

Fonctions de base de l'API pour le serveur de réunion Cisco

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Requêtes et réponses d'API](#)

[Configuration](#)

[CMS 2.9 et versions ultérieures](#)

[Configurer un objet API](#)

[Modifier un objet API déjà créé](#)

[Supprimer un objet API du menu API](#)

[Demandes API expliquées \(à l'aide d'outils tiers\)](#)

[HTTP POST](#)

[HTTP GET](#)

[HTTP PUT](#)

[HTTP DELETE](#)

[Clients de l'API](#)

[POSTER](#)

[POSTMAN](#)

[Vérification](#)

[Dépannage](#)

Introduction

Ce document décrit les quatre fonctions de base de l'API (Application Program Interface ou Interface de programmation d'application), GET, POST, PUT et DELETE utilisées sur CMS (Cisco Meeting Server ou serveur de réunion Cisco). Depuis CMS 2.9, l'interface utilisateur graphique de l'administrateur Web dispose d'un menu API disponible sous le menu Configuration. Ce document passe en revue ce nouveau menu et décrit également deux outils d'API différents : Poster et Postman et comment les utiliser pour la configuration de CMS.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Les informations de ce document peuvent être utilisées avec CMS 2.9 et versions ultérieures ou avec différents clients API comme Postman ou Poster. Ces outils tiers sont décrits dans la section clients API du 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.

Informations générales

L'API pour CMS est une façon extrêmement flexible de configurer beaucoup de ses fonctionnalités. Il y a beaucoup trop de fonctionnalités de l'API pour toutes les mémoriser ou les couvrir ici; assurez-vous alors de consulter la documentation de référence actuelle sur l'API. Au moment de la rédaction de ce document, les guides de référence actuels de l'API sont disponibles [ici](#).

Requêtes et réponses d'API

La communication avec l'API est une relation requête-réponse entre les clients et des serveurs. Le client fait une requête au serveur. Après avoir traité la requête (compléter une action, ou refuser de le faire), une réponse est retournée.

API Request and Response



Les quatre demandes décrites dans cet article sont les suivantes :

1. GET - Récupère les informations existantes
2. POST - Crée de nouvelles informations
3. PUT - Modifie les informations existantes
4. SUPPRIMER - Supprime les informations existantes.

Il s'agit des requêtes d'API de base utilisées pour configurer CMS.

La réponse la plus courante est un 200 OK. Les autres réponses sont 4xx et 5xx, qui sont des réponses d'erreur.

Configuration

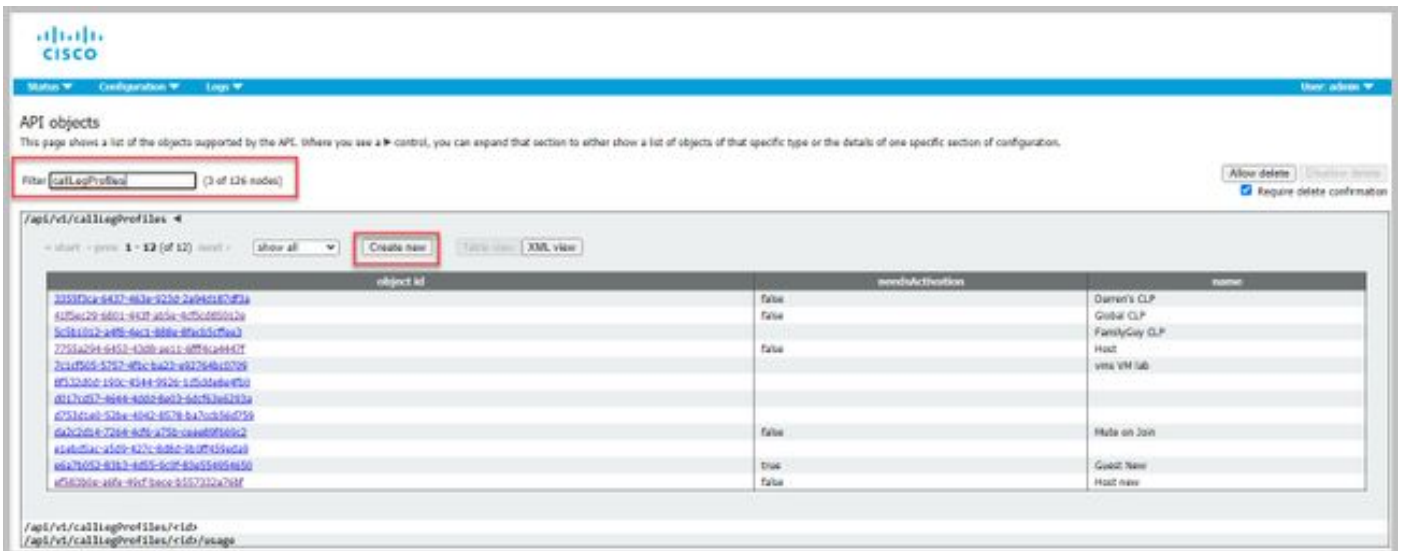
CMS 2.9 et versions ultérieures

CMS 2.9 a introduit un nouveau menu API qui facilite la modification des paramètres et leur réglage dans CMS. Lorsque vous utilisez le menu, tous les paramètres disponibles s'affichent, ce qui permet de modifier rapidement et facilement les paramètres et d'activer de nouvelles fonctionnalités.

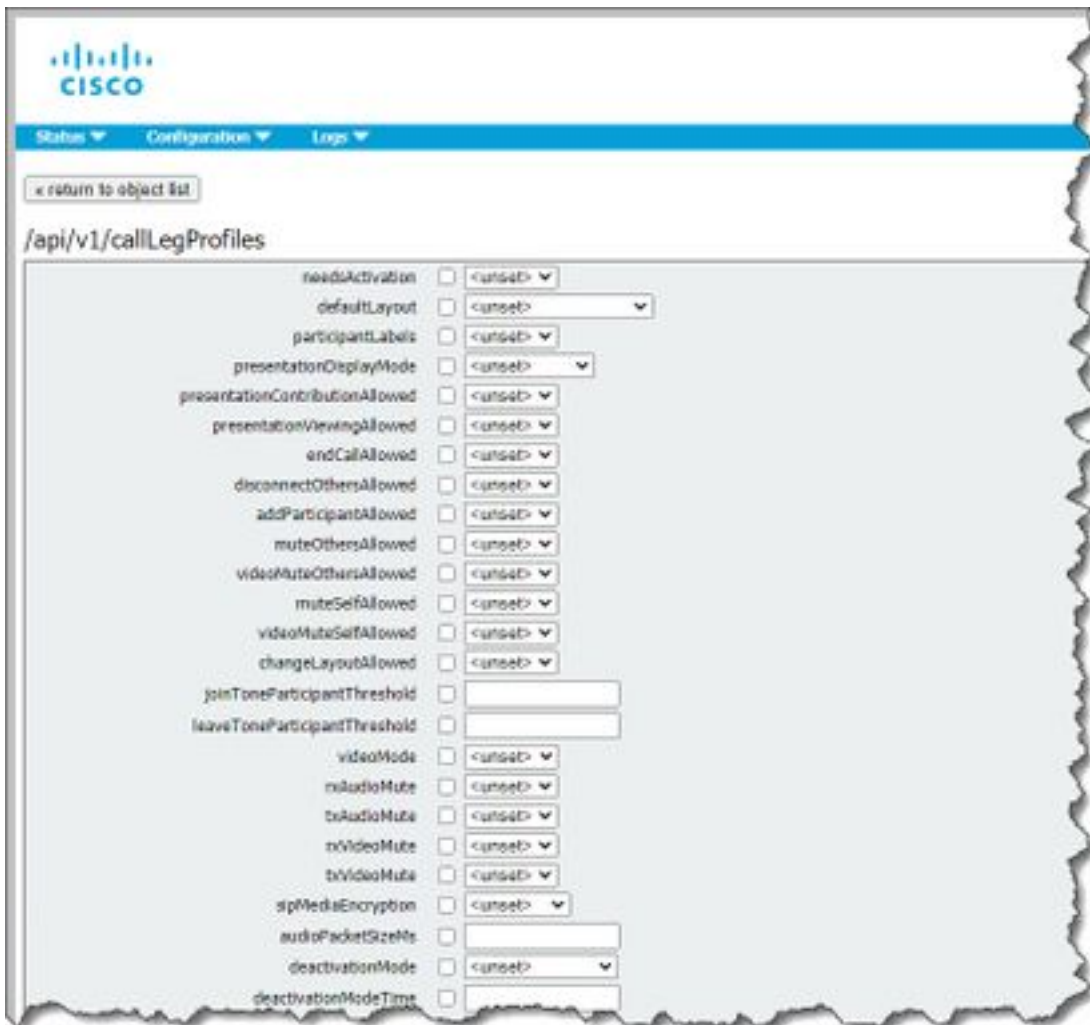


Configurer un objet API

Une fois dans le menu API, vous pouvez filtrer les objets API sur ce que vous souhaitez modifier/créer, puis cliquez sur la flèche noire en regard de l'objet pour effectuer ces modifications. Cet exemple montre comment rechercher callLegProfiles et créer un callLegProfile.



Lorsque vous cliquez sur **Créer**, cet écran affiche tous les paramètres disponibles pour CallLegProfiles. Lorsque vous survolez un paramètre particulier, vous obtenez une fenêtre contextuelle indiquant l'objectif de chaque option.



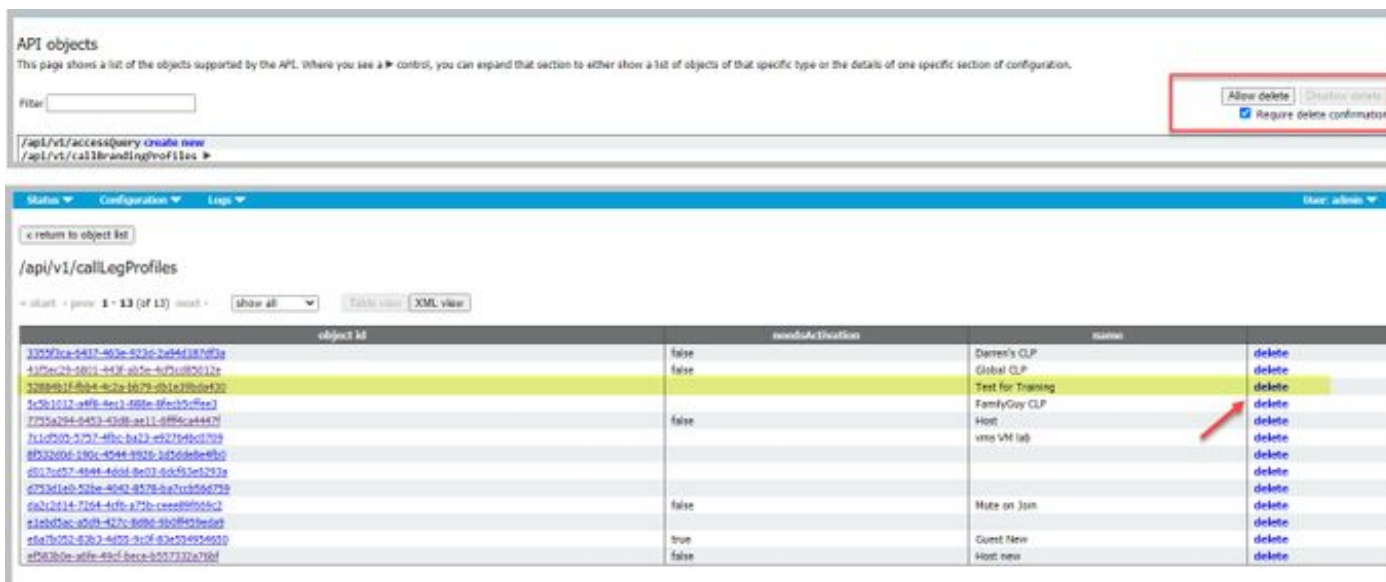
Modifier un objet API déjà créé

Lorsque vous modifiez les paramètres d'un objet, le bouton Modifier apparaît en bas. C'est la même chose qu'un PUT des outils tiers.



Supprimer un objet API du menu API

Afin de supprimer un objet, sur la page Liste d'objets principale, vous pouvez activer la possibilité de supprimer des éléments. Cliquez sur **Autoriser la suppression** afin d'activer l'option à supprimer, comme indiqué dans cet exemple :



Demandes API expliquées (à l'aide d'outils tiers)

Les quatre requêtes de base sont expliquées par l'intermédiaire d'un exemple de configuration.

HTTP POST

Étape 1. Utilisez **POST** pour créer un objet.

Dans cet exemple, un espace CMS est créé à l'aide de cette requête. Afin de créer l'espace via l'API, consultez la documentation de l'API. Pour cet exemple, j'ai utilisé le guide d'API CMS 2.4, mais vous devez utiliser les derniers guides d'API, [ici](#)

La section 6.2 contient des informations sur la création et la modification d'un espace cosmopolite.

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 première phrase mentionne que pour créer un espace, vous devez envoyer un POST à /coSpaces. Ensuite, on mentionne que l'ID de l'espace sera dans l'en-tête Location (emplacement) du 200 OK. Excellent, vous savez maintenant comment créer un espace. Vous envoyez simplement un **POST** à <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

Spécifiez les paramètres pour le **POST**.

Dans la section 6.2 de la documentation, vous voyez un tableau qui répertorie tous les paramètres que vous pouvez utiliser.

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.)

Exemple : Créez un espace portant le nom **APITest** et un utilisateur URI faisant partie de **APITestURI**

Le type de contenu est **application/x-www-form-urlencoded** et le contenu est **name=APITest&uri=APITestURI**

Lorsque vous ajoutez ces paramètres, la requête est terminée, comme illustré dans l'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
```

L'image affiche une réponse à la requête précédente.

```
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
```

Remarquez l'en-tête Location (emplacement) dans la réponse.

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

70ca0ed7-4e50-428c-b9ab-4e68faeb86ae correspond à l'ID du nouvel espace créé. L'ID est utile lorsque vous devez effectuer des demandes d'API futures qui ciblent le même espace.

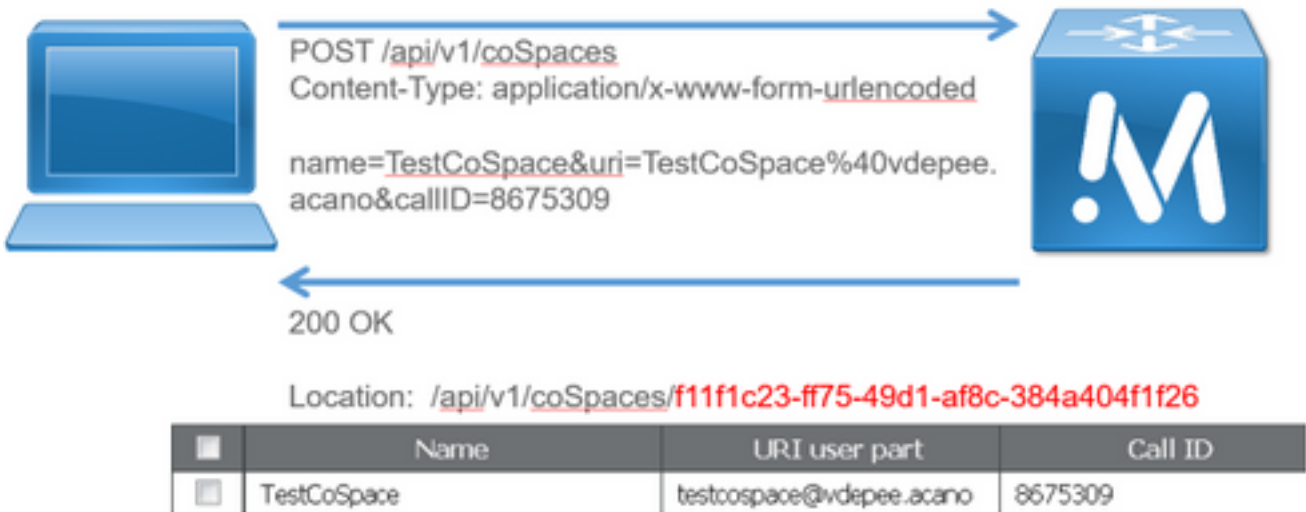
L'espace peut être vu dans le WebAdmin de CMS. Naviguez jusqu'à **Configuration > Spaces (Configuration > Espaces)**.

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

L'image représente un résumé de la requête **POST**.

HTTP POST

- Creates new object



HTTP GET

Étape 2. Après que l'espace ait été créé, extrayez sa configuration.

Utilisez la méthode HTTP GET dans ce but.

Utilisez l'ID de l'espace créé à partir de l'en-tête Location (emplacement). L'adresse URL est <https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae>. Effectuez un **GET sur cette page**.

Exemple de requête 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
```

Réponse pour la requête 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>
```

Note: La réponse est une configuration codée XML de l'espace.

L'image résume la requête **GET**.

HTTP GET

- Retrieves existing information
- No Content in Body



HTTP PUT

Étape 3. Apporter une modification à l'espace (si nécessaire).

Cet exemple montre comment modifier l'espace créé. Supposons qu'une partie utilisateur secondaire doit être ajoutée à l'espace.

Consultez le document de l'API. Il indique que le paramètre nécessaire pour effectuer cela est : **secondaryUri**.

Ajoutez une URI correspondant à asdf. Écrivez une requête semblable à celle créée pour le POST.

Exemple de requête 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

Réponse pour la requête 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
```

Les modifications peuvent être observées dans le WebAdmin de CMS. Naviguez jusqu'à **Configuration > Spaces (Configuration > Espaces)**.

<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

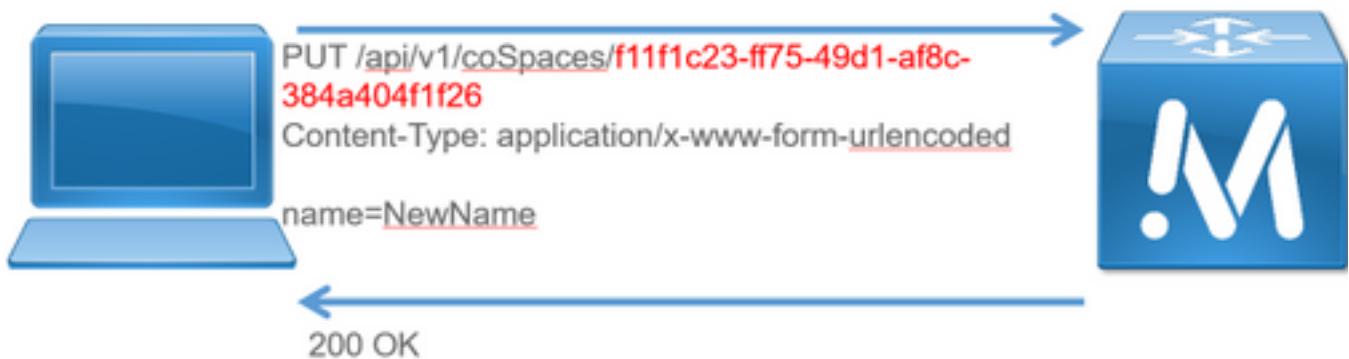
et par l'intermédiaire d'un **GET** :

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

L'image résume la requête 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

Étape 4. Supprimer l'espace (si nécessaire).

La méthode **DELETE** est semblable à la méthode **GET**.

Exemple de requête 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
```

Réponse pour la requête 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
```

Les modifications peuvent être observées dans le WebAdmin de CMS. Naviguez jusqu'à **Configuration > Spaces (Configuration > Espaces)**.

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"/>

et par l'intermédiaire d'un **GET** :

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

L'image résume la demande **DELETE**.

HTTP DELETE

- Destroys an object



Name	URI user part	Call ID

Clients de l'API

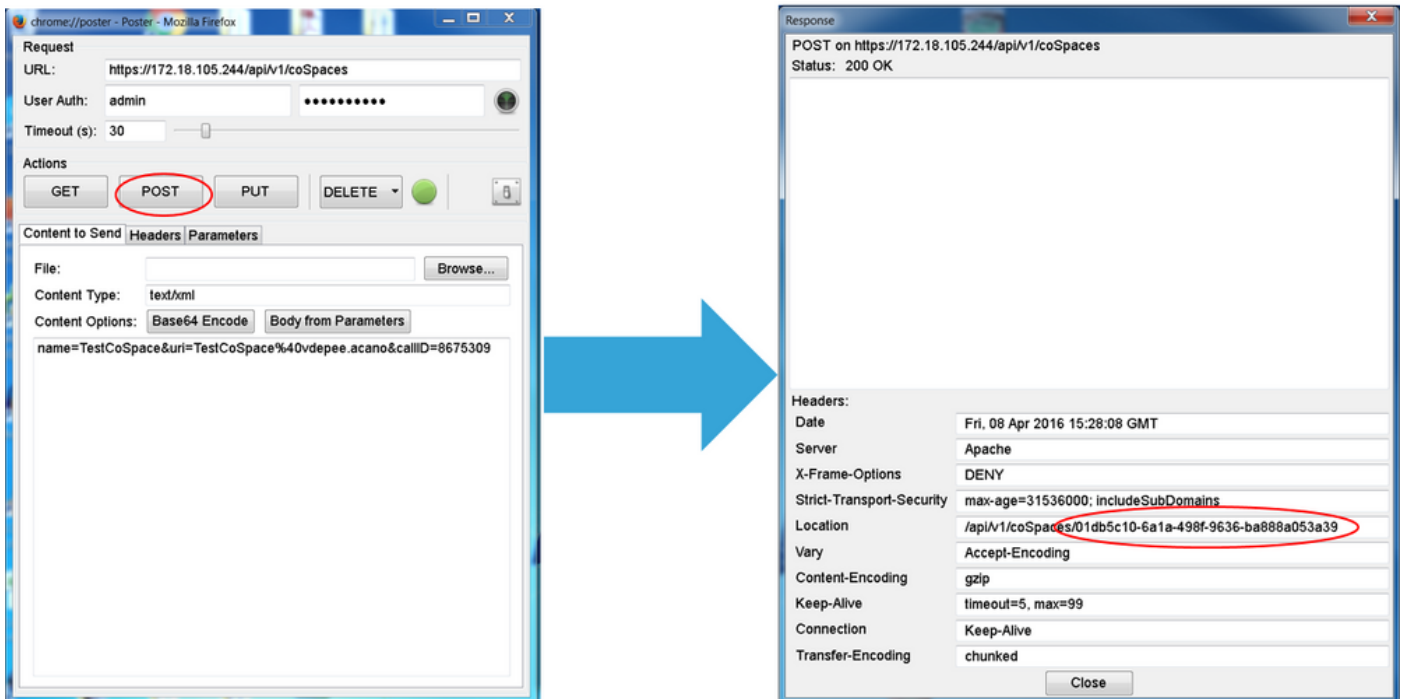
POSTER

La boîte du dessus dans Poster est l'endroit où vous entrez l'URL pour les requêtes.

Les champs User Auth (authentification de l'utilisateur) représentent l'endroit où vous entrez le nom d'utilisateur et le mot de passe dans cet ordre. Ensuite, si vous faites une **GET** ou une

DELETE, choisissez les boutons respectifs. Exemple : cliquez sur **GET** et une fenêtre contextuelle s'affiche avec votre réponse. Pour **DELETE**, assurez-vous que **DELETE est sélectionné**, puis cliquez sur le bouton vert.

Poster (Firefox)



Pour **POST et PUT**, du contenu doit être ajouté. Sélectionnez l'onglet Parameters (paramètres) et ajoutez les noms et les valeurs pour vos paramètres. Revenez ensuite au bouton Contenu à envoyer et choisissez **Corps** dans Paramètres.

Envoyez votre POST ou votre PUT.

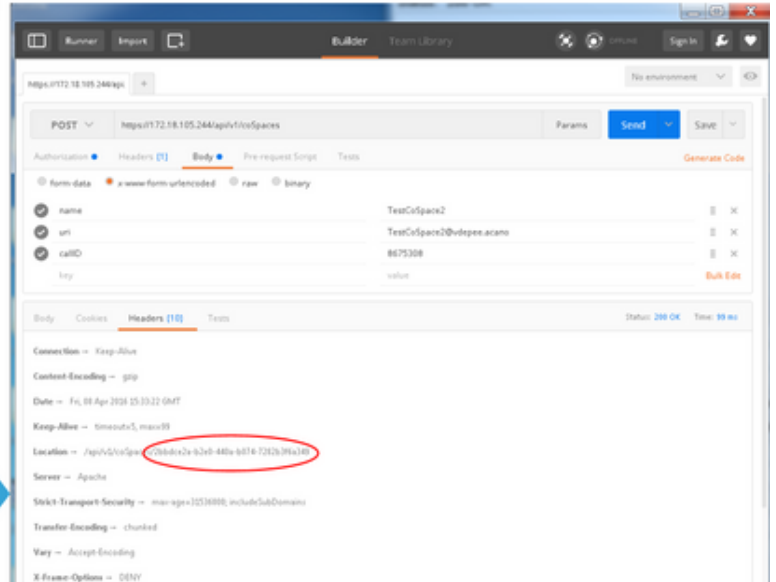
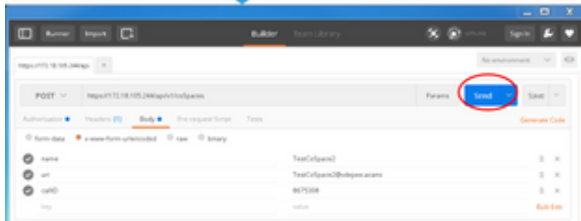
POSTMAN

Dans Postman, en haut à gauche, sélectionnez la méthode que vous souhaitez utiliser dans la liste déroulante et entrez l'URL de demande.

Pour Autorisation, choisissez **Basic-Auth** et entrez votre nom d'utilisateur et votre mot de passe. Ensuite, sélectionnez **Mettre à jour la demande**. Dans l'onglet Headers (en-têtes), vous voyez un en-tête d'autorisation.

Si votre demande est un POST/PUT, accédez à l'onglet Body, choisissez **x-www-form-urlencoded** et entrez vos paramètres et vos valeurs. Lorsque vous avez terminé, sélectionnez **Envoyer**.

Postman (Chrome)



Vérification

La méthode de vérification est expliquée dans chaque demande.

Dépannage

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.