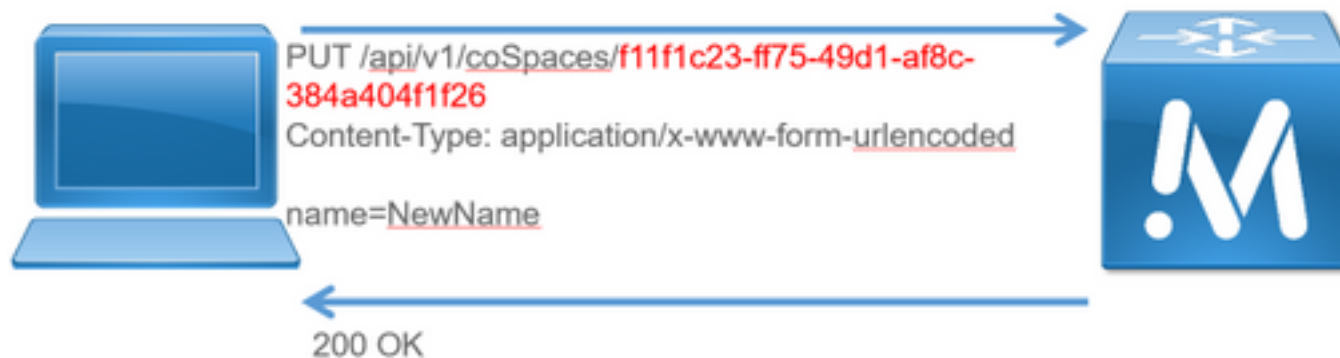
e por meio de **GET**:

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

A imagem resume a solicitação **PUT**.

HTTP PUT

- Modifies existing object



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

HTTP DELETE

Etapa 4. Exclua o espaço (se necessário).

O método **DELETE** é semelhante ao **GET**.

Exemplo de solicitação **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

Resposta para a solicitação 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

As alterações podem ser vistas no WebAdmin do CMS. Navegue até **Configuração > Espaços**.

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"/>

e por meio de **GET**:

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

A imagem resume a solicitação **DELETE**.

HTTP DELETE

- Destroys an object



Name	URI user part	Call ID

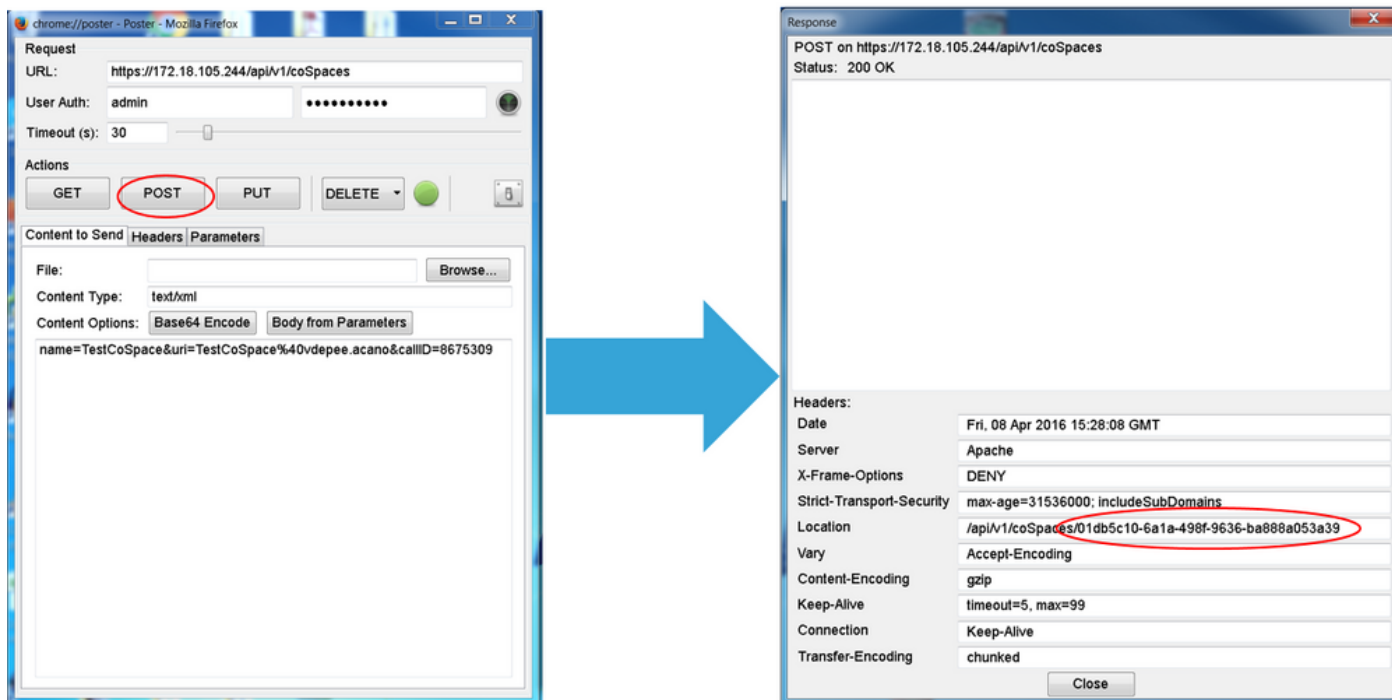
Clientes API

Poster

Você insere a URL para as solicitações na caixa superior no Poster.

Você insere o nome de usuário e senha nessa ordem, nos campos de autenticação de usuário. Em seguida, se estiver fazendo um **GET** ou um **DELETE**, escolha os respectivos botões. Por exemplo: clique em **GET** e um pop-up será exibido com sua resposta. Para **DELETE**, certifique-se de que **DELETE** esteja selecionada e clique no botão verde.

Poster (Firefox)



Para **POST** e **PUT**, o conteúdo precisa ser adicionado. Selecione a guia **Parâmetros** e adicione os nomes e valores para os parâmetros. Em seguida, retorne ao botão **Conteúdo** para enviar e escolha **Corpo dos Parâmetros**.

Envie seu **POST** e/ou **PUT**.

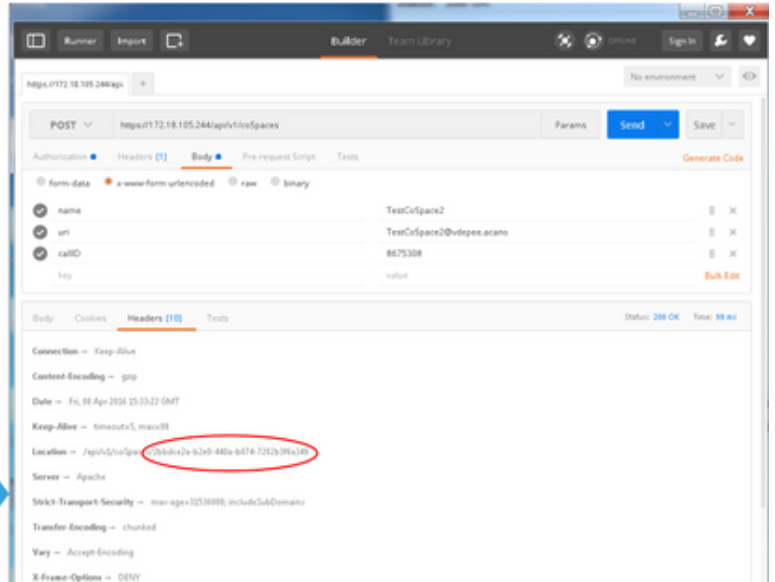
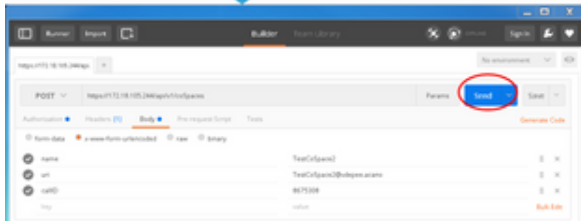
POSTMAN

No Postman, na parte superior esquerda, escolha o Método que você gostaria de usar na caixa suspensa e digite o URL da solicitação.

Para autorização, escolha **Basic-Auth** e digite seu nome de usuário e senha. Em seguida, escolha **Atualizar solicitação**. Na guia **Cabeçalhos**, você vê um Cabeçalho de autorização.

Se sua solicitação for um **POST/PUT**, navegue até a guia **Body (Corpo)**, escolha **x-www-form-urlencoded** e insira seus parâmetros e valores. Quando terminar, escolha **Enviar**.

Postman (Chrome)



Verificar

O método de verificação é explicado em cada pedido.

Troubleshoot

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.