

# 配置对 CMS 空间的访客和主机访问并进行故障排除

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[配置](#)

[1\)使用不同URI进行配置](#)

[验证](#)

[2\)使用相同URI但非空访客和主机PIN/密码进行配置](#)

[验证](#)

[3\)使用同一URI的配置，同时包含空访客PIN和非空主机PIN](#)

[验证](#)

[4\)主机用户是空间的成员，通过webRTC登录获得授权，访客用户使用callID加入会议。访客和主机参与者使用相同的URI和callID，访客用户的PIN/密码为空或非空](#)

[验证](#)

[故障排除](#)

[相关信息](#)

## 简介

本文档介绍如何使用 API 命令设置思科 Meeting Server (CMS) 空间的访问权限。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- 思科 Meeting Server (CMS) 空间设置以及能够向其发起呼叫
- API客户端 ( 如Poster、Postman ) 或
- [CMS API 指南](#)

### 使用的组件

本文档中的信息基于 CMS 版本 2.1

本文档中的信息基于特定实验室环境中的设备。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

# 背景信息

本文中会列出以下场景类型：

- 访客和主机参加者使用不同 URI 或呼叫 ID
- 访客和主机参加者使用根据 PIN 或密码条目（均非空）进行区分的相同 URI
- 访客和主机参加者使用根据 PIN 或密码条目（搭配使用空/非空）进行区分的相同 URI
- 主机用户是空间的成员，通过webRTC登录获得授权，访客用户使用callID加入会议。访客和主机参与者使用相同的URI和callID，访客用户的PIN/密码为空或非空

## 配置

在CMS中，访客参与者和主机参与者之间有四种可能的差异，如下面的4个示例中所述，它们主要基于不同的callLegProfiles，该不同的callLegProfiles确定加入空间的参与者的呼叫中行为。

首先，解释为访客和主机参与者使用不同URI（或呼叫ID）的方法，然后，在同一URI上使用不同的密码（或超时）附加该方法，以区分访客和主机参与者。第三种方法为访客用户引入超时或空PIN条目，作为CMS 2.1的新功能，如版本说明第2.4节所示。第四种方法说明如何在已分配所有者/成员的空间上设置访客和主机访问，并使空间（所有者）的成员成为空间的主机。

### 1)使用不同URI进行配置

这是 CMS 2.1 版本之前就有的基本配置，相当于不同的呼叫 ID。需要执行后续步骤才能在同一空间上实现访客/主机访问差异化：

1. 创建访客 callLegProfile (needsActivation = true)
2. 创建主机 callLegProfile (needsActivation = false)
3. 将访客 callLegProfile 分配到现有或新空间（默认的访问方法）
4. 在同一空间上使用不同 URI（和呼叫 ID）创建新的 accessMethod，并向其分配主机 callLegProfile

第 1 步：创建访客 callLegProfile (needsActivation = true)。

callLegProfile确定呼入行为，默认情况下，您会将访客呼入行为分配给空间，以便以后可以在同一空间上使用不同的访问方法，并让主机能够加入。

**注意：**此外，您还可以在租户级别 (/api/v1/tenants/<tenant-ID>) 或系统级别 (/api/v1/system/profiles) 等其他级别进行分配，将此设置应用到所有空间（或每个租户），不过此处只显示空间本身。考虑到呼叫中行为考虑了callLegProfile的最具体分配。

这里，needsActivation 是访客/主机行为最重要的一个参数，因为如果设置为 true，则在一个或多个 full/activator（主机）参加者加入空间之前，参加者无法接收或提供音频和视频。有关 callLegProfile的其他参数，请参阅API指南的第8.4.3节，在该节中，显示的参数也可与此设置相关（取决于您的要求）：

- presentationContributionAllowed
- rxAudioMute
- rxVideoMute
- deactivationMode（停用 |断开连接 |保留激活)和deactivationModeTime [当最后一个激活器离

开呼叫时要执行的操作]

要创建访客callLegProfile，请对/api/v1/callLegProfiles发出POST请求，其中首选参数和需要激活参数设置为true，以便您随后可以对该呼叫LegProfile-ID执行GET请求，例如：

```
needsActivation>true
```

```
needsActivation>
```

```
< deactivationMode>deactivate deactivationMode>
```

请记住**callLegProfile-ID**（标有粗体），因为这必须应用于步骤3中的空间，以便访客访问（默认）。

第2步：创建主机 callLegProfile (needsActivation = false)。

与为主机通话中行为创建主机 callLegProfile 类似。与前面提到的参数相同，但可以根据自己的偏好和要求选择参数。主要是将 needsActivation 参数设置为 false，赋予其主机角色。

您可以通过/api/v1/callLegProfiles的POST请求创建它，该请求使用首选参数，并且需要将激活参数设置为false，以便您随后可以使用以下结果对该callLegProfile-ID执行GET请求，例如：

```
needsActivation>
```

```
false
```

```
needsActivation>
```

请记住**callLegProfile-ID**（标有粗体），因为这必须应用于步骤4中的空间访问方法，以便主机访问。

第3步：将访客 callLegProfile 分配给现有或新空间（默认的 accessMethod）。

对现有空间 (/api/v1/coSpaces/<coSpace-ID>) 执行 PUT 命令修改空间，或者对 /api/v1/coSpaces 执行 POST 命令创建新空间，其中第1步中创建的访客 callLegProfile 参数作为相应空间的通话中行为。此外，您还可以为此空间设置 URI、[passcode](#) 和 call-ID 参数，也可以按照 [《API 指南》](#) 第 6.2 部分的说明根据需要进行设置。

对空间 (/api/v1/coSpaces/<coSpace-ID>) 执行 GET 请求，验证访客 callLegProfile 是否与其关联，以及 URI 和 call-ID 的值。此示例在步骤1中创建的访客callLegProfile的输出示例是URI值为 guest.space 且call-ID 为628821815（未设置密码）的输出示例：

```
uriCallIdcallLegProfile>
```

```
d4bfe12d-68cd-41c0-a671-48395ee170ab
```

```
callLegProfile>
```

请记住空格ID (用粗体标记)，因为这必须用于在第4步中在该特定空间上创建**accessMethod**。

**第4步：使用其他 URI (和 call-ID) 在空间上创建新的 accessMethod，并向其分配主机 callLegProfile。**

您希望创建与访客访问 (当前是默认访客访问) 不同的访问空间方式。通过在 `/api/v1/coSpaces/<coSpace-ID>/accessMethods` 上使用 POST 命令在空间本身上指定 `accessMethod`，其中 `coSpace-ID` 是步骤3(7cc797c9-c0a8-47cf-b中的粗体标记值519-8dc5a01f1ade)，其中应用了步骤2的主机 `callLegProfile`，以及不同的 URI 和呼叫 ID 字段。

在对该空间 `accessMethod(/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>)` 发出 GET 请求后，您必须能够看到与此类输出类似的输出，在此输出中，您可以看到不同的 **URI(host.space)** 和 **call-ID(888)**，与第2步中设置的空间的默认访问方法以及特别关联的主机 `callLegProfile` 相反：

```
uricallIdpasscodecallLegProfile>
```

```
7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912
```

```
callLegProfile>
```

## 验证

现在您可以拨入同一个会议：

- 以访客身份
  - 向 `guest.space` URI (后跟呼叫匹配规则中配置的域名) 拨号
  - 通过 IVR 或 WebRTC 输入 `call-ID` 的值 628821815 加入 (无 passcode)
- 以主机身份
  - 向 `host.space` URI (后跟呼叫匹配规则中配置的域名) 拨号
  - 通过 IVR 或 WebRTC 输入 `call-ID` 的值 888 加入 (无 passcode)

如果只有访客加入空间，系统会将其转至休息室等待主机加入。主机加入后，所有访客和主机都会加入会议。如果空间中不再有主机加入，但仍有一些访客，则根据访客 `callLegProfile` 上的 `deactivate` 参数的 `deactivate` 配置，这些主机返回到大厅屏幕，如步骤1所示。

## 2)使用相同URI但非空访客和主机PIN/密码进行配置

此配置与上例中的配置类似，在CMS 2.1版本之前也可用。它需要访客和主机输入非空 PIN 或密码，才能在向同一 URI 拨号时进行区分。

配置步骤与上一个配置示例非常相似：

1. 创建访客 `callLegProfile` (`needsActivation = true`)
2. 创建主机 `callLegProfile` (`needsActivation = false`)
3. 将访客 `callLegProfile` 分配到指定访客密码 (PIN) 的现有或新空间 ( 默认的访问方法 )
4. 在同一空间上使用相同 URI ( 不同 call-ID ) 创建新的 `accessMethod`，并向其分配主机 `callLegProfile`，包含主机密码 (PIN)

第 1 步：创建访客 `callLegProfile` (`needsActivation = true`)。

与上例1中的配置相同，甚至可以使用相同的访客`callLegProfile(d4bfe12d-68cd-41c0-a671-48395ee170ab)`，如图所示。

第 2 步：创建主机 `callLegProfile` (`needsActivation = false`)

与上例1中的配置相同，甚至可以使用相同的主机`callLegProfile(7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912)`，如图所示。

第 3 步：将访客 `callLegProfile` 分配到指定访客密码 (PIN) 的现有或新空间 ( 默认为 `accessMethod` )。

同样，您也可以对现有空间(`/api/v1/coSpaces/<cospace-ID>`)执行PUT操作，或使用URI、密码和呼叫ID的所需参数(`/api/v1/coSpaces`)创建新空间(`/api/coSpaces`)根据API指南的6.2节分配给它的 `callLegProfile` ( 从第1步开始 )。

如果对该空间执行GET请求，则您必须能够看到类似的输出，其中您看到URI `guestpin.space`、呼叫ID 189、我们之前创建的访客呼叫`LegProfile` 和密码789:

```
uricallIdcallLegProfile>
```

```
    d4bfe12d-68cd-41c0-a671-48395ee170ab
```

```
    callLegProfile><
```

```
    passcode>789
```

```
    passcode>
```

请记住空格ID ( 用粗体标记 )，因为这必须用于在第4步中在该特定空间上创建`accessMethod`。

第 4 步：在空间上使用相同 URI ( 不同 call-ID ) 创建新的 `accessMethod`，并向其分配主机

**callLegProfile**，包含主机密码 (PIN)。

此外，我们还可以在此空间上为主机创建其他访问方法（因为访客 **callLegProfile** 作为默认加入选项分配到此空间本身上），正如第一个配置示例。这可通过对 `/api/v1/coSpaces/<coSpace-ID>/accessMethods` 执行 **POST** 命令完成，其中我们空间的 **coSpace-ID** 值为 **22d9f4ca-8b88-4d11-bba9-e2a2f7428c46**（正如之前步骤中突出显示的那样）。在此 **POST** 命令上，可以提供不同的参数，如 **URI** (`guestpin.space`，与原始参数相同)、**呼叫ID**(889)、**主机呼叫LegProfile**（如步骤 2 所定义）以及**主机密码或PIN**（本例中为 1234）。

如果对该 **accessMethod** 执行 **GET** 请求，则必须能够看到显示相同 **URI** 的类似输出：`guestpin.space`、**call-ID** 889、**主机callLegProfile** 引用和**主机PIN** 1234:

```
uricallId:passcode>1234
```

```
passcode><
```

```
callLegProfile>
```

```
7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912
```

```
callLegProfile>
```

## 验证

现在您可以拨入同一个会议：

- 以访客身份
  - 通过拨打 `guestpin.space` **URI**（后跟在呼叫匹配规则上配置的域）并输入 **PIN** 789
  - 通过 **IVR** 或 **WebRTC** 输入 **call-ID** 值 189 加入，**PIN** 为 789
- 以主机身份
  - 通过拨打 `guestpin.space` **URI**(后跟在呼叫匹配规则上配置的域)并输入 **PIN** 1234
  - 通过 **IVR** 或 **WebRTC** 输入 **call-ID** 值 889 加入，**PIN** 为 1234

如果只有访客加入空间，系统会将其转至休息室等待主机加入。主机加入后，所有访客和主机都会加入会议。如果空间中不再有主机加入，但仍有一些访客，则根据访客 **callLegProfile** 上的 **deactivate** 参数的 **deactivate** 配置，这些主机返回到大厅屏幕，如步骤 1 所示。

## 3)使用同一URI的配置，同时包含空访客PIN和非空主机PIN

此配置仅从 **CMS** 版本 2.1 开始可用，因为 **callProfile** 部分新增了一些 **API** 命令 **passcodeMode** 和 **passcodeTimeout**。这样可在主机使用 **PIN** 访问空间并激活时，为访客提供空 **PIN** 加入（输入 # 或

超时)。callProfile 可以控制 SIP ( 包括 Lync ) 呼叫的通话中体验，因此不适合 CMA 客户端 ( 厚客户端和 WebRTC )。

配置步骤与示例 2 中的步骤类似，但增加了 callProfile：

1. 创建访客 callLegProfile (needsActivation = true)
2. 创建主机 callLegProfile (needsActivation = false)
3. 使用所需的 passcodeMode 和 passcodeTimeout 配置创建 callProfile
4. 将第 3 步的访客 callLegProfile 和 callProfile 分配到指定访客密码 (PIN) 的现有或新空间 (默认访问方法)
5. 在同一空间上使用相同 URI (不同 call-ID) 创建新的 accessMethod，并向其分配主机 callLegProfile，包含主机密码 (PIN)

由于配置与配置示例 1 和 2 非常相同，因此有参考。实际上，在测试中，使用的空间与示例 2 中相同，但现在已添加 callProfile。

第 1 步：创建访客 callLegProfile (needsActivation = true)。

与上一个示例 1 中的配置相同，甚至可以使用相同的访客 callLegProfile(d4bfe12d-68cd-41c0-a671-48395ee170ab)。

第 2 步：创建主机 callLegProfile (needsActivation = false)。

与上一个示例 1 中的配置相同，甚至可以使用相同的主机 callLegProfile(7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912)。

第 3 步：使用所需的 passcodeMode 和 passcodeTimeout 配置创建 callProfile。

您可以创建用于确定 SIP ( 包括 Lync ) 呼叫的通话中体验的 callProfile。这里有几种可能的配置，例如允许录音、流传输或参加者上限，但这里重点介绍 CMS 2.1 中与密码处理相关的新增的 API 功能。其他参数可参见 [《API 指南》](#) 第 8.2 部分。

此处有两个用于确定密码行为的参数：

- **passcodeMode**

- **必需**：IVR 会永远等待用户输入 PIN 或 # 以输入空 PIN ( 对于访客 )

- **超时**：IVR 等待 *passcodeTimeout* 秒数让参与者输入 PIN，如果在该时间内未输入任何条目，则它假定已输入空白 (#) PIN

- **passcodeTimeout**：仅在 passcodeMode 设置为超时时进行设置，可控制将密码解释为空白之前的时间

要创建 callProfile，请对 /api/v1/callProfiles 执行 POST 命令 ( 或者要修改现有的 callProfile，则对 /api/v1/callProfiles/<callProfile-ID> 执行 PUT )，并为 passcodeMode 和 passcodeTimeout 设置所需的参数值。如果对特定的 callProfile 执行 GET 命令，必须获取类似结果，即将模式设置为超时，并且超时值为 5 秒：

```
passcodeMode>timeout
```

```
passcodeMode><
```

```
passcodeTimeout>5
```

```
passcodeTimeout>
```

请记住**callProfile-ID** (用粗体标记)，因为这必须用于分配给空间，以便在步骤4中执行此呼入行为。

第 4 步：将第 3 步的访客 **callLegProfile** 和 **callProfile** 分配到指定访客密码 (PIN) 的现有或新空间 (默认的访问方法)。

与之前类似，您可以对现有空间 (`/api/v1/coSpaces/<cospace-ID>`) 执行 PUT 操作，或者执行 POST 操作创建新空间 (`/api/v1/coSpaces`)，为 URI 和 call-ID 等参数以及访客 **callLegProfile** (通过第 1 步) 设置所需的参数值。与之前示例的不同之处在于第 3 步中创建的 **callProfile**，以及未向其分配密码这一事实。

如果对该空间执行 GET 请求，则必须能够看到与本例类似的输出，其中您会看到 URI `guestpin.space`、**call-ID** 189、先前创建的访客 **callLegProfile** 和 **callProfile**，如步骤 3 中所设置：

```
uricallIdcallLegProfile>
```

```
d4bfe12d-68cd-41c0-a671-48395ee170ab
```

```
callLegProfile><
```

```
callProfile>
```

```
4b0eff60-e4aa-4303-8646-a7e800a4eac6
```

```
callProfile>
```

请记住空间 ID (标有粗体)，因为必须使用它在步骤 5 中的该特定空间上创建 `accessMethod`。

第 5 步：在同一空间上使用相同 URI (不同 **call-ID**) 创建新的 `accessMethod`，并向其分配主机 **callLegProfile**，包含主机密码 (PIN)。

此外，我们还可以在此空间上为主机创建其他访问方法 (因为访客 **callLegProfile** 作为默认加入选项分配到此空间本身上)，正如第一个配置示例。这可通过对 `/api/v1/coSpaces/<coSpace-ID>/accessMethods` 执行 POST 命令完成，其中 `coSpace-ID` 值替换为空间的值 `22d9f4ca-8b88-4d11-bba9-e2a2f7428c46` (正如之前步骤中针对这一情况突出显示的那样)。在此 POST 命令中，可以提供不同的参数，如 URI (`guestpin.space`，与原始参数相同)、**call-ID**(889)、主机 **callLegProfile** (如步骤 2 所定义) 以及主机密码或 PIN (本例中为 1234)。



如果对该accessMethod执行GET请求，则必须能够看到显示相同URI的类似输出:guestpin.space、call-ID 889、主机callLegProfile引用和主机PIN 1234:

```
uricallId>passcode>1234
```

```
passcode><
```

```
callLegProfile>
```

```
7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912
```

```
callLegProfile>
```

## 验证

现在您可以拨入同一个会议：

- 以访客身份
  - 通过拨打guestpin.space URI（后跟在呼叫匹配规则上配置的域），并输入#作为PIN或让它在5秒后超时
    - 通过 IVR 或 WebRTC 输入 call-ID 值 189 加入
- 以主机身份
  - 通过拨号至guestpin.space URI（后跟在呼叫匹配规则上配置的域）并输入PIN 1234
    - 通过 IVR 或 WebRTC 输入 call-ID 值 889 加入，PIN 为 1234

**4)主机用户是空间的成员，通过webRTC登录获得授权，访客用户使用callID加入会议。访客和主机参与者使用相同的URI和callID，访客用户的PIN/密码为空或非空**

需要执行后续步骤，才能在同一空间上为该空间的成员和非成员区分访客/主机访问：

1. 创建访客 callLegProfile (needsActivation = true)
  2. 创建主机 callLegProfile (needsActivation = false)
  3. 将访客 callLegProfile 分配到现有或新空间（默认的访问方法）
  4. 在具有相同URI（和呼叫ID）的相同空间上创建新的accessMethod，并为其分配主机 callLegProfile。
  5. 将用户的ownerJID分配到相同的空间。（如果未分配）
  6. 将该ownerID作为成员用户添加到相同的空间，并将hostcallLegProfile分配给该成员用户
- 第 1 步：创建访客 callLegProfile (needsActivation = true)。**

与上例1和访客callLegProfile(bfe7d07f-c7cb-4e90-a46e-4811bbaf6978)的配置相同。

请记住**callLegProfile-ID** ( 标有粗体 ) , 因为这必须应用于步骤3中的空间 , 以便访客访问。

## 第 2 步 : 创建主机 callLegProfile (needsActivation = false)

本示例中使用与上一个示例1和主机callLegProfile(0e76e943-6d90-43df-9f23-7f1985a74639)相同的配置。

请记住**callLegProfile-ID** ( 标有粗体 ) , 因为这必须应用于步骤4中用于主机访问的**accessMethod**和步骤6中的coSpace成员。

**步骤3.**将访客callLegProfile分配给现有或新空间 ( 即默认accessMethod ) 。

对现有空间(/api/v1/coSpaces/<coSpace-ID>)执行PUT命令以调整空间 , 或对/api/v1/coSpaces执行POST命令以创建新的访客callLegProfile参数 , 如步骤1中所创建 , 作为呼入行为为那个空间。您还可以根据API指南的第6.2节为该空间设置URI和call-ID参数 , 以及根据您的[需求设置](#)。

对该空间(/api/v1/coSpaces/<coSpace-ID>)执行GET请求 , 以验证访客callLegProfile是否与其关联 , 以及URI和呼叫ID值。在步骤1中 , 使用此示例创建的访客callLegProfile的输出示例是URI值为**global**,call-ID 为1234 ( 未设置密码 ) , nonMemberAccessset totrue:

```
<?xml version="1.0" ?>
<coSpace id="96d28acb-86c6-478d-b81a-a37ffb0adafc">
  <name>Global</name>
  <autoGenerated>>false</autoGenerated>
  <uri>global</uri>
  <callId>1234</callId>
  <callLegProfile>bfe7d07f-c7cb-4e90-a46e-4811bbaf6978</callLegProfile>
  <nonMemberAccess>true</nonMemberAccess>
  <secret>0w4O2zTTF0WdL4ymF8D0_A</secret>
  <defaultLayout>allEqual</defaultLayout>
</coSpace>
```

请记住空格ID ( 用粗体标记 ) , 因为这必须用于在第4步中在该特定空间上创建**accessMethod**。

**步骤4.**在具有相同URI ( 和呼叫ID ) 的该空间上创建新访问方法 , 并为其分配主机callLegProfile。

您希望创建与访客访问 ( 当前是默认访客访问 ) 不同的访问空间方式。通过在步骤3(96d28acb-86c6-477)中对/api/v1/coSpaces/<coSpace-ID>/accessMethods指定accessMethod来在空间本身指定accessMethodd-b81a-a37ffb0adafc) , 其中应用了第2步的主机callLegProfile , 以及相同的URI和call-ID字段。您可以为通过callID连接的主机添加非空密码 ( 无需以用户身份通过webRTC登录 ) 。

在该空间accessMethod(/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>)上发出GET请求后 , 您必须能够看到与此类输出类似的输出 , 在此输出中 , 您可以看到相同URI ( 全局 ) 和call-ID1234) , 以及步骤2中设置的特别关联的主机callLegProfile ( 如下 ) 和主机密码 (12345):

```
<?xml version="1.0" ?>
<accessMethod id="c4ecc16e-945f-4e35-ba03-d9b69107b32c">
  <uri>global</uri>
  <callId>1234</callId>
  <passcode>12345</passcode>
  <callLegProfile>0e76e943-6d90-43df-9f23-7f1985a74639</callLegProfile>
  <secret>kff01zTTE0feL4fsdf43w_B </secret>
</accessMethod>
```

**步骤5.**分配用户的所有者跳至空间。 ( 如果未分配 ) 。 在空间上通过aPUT命令在

/api/v1/coSpaces/<coSpace-ID>指定yingownerJid([user1@evacanoalone.net](mailto:user1@evacanoalone.net))将ownerJID添加到空间

在该空间上发出GET请求后，您必须能够看到已将ownerIdand ownerJidhave分配给该空间：

```
<?xml version="1.0" ?>
<coSpace id="96d28acb-86c6-478d-b81a-a37ffb0adafc">
  <name>Global</name>
  <autoGenerated>>false</autoGenerated>
  <uri>global</uri>
  <callId>1234</callId>
  <callLegProfile>bfe7d07f-c7cb-4e90-a46e-4811bbaf6978</callLegProfile>
  <nonMemberAccess>>true</nonMemberAccess>
  <ownerId>1d942281-413e-4a2a-b776-91a674c3a5a9</ownerId>
  <ownerJid>user1@evacanoalone.net</ownerJid>
  <secret>0w402zTTF0WdL4ymF8D0_A</secret>
  <numAccessMethods>1</numAccessMethods>
  <defaultLayout>allEqual</defaultLayout>
</coSpace>
```

记下ownerID(1d942281-413e-4a2a-b776-91a674c3a5a9)。

**步骤6.**将步骤5中的ownerID(1d942281-413e-4a2a-b776-91a674c3a5a9)作为成员用户添加到空间，并为该成员用户分配hostcallLegProfile。通过POST命令 (/api/v1/coSpaces/<coSpaceID>/coSpaceUsers)在空间本身（指定coSpaceID）中指定指定 userJidandhost callLegProfile来完成此操作。**co上的其他参数SpaceUsers**可在API指南的6.4.2部分找到，在该部分中，所显示的用户也可与此设置相关：

```
<canDestroy>>true</canDestroy>

<canAddRemoveMember>>true</canAddRemoveMember>

<canChangeName>>true</canChangeName>

<canChangeUri>>false</canChangeUri>

<canChangeCallId>>false</canChangeCallId>

<canChangePasscode>>true</canChangePasscode>

<canPostMessage>>true</canPostMessage>

<canDeleteAllMessages>>false</canDeleteAllMessages>

<canRemoveSelf>>false</canRemoveSelf>
```

已通过aGET命令(/api/v1/coSpaces/<coSpaceID>/coSpaceUsers?)将该成员用户添加到空间中，对此进行验证。

```
<?xml version="1.0" ?>
<coSpaceUsers total="1">
<coSpaceUser id="1d942281-413e-4a2a-b776-91a674c3a5a9">
  <userId>1d942281-413e-4a2a-b776-91a674c3a5a9</userId>
  <userJid>user1@evacanoalone.net</userJid>
  <autoGenerated>>false</autoGenerated>
```

```
</coSpaceUser>
</coSpaceUsers>
```

记下userID ( 如果表单ownerID表单步骤5不同 ) 。验证是否已通过aGET请求将coSpaceID和userID(/api/v1/coSpaces/<coSpaceID>/coSpaceUsers/<userID>)分配给coSpaceUser

```
<?xml version="1.0" ?>
<coSpaceUser id="1d942281-413e-4a2a-b776-91a674c3a5a9">
```

```
  <autoGenerated>false</autoGenerated>
  <canDestroy>true</canDestroy>
  <canAddRemoveMember>true</canAddRemoveMember>
  <canChangeName>true</canChangeName>
  <canChangeUri>false</canChangeUri>
  <canChangeCallId>false</canChangeCallId>
  <canChangePasscode>true</canChangePasscode>
  <canPostMessage>true</canPostMessage>
  <canDeleteAllMessages>false</canDeleteAllMessages>
  <canRemoveSelf>false</canRemoveSelf>
  <canChangeNonMemberAccessAllowed>true</canChangeNonMemberAccessAllowed>
```

```
    0e76e943-6d90-43df-9f23-7f1985a74639
```

```
</coSpaceUser>
```

## 验证

现在您可以拨入同一个会议：

- 以访客身份

- 通过拨号到URI ( 后跟在呼叫匹配规则上配置的域 )

- 通过 IVR 或 WebRTC 输入 call-ID 的值 1234 加入 ( 无 passcode )

- 以主机身份

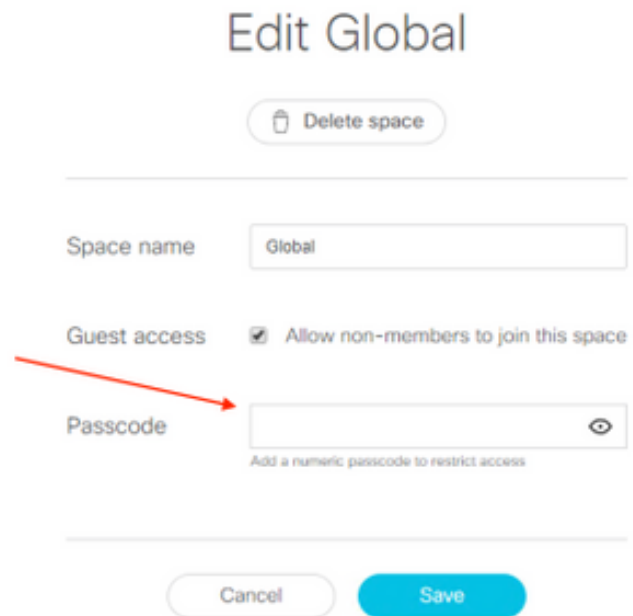
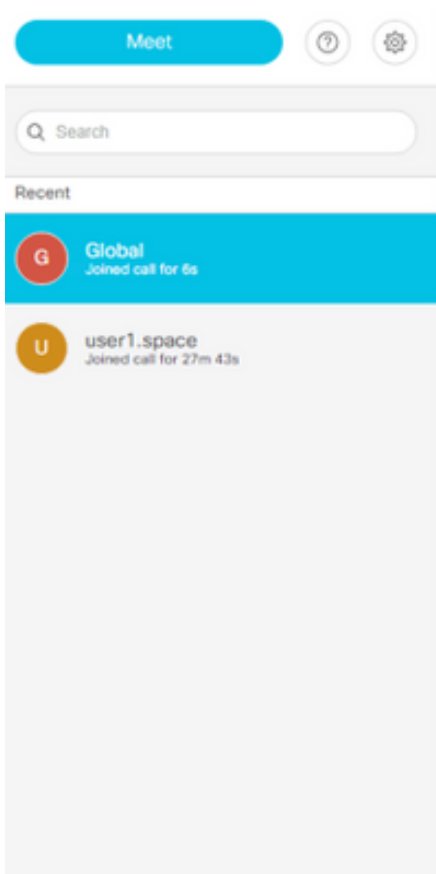
通过webRTC以用户身份(分配了“host” callLegProfile的空间成员，在此场景中为user1@evacanoalone.net)登录并加入空间(“global”URI)。

- 通过拨号到“全局”URI ( 后跟呼叫匹配规则上配置的域 ) 和密码12345。

- 通过IVR或WebRTC加入输入call-ID值1234(主机密码为12345)

如果只有访客加入空间，系统会将其转至休息室等待主机加入。主机加入后，所有访客和主机都会加入会议。如果空间中不再有主机加入，但仍有一些访客，则根据访客callLegProfile上的deactivate参数的deactivate配置，这些主机返回到大厅屏幕，如步骤1所示。

主机 ( 所有者/成员 ) 可以直接在WebRTC应用中为访客设置 ( 编辑/删除 ) 密码，或完全禁用非成员 ( 访客 ) 访问空间：



## 故障排除

本部分提供了可用于对配置进行故障排除的信息。

CMS 的日志记录并未向我们简单显示您何时以访客身份加入或者第一个主机何时加入，但最好使用 callProfile 的 GET 请求以及访客和主机 callLegProfile 定义，以及它们在各自 accessMethods（或默认访问方法）或更高级别（全局级别或租户级别）的分配进行验证。

您可以按照此结构获取所有信息：

1. 对 `/api/v1/callProfiles` 执行 GET 操作（如果使用 `passcodeMode`）  
>使用GET on `/api/v1/callProfiles/<callProfile-ID>`详细验证所需的callProfile-ID
2. 对 `/api/v1/callLegProfiles` 执行 GET 操作  
>使用GET on `/api/v1/callProfiles/<callProfile-ID>`，详细验证访客和主机所需的callLegProfile-ID
3. 对 `/api/v1/coSpaces` 执行 GET 操作  
>在 `/api/v1/coSpaces/<coSpace-ID>`上使用GET详细验证所需的空间ID  
>检查所需的callProfile-ID（步骤1）和访客callLegProfile（步骤2）是否与此空间关联  
[如果未关联到此空间，请检查租户 (`/api/v1/tenants/<tenant-ID>`) 或全局 (`/api/v1/system/profiles`) 级别等范围更广的元素]
4. 对 `/api/v1/coSpaces/<coSpace-ID>/accessMethods` 执行 GET 操作  
>详细验证在 `/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>`上使用GET来检查是否已分配主机callLegProfile。

在示例中显示的CMS日志记录中，您有前两个访客参与者进入(从2000@steven.lab呼叫38，从1060@steven.lab呼叫39)，他们超时到guestpin.space@acano.steven.lab空间，然后主机加入。从

代码片段中可以看到，对于访客，代码会提示我们需要对其执行什么操作（禁用），并且我们可以看到这些呼叫的此行为会在主机 (stejanss.movi@steven.lab) 加入空间时更改（停止禁用）。同样，当不再有主机加入空间，并且访客重新回到休息区后，您会再次看到相同的日志记录（禁用）。

```
2017-02-21 17:48:54.809 Info      call 38: incoming encrypted SIP call from
"sip:2000@steven.lab" to local URI "sip:guestpin.space@acano.steven.lab" 2017-02-21 17:48:54.822
Info call 38: setting up UDT RTP session for DTLS (combined media and control) 2017-02-21
17:48:54.837 Info call 38: compensating for far end not matching payload types 2017-02-21
17:48:54.847 Info sending prompt response (2) to BFCP message 2017-02-21 17:48:54.847 Info call
38: sending BFCP hello as client following receipt of hello when BFCP not active 2017-02-21
17:48:54.883 Warning call 38: replacing pending BFCP message "PrimitiveHelloAck" with
"PrimitiveHelloAck" 2017-02-21 17:48:54.883 Info call 38: BFCP (client role) now active 2017-02-
21 17:48:59.294 Info call 39: incoming encrypted SIP call from "sip:1060@steven.lab" to local
URI "sip:guestpin.space@acano.steven.lab" 2017-02-21 17:48:59.310 Info call 39: setting up UDT
RTP session for DTLS (combined media and control) 2017-02-21 17:48:59.323 Info call 39:
compensating for far end not matching payload types 2017-02-21 17:48:59.569 Info sending prompt
response (2) to BFCP message 2017-02-21 17:48:59.569 Info call 39: sending BFCP hello as client
following receipt of hello when BFCP not active 2017-02-21 17:48:59.746 Info call 39: BFCP
(client role) now active 2017-02-21 17:49:07.971 Info configuring call e2264fb0-483f-45bc-a4f3-
5a4ce326e72c to be deactivated
2017-02-21 17:49:07.972 Info      participant "2000@steven.lab" joined space 22d9f4ca-8b88-
4d11-bba9-e2a2f7428c46 (Guest/Host PIN)
2017-02-21 17:49:12.463 Info      configuring call b1b5d433-5ab5-49e1-9ae3-3f4f71703d1b to be
deactivated
2017-02-21 17:49:12.463 Info      participant "1060@steven.lab" joined space 22d9f4ca-8b88-
4d11-bba9-e2a2f7428c46 (Guest/Host PIN)
2017-02-21 17:49:12.463 Info      conference "Guest/Host PIN": unencrypted call legs now
present
2017-02-21 17:49:16.872 Info      call 40: incoming encrypted SIP call from
"sip:stejanss.movi@steven.lab" to local URI "sip:guestpin.space@acano.steven.lab" 2017-02-21
17:49:16.885 Info call 40: setting up UDT RTP session for DTLS (combined media and control)
2017-02-21 17:49:24.260 Info call 40: audio prompt play time out 2017-02-21 17:49:26.670 Info
participant "stejanss.movi@steven.lab" joined space 22d9f4ca-8b88-4d11-bba9-e2a2f7428c46
(Guest/Host PIN)
2017-02-21 17:49:26.670 Info      call e2264fb0-483f-45bc-a4f3-5a4ce326e72c ceasing to be
deactivated
2017-02-21 17:49:26.670 Info      call b1b5d433-5ab5-49e1-9ae3-3f4f71703d1b ceasing to be
deactivated
2017-02-21 17:49:30.832 Info      call 40: ending; remote SIP teardown - connected for 0:14
2017-02-21 17:49:30.833 Info      participant "stejanss.movi@steven.lab" left space 22d9f4ca-
8b88-4d11-bba9-e2a2f7428c46 (Guest/Host PIN)
2017-02-21 17:49:30.833 Info      configuring call e2264fb0-483f-45bc-a4f3-5a4ce326e72c to be
deactivated
2017-02-21 17:49:30.833 Info      configuring call b1b5d433-5ab5-49e1-9ae3-3f4f71703d1b to be
deactivated
```

## 相关信息

- [技术支持和文档 - Cisco Systems](#)
- [CMS文档](#)