

Cisco Meeting Server Basic API 기능

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[API 요청 및 응답](#)

[구성](#)

[CMS 2.9 이상](#)

[API 개체 구성](#)

[이미 생성된 API 객체 수정](#)

[API 메뉴에서 API 개체 삭제](#)

[설명된 API 요청 \(타사 툴 사용\)](#)

[HTTP POST](#)

[HTTP GET](#)

[HTTP 입력](#)

[HTTP 삭제](#)

[API 클라이언트](#)

[포스터](#)

[포스트맨](#)

[다음을 확인합니다.](#)

[문제 해결](#)

소개

이 문서에서는 CMS(Cisco Meeting Server)에서 사용되는 GET, POST, PUT, DELETE의 4가지 기본 API(Application Program Interface) 기능에 대해 설명합니다. CMS 2.9부터 웹 관리 GUI에는 Configuration(컨피그레이션) 메뉴에서 사용할 수 있는 API 메뉴가 있습니다. 이 문서에서는 새로운 메뉴를 검토하고 두 가지 다른 API 툴을 설명합니다. 포스터 및 Postman과 이를 CMS 구성에 사용하는 방법

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 CMS 2.9 이상 또는 Postman 또는 Poster와 같은 다른 API 클라이언트에서 사용

할 수 있습니다. 이러한 타사 툴은 문서의 API 클라이언트 섹션에서 설명합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

CMS용 API는 여러 기능을 구성할 수 있는 매우 유연한 방법입니다. 여기에는 너무 많은 API 기능을 기억하거나 커버할 수 있으므로 현재 API 참조 설명서를 참조해야 합니다. 이 작성 시점부터 [여기](#)에서 현재 API 참조 가이드를 사용할 수 있습니다.

API 요청 및 응답

API 통신은 클라이언트와 서버 간의 요청-응답 관계입니다. 클라이언트가 서버를 요청합니다. 요청을 처리한 후(작업 완료 또는 거부) 응답이 반환됩니다.

API Request and Response



이 문서에서 설명한 네 가지 요청은 다음과 같습니다.

1. GET - 기존 정보 검색
2. POST - 새 정보 생성
3. PUT - 기존 정보를 수정합니다.
4. DELETE - 기존 정보를 삭제합니다.

CMS를 구성하는 데 사용되는 기본 API 요청입니다.

가장 일반적인 응답은 200입니다. 다른 응답은 4xx 및 5xx이며 오류 응답입니다.

구성

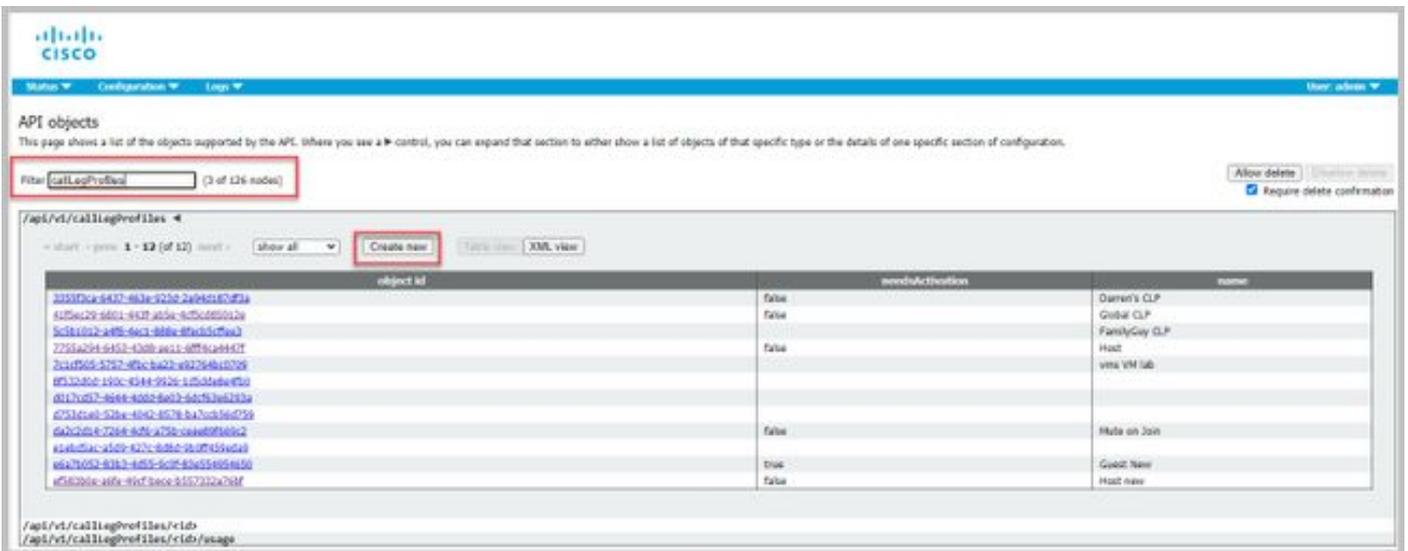
CMS 2.9 이상

CMS 2.9는 관리자가 CMS의 설정을 훨씬 쉽게 수정하고 설정을 세부적으로 조정할 수 있는 새로운 API 메뉴를 선보였습니다. 메뉴를 사용할 때 사용 가능한 모든 매개변수가 표시되므로 설정을 쉽고 빠르게 변경하고 새 기능을 활성화할 수 있습니다.

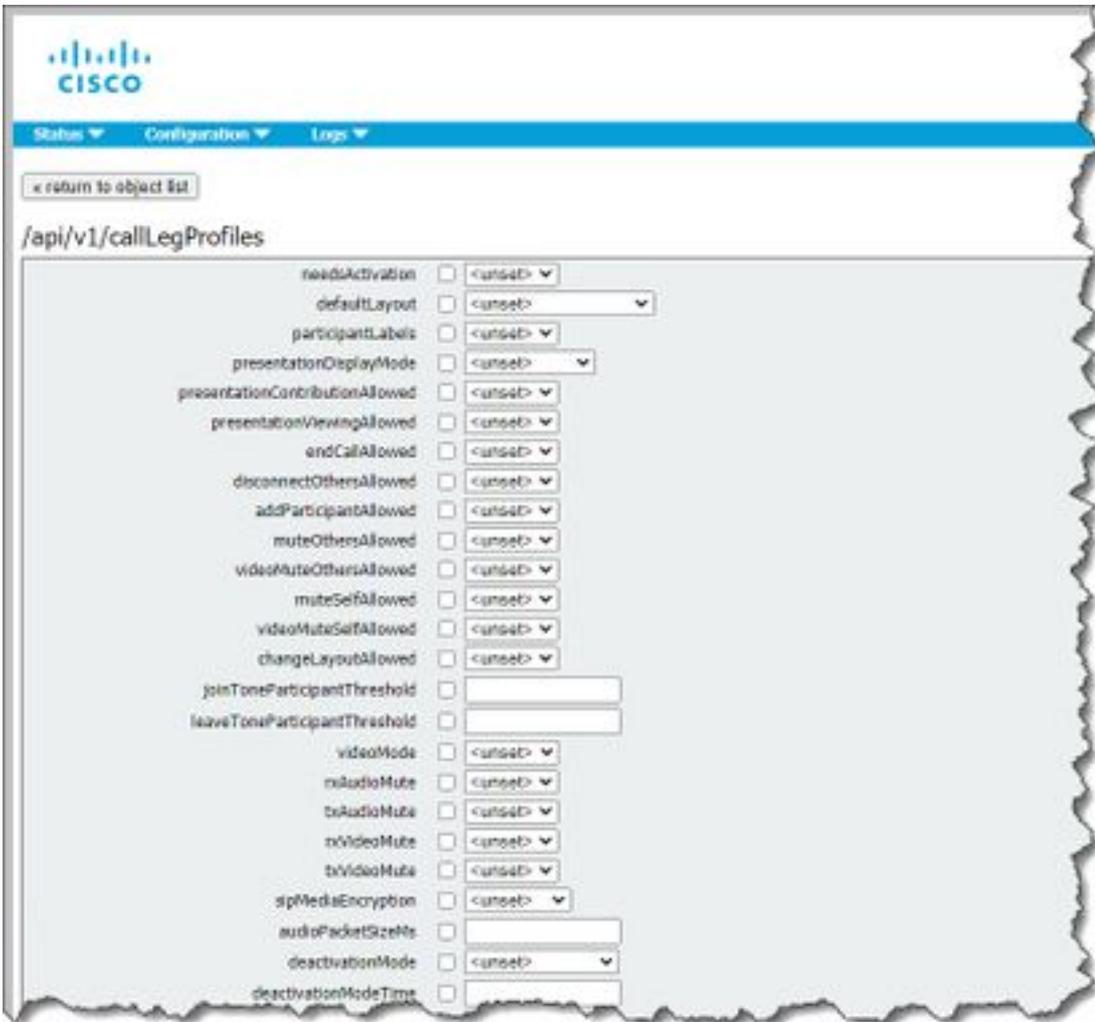


API 개체 구성

API 메뉴에서 API 객체를 수정/생성하려는 항목으로 필터링한 다음 객체 옆에 있는 검은색 화살표를 클릭하여 변경할 수 있습니다. 다음 예에서는 callLegProfiles를 조회하고 새 callLegProfile을 만드는 방법을 보여 줍니다.

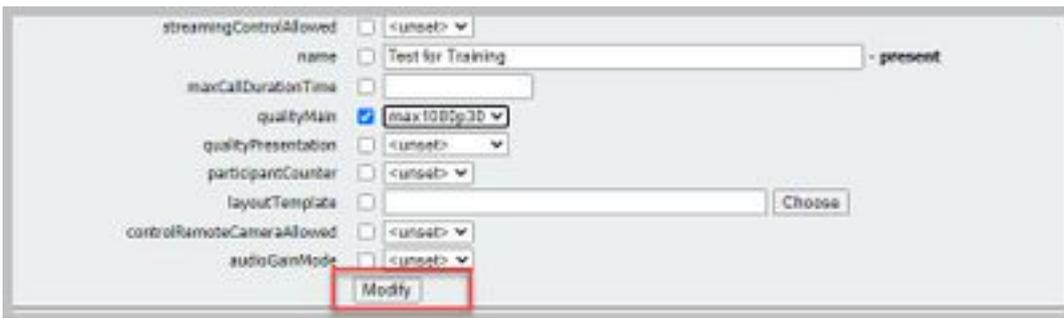


새로 만들기를 클릭하면 CallLegProfiles에 대해 사용 가능한 모든 매개변수를 표시하는 이 화면이 나타납니다. 특정 매개 변수 위에 마우스를 놓으면 각 옵션의 용도를 표시하는 팝업이 표시됩니다.



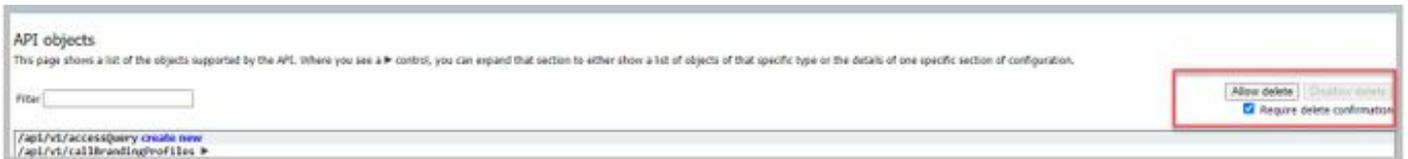
이미 생성된 API 객체 수정

객체의 설정을 변경하면 아래쪽에 수정 단추가 표시됩니다. 이는 타사 툴의 PUT와 동일합니다.



API 메뉴에서 API 객체 삭제

객체를 삭제하려면 기본 객체 목록 페이지에서 항목을 삭제할 수 있는 기능을 활성화할 수 있습니다. 다음 예와 같이 삭제할 옵션을 활성화하려면 Allow delete를 클릭합니다.



Configuration

[return to object list](#)

[/api/v1/callLegProfiles](#)

show all

object id	needsActivation	name	
330572ca-5437-403e-923c-2a1d43876f39	false	Darren's CLP	delete
4375e7d-1871-443f-b07e-4b71c820112e	false	Global CLP	delete
3288911f-8b24-4e2a-b679-d01e228b0e30		Test for Training	delete
5e2b1012-e4f6-4ec3-886e-8fec35c98e31		FamilyGuy CLP	delete
7795a794-8453-4368-ae11-6866c4442f		Host	delete
711c702-5757-4f8c-b323-e927148c0109	false	vms VM lab	delete
8f532866-190c-4544-8926-3d5d8e8e8f0c			delete
e012e57-4844-4dad-8e03-6c453e5233a			delete
e793d1e0-52be-4040-8578-a71cd564799			delete
ca21d14-7104-41f9-a79c-cc8e8f9505c1	false	Mate on Jan	delete
a1ab05ac-8508-452c-8696-8a0c852e8e8d			delete
87a7b252-82b3-4d30-812f-82e379495493f	true	Guest New	delete
af08280e-81f6-49c1-bccc-8237332a796f	false	Host new	delete

설명 된 API 요청 (타사 톨 사용)

4개의 기본 요청은 컨피그레이션 예를 통해 설명합니다.

HTTP POST

1단계. **POST**를 사용하여 객체를 생성합니다.

이 예에서는 이 요청을 사용하여 CMS 공간을 생성합니다.API를 통해 공간을 생성하려면 API 설명서를 참조하십시오.이 예에서는 CMS 2.4 API 가이드를 사용했지만 [여기](#)에 있는 최신 API 가이드를 사용해야 합니다.

섹션 6.2에는 공백을 만들고 수정하는 방법에 대한 정보가 있습니다.

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

첫 번째 문장에서 스페이스를 만들려면 /coSpaces에 게시물을 보내야 합니다.그런 다음 스페이스의 ID가 200 OK의 Location 헤더에 있다고 표시됩니다.좋습니다. 이제 스페이스를 만드는 방법을 알고 있습니다.**POST**를 <https://<WebAdminIP>/api/v1/coSpaces>로 보내면 됩니다.

- **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

POST의 매개변수를 지정합니다.

설명서의 섹션 6.2에서는 사용할 수 있는 모든 매개변수를 나열하는 테이블을 볼 수 있습니다.

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.)

예를 들면 다음과 같습니다. APITest라는 이름과 APITestURI의 URI 사용자 부분으로 스페이스 만들기

콘텐츠 형식은 application/x-www-form-urlencoded이며 콘텐츠는 name=APITest&uri=APITestURI 이 매개변수를 추가하면 이미지에 표시된 대로 요청이 완료됩니다.

```
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

이 그림에서는 이전 요청에 대한 응답을 보여줍니다.

```
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
```

응답의 Location 헤더를 확인합니다.

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

70ca0ed7-4e50-428c-b9ab-4e68faeb86ae는 새로 생성된 공간의 ID입니다. ID는 동일한 스페이스를 대상으로 하는 향후 API 요청을 해야 할 때 유용합니다.

스페이스는 CMS의 WebAdmin에서 볼 수 있습니다. Configuration(컨피그레이션) > Spaces(스페이스)로 이동합니다.

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

이 그림에서는 요청 POST를 요약합니다.

HTTP POST

- Creates new object



HTTP GET

2단계. 스페이스가 생성된 후 해당 컨피그레이션을 가져옵니다.

이 용도로 HTTP GET 메서드를 사용합니다.

Location 헤더에서 생성한 스페이스의 ID를 사용합니다. URL은 <https://<WebAdminIP>/api/v1/coSpaces/70ca0ed7-4e50-428c-b9ab-4e68faeb86ae>입니다. 이 페이지에서 GET을 수행합니다.

GET 요청 예:

```
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
```

GET 요청에 대한 응답:

```
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
```

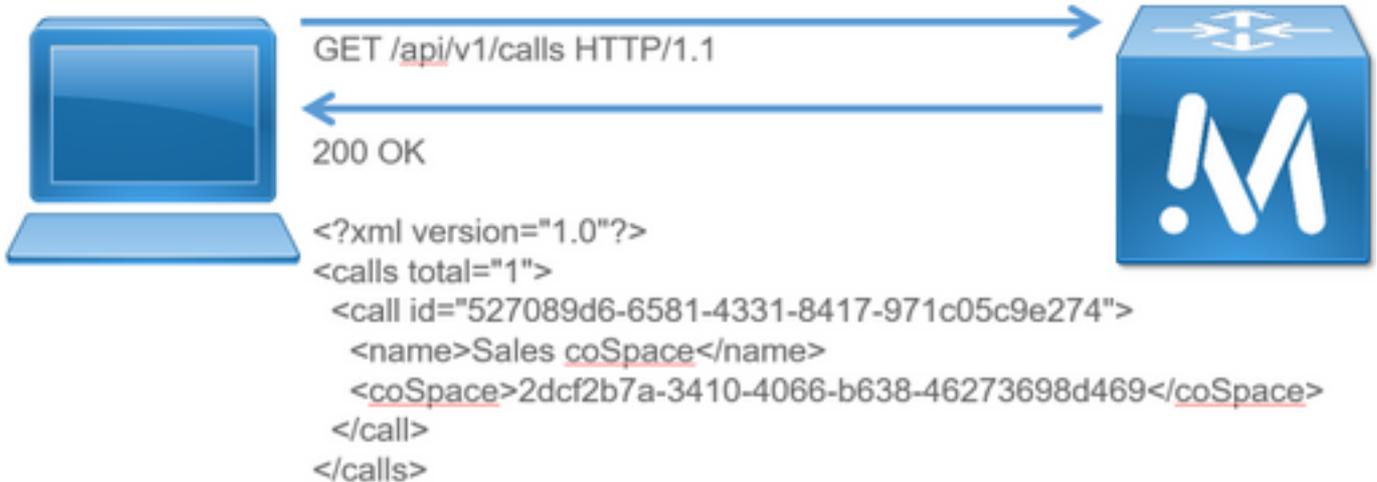
pace>

참고:응답은 스페이스의 XML로 인코딩된 컨피그레이션입니다.

이 그림에서는 요청 GET을 요약합니다.

HTTP GET

- Retrieves existing information
- No Content in Body



HTTP 입력

3단계. 필요한 경우 공간을 변경합니다.

이 예에서는 생성된 공간을 수정하는 방법을 보여 줍니다.스페이스에 보조 사용자 부분을 추가해야 한다고 상상해 보십시오.

API 문서를 참조하십시오.이를 사용하는 데 필요한 매개 변수는 다음과 같습니다.**secondaryUri**.

asdf의 URI를 추가합니다.POST를 위해 생성된 요청과 유사한 요청을 작성합니다.

PUT 요청 예:

```
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

PUT 요청에 대한 응답:

```

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

```

변경 사항은 CMS의 WebAdmin에서 확인할 수 있습니다. Configuration(컨피그레이션) > Spaces(스페이스)로 이동합니다.

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

GET을 통해:

```

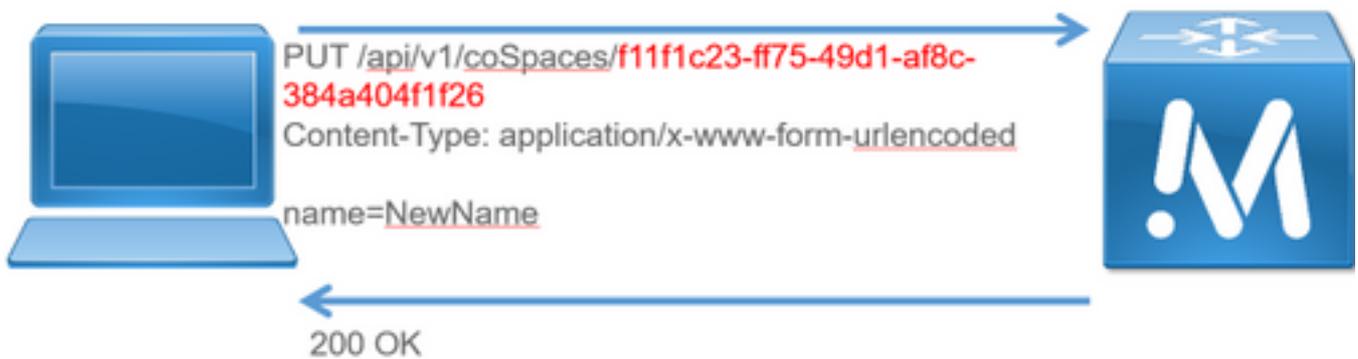
<?xml version="1.0"?><coSpace id="70ca0ed7-4e50-428c-b9ab-4e68faeb86ae"><name>APITest</name><autoGenerated>>false</autoGenerated><uri>apitesturi</uri><secondaryUri>asdf</secondaryUri></coSpace>

```

이 그림에서는 요청 PUT를 요약합니다.

HTTP PUT

- Modifies existing object



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

HTTP 삭제

4단계. 필요한 경우 공간을 삭제합니다.

DELETE 메서드는 GET 메서드와 유사합니다.

DELETE 요청 예:

```

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

```

DELETErequest에 대한 응답:

```

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

```

변경 사항은 CMS의 WebAdmin에서 확인할 수 있습니다.Configuration(컨피그레이션) > Spaces(스페이스)로 이동합니다.

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

GET을 통해:

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

이 그림에서는 요청 DELETE를 요약합니다.

HTTP DELETE

- Destroys an object



Name	URI user part	Call ID

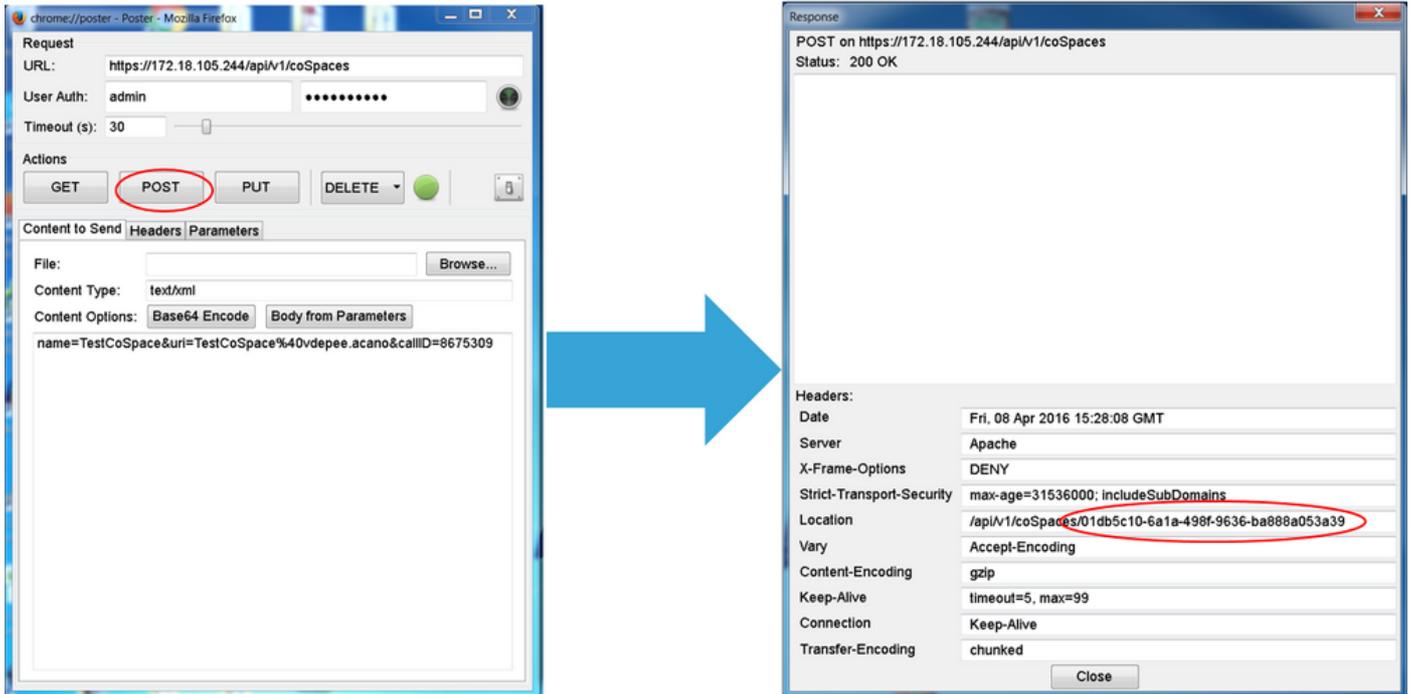
API 클라이언트

포스터

포스터의 맨 위 상자는 요청에 대한 URL을 입력합니다.

User Auth(사용자 인증) 필드는 해당 순서로 사용자 이름과 비밀번호를 입력합니다.그런 다음 GET 또는 DELETE를 수행할 경우 해당 버튼을 선택합니다.예를 들면 다음과 같습니다.GET을 클릭하면 응답이 있는 팝업이 나타납니다.DELETE(삭제)의 경우 DELETE(삭제)가 선택되었는지 확인하고 녹색 버튼을 클릭합니다.

Poster (Firefox)



POST 및 PUT의 경우 콘텐츠를 추가해야 합니다. 매개변수 탭을 선택하고 매개변수의 이름과 값을 추가합니다. 그런 다음 Content to send(전송할 콘텐츠) 단추로 돌아가 Parameters(매개변수)에서 Body(본문)를 선택합니다.

POST 및/또는 PUT를 보냅니다.

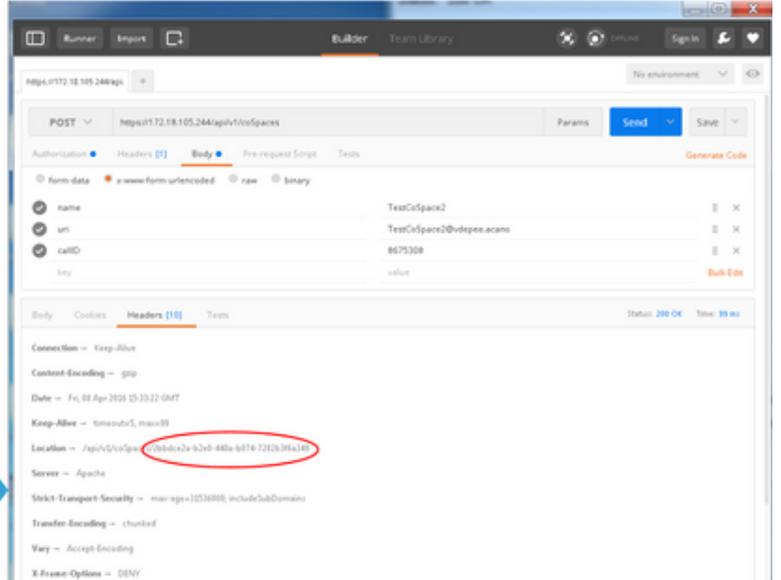
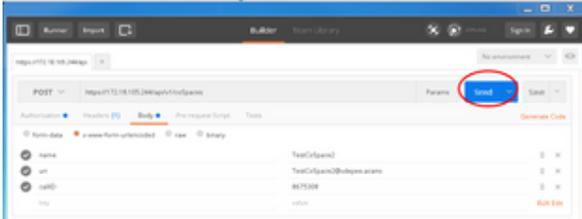
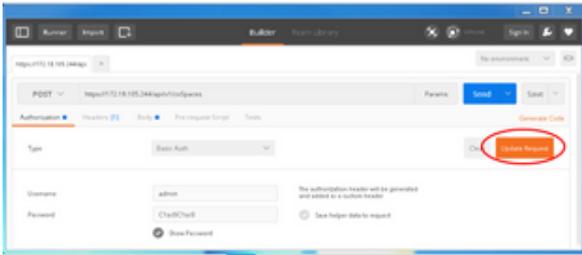
포스트맨

Postman의 왼쪽 상단에 있는 드롭다운 상자에서 사용할 방법을 선택하고 요청 URL을 입력합니다.

Authorization(권한 부여)에서 **Basic-Auth(기본 인증)**를 선택하고 사용자 이름과 비밀번호를 입력합니다. 그런 다음 **Update Request**를 선택합니다. Headers(헤더) 탭에 Authorization Header(권한 부여 헤더)가 표시됩니다.

요청이 POST/PUT인 경우 Body(본문) 탭으로 이동하여 **x-www-form-urlencoded**를 선택하고 매개변수 및 값을 입력합니다. 완료되면 [보내기]를 선택합니다.

Postman (Chrome)



다음을 확인합니다.

확인 방법은 각 요청에 설명되어 있습니다.

문제 해결

현재 이 컨피그레이션에 사용할 수 있는 특정 문제 해결 정보가 없습니다.