

# Configurar e solucionar problemas de acesso do convidado e do host em espaços do CMS

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[1\) Configuração usando URIs diferentes](#)

[Verificar](#)

[2\) Configuração usando os mesmos URIs, mas PINs/senhas de host e convidado não vazios](#)

[Verificar](#)

[3\) Configuração usando os mesmos URIs com combinação de PIN de convidado vazio e PIN de host não vazio](#)

[Verificar](#)

[4\) Um usuário host é um membro do espaço e autorizado por meio do logon do WebRTC, os usuários convidados participam da reunião com callID. O mesmo URI e callID são usados por participantes convidados e hosts com PIN/senhas vazias ou não vazias para usuários convidados](#)

[Verificar](#)

[Troubleshoot](#)

[Informações Relacionadas](#)

## Introduction

Este documento descreve como configurar o acesso do host e do convidado em espaços de seu Cisco Meeting Server (CMS) usando os comandos da API.

## Prerequisites

### Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Cisco Meeting Server (CMS) com espaços configurados e capazes de fazer chamadas
- Cliente API (como Poster, Postman) ou
- [Guia da API do CMS](#)

### Componentes Utilizados

As informações neste documento têm como base no CMS versão 2.1

As informações neste documento foram criadas em um dispositivo de um ambiente de laboratório específico. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Informações de Apoio

O documento destaca tipos de situações:

- Diferentes URIs ou IDs de chamada são usados pelo host e o convidado participantes
- A mesma URI é usada pelos participantes convidados e participantes host, sendo que o diferencial é feito com base no PIN ou senha inserida (ambos não em branco)
- A mesma URI é usada pelos participantes convidados e participantes host, sendo que o diferencial é feito com base no PIN ou senha inserida (mistura de em branco/não em branco)
- Um usuário host é um membro do espaço e autorizado por meio do logon do WebRTC, os usuários convidados entram na reunião com callID. O mesmo URI e callID são usados por participantes convidados e hosts com PIN/senhas vazias ou não vazias para usuários convidados

## Configurar

Há quatro possibilidades de diferenciação entre participantes Convidado e Host no CMS, descritas nos 4 exemplos seguintes, e são baseadas principalmente em diferentes **callLegProfiles** que determinam o comportamento de chamada para os participantes que se juntam no espaço.

Primeiro, explica-se o método usando uma **URI** diferente (ou call-ID) para convidados e participantes de host, e depois ele é adicionado usando diferentes senhas (ou timeout) na mesma **URI**, para fazer a diferenciação entre participantes de host e convidados. O terceiro método de uma entrada de PIN de tempo limite ou vazia para usuários convidados foi apresentado como um novo recurso no CMS 2.1, como mostrado na seção 2.4 das [notas de versão](#). O quarto método explica como configurar o acesso de convidado e host em espaços com proprietário/membros atribuídos e tornar o membro do espaço (proprietário) o host do espaço.

### 1) Configuração usando URIs diferentes

Essa é a configuração básica disponibilizada antes da versão 2.1 do CMS e é a mesma para uma ID de chamada diferente. As próximas etapas precisam ser executadas para obter a diferenciação de acesso de convidado/host no mesmo espaço:

1. Crie um convidado **callLegProfile** (**needsActivation = true**)
2. Crie um host **callLegProfile** (**needsActivation = false**)
3. Atribua o convidado **callLegProfile** para um espaço novo ou existente (sendo o método de **acesso padrão**)
4. Crie um novo **accessMethod** naquele mesmo espaço com diferentes **URI** (e ID da chamada) e atribua o **host callLegProfile** a ele

Etapas 1. Crie um convidado **callLegProfile** (**precisaAtivation = verdadeiro**).

Um **callLegProfile** determina o comportamento na chamada e, por padrão, você atribui o comportamento da chamada do convidado ao espaço para que você possa mais tarde ter um

método de acesso diferente nesse mesmo espaço, assim como para que o host possa participar.

**Note:** Você também pode atribuir isso no nível de locatário (`/api/v1/espacos/<tenant-ID >`) ou nível de sistema (`/api/v1/system/profiles`) por exemplo, para aplicar a todos os espaços (ou por espaço), no entanto, aqui, ele é mostrado em um espaço em si. Considere que a alocação mais específica do **callLegProfile** é levada em conta para o comportamento na chamada.

O parâmetro **needsActivation** é o mais importante para o comportamento de convidado/host, caso **esteja definido como true**, o participante não consegue receber ou contribuir com áudio e vídeo até que um ou mais participantes **full/ativador (host) entrem**. Outros parâmetros no **callLegProfile** podem ser encontrados na seção 8.4.3 do [guia de API](#), sob a qual os parâmetros mostrados também podem ser relevantes nesta configuração (dependendo de seus requisitos):

- presentationContributionAllowed
- rxAudioMute
- rxVideoMute
- deactivationMode (deactive | desconectar | continueAtivado) e deactivationModeTime [ação a ser executada quando o último ativador sair da chamada]

Para criar o convidado callLegProfile, faça uma solicitação POST em `/api/v1/callLegProfiles` com os parâmetros preferenciais e precisa do parâmetro **Ativation** definido como **true** para que você possa executar uma solicitação **GET** nessa **callLegProfile-ID** depois com este resultado, por exemplo:

```
needsActivation>true
```

```
needsActivation>
```

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

Anote a **callLegProfile-ID**, conforme marcado em negrito, pois ela deve ser aplicada no espaço na etapa 3 para o acesso de convidado (padrão).

Etapa 2. Crie um host **callLegProfile** (**precisaAtivation = false**).

Da mesma forma, crie o host **callLegProfile** para o comportamento do host da chamada. Os mesmos parâmetros mencionados anteriormente se aplicam, embora os parâmetros possam ser selecionados de acordo com sua própria preferência e requisitos. Aqui, o elemento principal é para definir o parâmetro **needsActivation** como **false** para que ele tenha a função de host.

Crie-o por meio de uma solicitação POST em `/api/v1/callLegProfiles` com os parâmetros preferenciais e precisa do parâmetro **Ativation** definido como **false** para que você possa executar uma solicitação **GET** nessa **callLegProfile-ID** depois com este resultado, por exemplo:

```
needsActivation>
```

```
false
```

```
needsActivation>
```

Anote a **callLegProfile-ID**, conforme marcado em negrito, pois ela deve ser aplicada no espaço **accessMethod** na etapa 4 para o acesso do host.

Etapa 3. Atribua o convidado **callLegProfile** para um espaço novo ou existente (sendo o **accessMethod** padrão).

Execute um comando **PUT** em um espaço existente (`/api/v1/coSpaces/<coSpace-ID>`) para adaptar o espaço ou um comando **POST** em `/api/v1/coSpaces` para criar um novo com o parâmetro de convidado **callLegProfile**, conforme criado na etapa 1 como o comportamento na chamada para esse espaço. Você também pode definir os parâmetros **URI**, [senha](#) e **call-ID** para esse espaço, ou como quiser, de acordo com a seção 6.2 do [guia da API](#).

Execute uma solicitação **GET** nesse espaço (`/api/v1/coSpaces/<coSpace-ID>`) para verificar se o convidado **callLegProfile** está associado a ele, bem como a **URI** e o valor **call-ID**. Um exemplo de saída com este exemplo criado **callLegProfile** na etapa 1 é este com um valor **URI** de **guest.space** e **call-ID** de 628821815 (sem conjunto de senhas):

```
uriCallIdcallLegProfile>
```

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

```
callLegProfile>
```

Anote a ID de espaço como marcada em negrito, pois ela deve ser usada para criar o **accessMethod** nesse espaço específico na etapa 4.

Etapa 4. Crie um novo **accessMethod** nesse espaço com um **URI** diferente (e **call-ID**) e atribua o host **callLegProfile** a ele.

Você deseja criar uma maneira diferente de acessar o espaço do que o acesso de convidado, que atualmente é o padrão. Isso é feito especificando um **accessMethod** no próprio espaço por um comando **POST** em `/api/v1/coSpaces/<coSpace-ID>/accessMethods` com aqui o **coSpace-ID** sendo o valor marcado em negrito na etapa 3 (**7cc797c9-c0a8-47cf-b519199 8dc5a01f1ade**) na qual o host **callLegProfile** da etapa 2 é aplicado, bem como os diferentes campos **URI** e **call-ID**.

Depois de uma solicitação **GET** nesse espaço **accessMethod** (`/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>`), você deve ser capaz de ver um tipo de saída semelhante a esta, onde você pode ver uma **URI** (**host.space**) e uma **call-ID** (**888**) diferente ao **accessMethod** padrão do espaço, bem como o host especialmente associado **callLegProfile** conforme configurado na etapa 2:

```
uricallIdpasscodecallLegProfile>
```

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

```
callLegProfile>
```

## Verificar

Agora você pode fazer a discagem para a mesma reunião:

- Como um convidado
  - discando para a URI guest.space (seguido do domínio como configurado em suas regras de correspondência de chamada)
  - inserindo call-ID de valores 628821815 via participação por IVR ou WebRTC (nenhuma senha)
- Como um host
  - discando para a URI host.space (seguido do domínio como configurado em suas regras de correspondência de chamada)
  - inserindo call-ID de valores 888 via participação por IVR ou WebRTC (nenhuma senha)

Quando há apenas convidados que entram espaço, eles são colocados em uma sala de lobby, esperando a entrada do host. Quando um host se junta, todos os convidados e hosts são colocados em conferência. Se não houver mais hosts unidos no espaço, mas ainda houver alguns convidados, eles retornarão à tela do lobby de acordo com a configuração de **desativar** no parâmetro **deactivationMode** no convidado **callLegProfile**, como mostrado na Etapa 1.

## 2) Configuração usando os mesmos URIs, mas PINs/senhas de host e convidado não vazios

Essa configuração é semelhante à do exemplo anterior e também está disponível antes do lançamento do CMS 2.1. Ela requer que o convidado e o host insira um PIN ou uma senha não em branco para que a diferenciação possa ser feita depois disso, durante a discagem para a mesma **URI**.

As etapas de configuração são bastante semelhantes ao exemplo de configuração anterior:

1. Crie um convidado **callLegProfile** (**needsActivation = true**)
2. Crie um host **callLegProfile** (**needsActivation = false**)
3. Atribua o convidado **callLegProfile** para um espaço novo ou existente especificando uma **senha de convidado (PIN)** (sendo o método de acesso padrão)
4. Crie um novo **accessMethod** naquele mesmo espaço com diferentes **URI** (diferente call-ID) e atribua o host **callLegProfile** a ele, incluindo uma **senha de host (PIN)**

Etapa 1. Crie um convidado **callLegProfile** (**precisaAtivation = verdadeiro**).

A mesma configuração do exemplo anterior 1 e até mesmo o mesmo convidado **callLegProfile** (**d4bfe12d-68cd-41c0-a671-48395ee170ab**) pode ser usada como demonstrado.

Etapa 2. Crie um host **callLegProfile** (**needsActivation = false**)

A mesma configuração que no exemplo 1 anterior e até mesmo o mesmo host **callLegProfile** (**7306d2c1-bc15-4dbf-ab4a-1cbdabd1912**) pode ser usada como demonstrado.

Etapa 3. Atribua o convidado **callLegProfile** para um espaço novo ou existente especificando uma senha de convidado (PIN) (sendo o **accessMethod** padrão).

Da mesma forma como antes, você pode executar uma operação PUT em um espaço existente (/api/v1/coSpaces/<cospace-ID>) ou uma operação POST para criar um novo espaço (/api/v1/coSpaces) com os parâmetros desejados para URI, senha e call-ID por exemplo, bem como o convidado Perfil (da etapa 1) que você atribuiu a ele conforme a seção 6.2 do [guia de API](#).

Se executar uma solicitação GET nesse espaço, você deverá ser capaz de ver um tipo de saída semelhante a esta onde você vê uma **URI** de guestpin.space, uma **call-ID** de 189, nosso convidado criado anteriormente **callLegProfile** e uma **senha** de 789:

```
uricallIdcallLegProfile>
```

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

```
callLegProfile><
```

```
passcode>789
```

```
passcode>
```

Anote a ID de espaço como marcada em negrito, pois ela deve ser usada para criar o **accessMethod** nesse espaço específico na etapa 4.

Etapa 4. Crie um novo **accessMethod** nesse espaço com o mesmo **URI** (diferente **call-ID**) e atribua o host **callLegProfile** a ele incluindo uma senha de host (PIN).

Nesse espaço você também criou um método de acesso diferente para os hosts (como um convidado **callLegProfile** é atribuído em um espaço como a opção de participação padrão), como no primeiro exemplo de configuração. Isso é feito usando um comando **POST** em /api/v1/coSpaces/<coSpace-ID>/accessMethods com o valor de coSpace-ID para nossos espaço sendo **22d9f4ca-8b88-4d11-bba9-e2a2f7428c46**, como destacado na etapa anterior. Nesse

comando POST, você pode fornecer os diferentes parâmetros, como URI (guestpin.space, o mesmo que o original), call-ID (889), host **callLegProfile**, como definido na etapa 2 e o host **passcode** ou **PIN** (1234 neste caso).

Se você executar uma solicitação GET nesse accessMethod, você deverá conseguir ver um tipo similar de saída mostrando a mesma URI do guestpin.space, uma **call-ID** de 889, a referência do host **callLegProfile** e o host **PIN** de 1234:

```
uricallId>passcode>1234
```

```
passcode><
```

```
callLegProfile>
```

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

```
callLegProfile>
```

## Verificar

Agora você pode fazer a discagem para a mesma reunião:

- Como um convidado
  - discando para o **URI** guestpin.space (seguido pelo domínio configurado em suas regras de correspondência de chamadas) e inserindo o **PIN** 789
  - inserindo o **call-ID** com o valor **189** via participação por **IVR** ou **WebRTC** com o **PIN** 789
- Como um host
  - discando para o **URI** guestpin.space (seguido pelo domínio configurado nas regras de correspondência de chamadas) e inserindo o **PIN** 1234
  - inserindo o **call-ID** com o valor **889** via participação por **IVR** ou **WebRTC** com o **PIN** 1234

Quando há apenas convidados que entram espaço, eles são colocados em uma sala de lobby, esperando a entrada do host. Quando um host se junta, todos os convidados e hosts são colocados em conferência. Se não houver mais hosts unidos no espaço, mas ainda houver alguns convidados, eles retornarão à tela do lobby de acordo com a configuração de **desativar** no parâmetro **deactivationMode** no convidado **callLegProfile**, como mostrado na Etapa 1.

## 3) Configuração usando os mesmos URIs com combinação de PIN de convidado vazio e PIN de host não vazio

Esta configuração só está disponível a partir da versão 2.1 do CMS, devido a alguns comandos

API recém-adicionados de `passcodeMode` e `passcodeTimeout` na seção `callProfile` . Isso permite um PIN em branco para a participação dos convidados (insira # ou tempo limite) enquanto o host tem um PIN para acessar o espaço e ativá-lo. O `callProfile` controla a experiência da chamada para chamadas SIP (incluindo Lync) e, portanto, não é aplicável para clientes de CMA (thick client e WebRTC).

As etapas de configuração são semelhantes às do exemplo 2, com o acréscimo do `callProfile`:

1. Crie um convidado `callLegProfile` (`needsActivation = true`)
2. Crie um host `callLegProfile` (`needsActivation = false`)
3. Crie um `callProfile` com a configuração desejada de `passcodeMode` e configuração de `passcodeTimeout`
4. Atribua o convidado `callLegProfile` e `callProfile` para um espaço novo ou existente especificando uma senha de convidado (PIN) (sendo o método de acesso padrão)
5. Crie um novo `accessMethod` naquele mesmo espaço com diferentes URI (diferente `call-ID`) e atribua o host `callLegProfile` a ele, incluindo uma senha de host (PIN)

Como as configurações são bastante idênticas aos exemplos de configuração 1 e 2, há referências a esses exemplos. Na verdade, para o teste, o mesmo espaço foi usado como no exemplo 2, mas adicionado com `callProfile` agora.

Etapa 1. Crie um convidado `callLegProfile` (`precisaAtivation = verdadeiro`).

A mesma configuração que no exemplo 1 anterior e até mesmo o mesmo convidado `callLegProfile` (`d4bfe12d-68cd-41c0-a671-48395ee170ab`) pode ser usada como demonstrado.

Etapa 2. Crie um host `callLegProfile` (`precisaAtivation = false`).

A mesma configuração que no exemplo 1 anterior e até mesmo o mesmo host `callLegProfile` (`7306d2c1-bc15-4dbf-ab4a-1cbdabd1912`) pode ser usada como demonstrado.

Etapa 3. Crie um `callProfile` com a configuração desejada `passcodeMode` e `passcodeTimeout`.

Você pode criar um `callProfile` que determina a experiência da chamada para chamadas SIP (incluindo Lync). Existem algumas configurações possíveis aqui, como permissão de gravação ou transmissão ou o limite máximo de participantes, por exemplo, mas o foco está nas novas adições de API do CMS 2.1 relacionadas ao tratamento de senha. Os outros parâmetros podem ser encontrados na seção 8.2 do [guia da API](#).

Dois parâmetros determinam o comportamento de senha neste exemplo, sendo:

- **`passcodeMode`**

- **obrigatório:** a IVR espera para sempre que um usuário digite o PIN ou # para um PIN vazio (para convidados)

- **tempo limite:** o IVR espera por uma quantidade de segundos `passcodeTimeout` para o participante digitar o PIN e se nenhuma entrada tiver sido feita nesse período, ele supõe que um PIN (#) vazio foi inserido

- **`passcodeTimeout`** : só precisa ser definido quando o `passcodeMode` é definido como tempo limite e controla a quantidade de tempo antes de interpretar a senha como sendo em branco

Para criar o `callProfile`, execute um comando **POST** em `/api/v1/callProfiles` (ou **PUT** em



**/api/v1/callProfiles/<callProfile-ID>** se você deseja modificar um existente) com os parâmetros desejados para **passcodeMode** e **passcodeTimeout**. Se você executar um comando **GET** nesse **callProfile** específico, você deve obter um tipo similar de resultado, por exemplo, onde você configurou o modo como tempo limite e um valor de tempo limite de 5 segundos:

```
passcodeMode>timeout
```

```
passcodeMode><
```

```
passcodeTimeout>5
```

```
passcodeTimeout>
```

Anote a **callProfile-ID**, conforme marcado em negrito, pois ela deve ser usada para atribuir ao espaço para ter esse comportamento na chamada na etapa 4.

Etapa 4. Atribua o convidado **callLegProfile** e **callProfile** da etapa 3 a um espaço novo ou existente especificando uma senha de convidado (PIN) (sendo o método de acesso padrão).

Da mesma forma como antes, você pode executar uma operação **PUT** em um espaço existente (**/api/v1/coSpaces/<cospace-ID>**) ou uma operação **POST** para criar um novo espaço (**/api/v1/coSpaces**) com os parâmetros desejados para a **URI** e **call-ID** por exemplo, como o convidado **callLegProfile** (da Etapa 1). A diferença de exemplos anteriores é o **callProfile** na etapa 3 e o fato de que nenhuma senha é atribuída a ele.

Se executar uma solicitação **GET** nesse espaço, você deverá ser capaz de ver um tipo de saída semelhante a este exemplo, onde você vê a **URI** de **guestpin.space**, uma **call-ID** de 189, o convidado criado anteriormente **callLegProfile** e a **callProfile** configurada na etapa 3:

```
uricallIdcallLegProfile>
```

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

```
callLegProfile><
```

```
callProfile>
```

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

```
callProfile>
```

Anote a ID de espaço como marcada em negrito, pois ela deve ser usada para criar o `accessMethod` nesse espaço específico na etapa 5.

Etapa 5. Crie um novo `accessMethod` nesse mesmo espaço com o mesmo **URI** (diferente **call-ID**) e atribua o host `callLegProfile` a ele incluindo uma senha de host (PIN).

Nesse espaço você também criou um método de acesso diferente para os hosts (como um convidado `callLegProfile` é atribuído em um espaço como a opção de participação padrão), como no primeiro exemplo de configuração. Isso é feito usando um comando **POST** em `/api/v1/coSpaces/<coSpace-ID>/accessMethods` onde o valor de `coSpace-ID` é substituído pelo valor do seu espaço **22d9f4ca-8b88-4d11-bba9-e2a2f7428c46**, como destacado na etapa anterior. Nesse comando **POST**, você pode fornecer os diferentes parâmetros, como **URI** (`guestpin.space`, o mesmo que o original), **call-ID** (889), host `callLegProfile`, conforme definido na Etapa 2 e a senha do host ou PIN (1234 neste caso).

Se você executar uma solicitação **GET** nesse `accessMethod`, você deverá conseguir ver um tipo similar de saída mostrando a mesma URI do `guestpin.space`, uma **call-ID** de 889, a referência do host `callLegProfile` e o host **PIN** de 1234:

```
uricallIdpasscode>1234
```

```
passcode><
```

```
callLegProfile>
```

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

```
callLegProfile>
```

## Verificar

Agora você pode fazer a discagem para a mesma reunião:

- Como um convidado
  - discando para o **URI** `guestpin.space` (seguido pelo domínio configurado nas regras de correspondência de chamadas) e inserindo **#** como **PIN** ou deixando-o expirar após 5 segundos
  - inserindo o **call-ID** com o valor **189** via participação por **IVR** ou **WebRTC**
- Como um host
  - discando para **URI** `guestpin.space` (seguido pelo domínio configurado em suas regras de correspondência de chamadas) e inserindo o **PIN** 1234
  - inserindo o **call-ID** com o valor 889 via participação por **IVR** ou **WebRTC** com o PIN

**4) Um usuário host é um membro do espaço e autorizado por meio do logon do WebRTC, os usuários convidados participam da reunião com callID. O mesmo URI e callID são usados por participantes convidados e hosts com PIN/senhas vazias ou não vazias para usuários convidados**

As próximas etapas precisam ser executadas para obter a diferenciação de acesso de convidado/host no mesmo espaço para membros e não membros do espaço:

1. Crie um convidado callLegProfile (**needsActivation = true**)
2. Crie um host callLegProfile (**needsActivation = false**)
3. Atribua o convidado callLegProfile para um espaço novo ou existente (sendo o método de acesso padrão)
4. Crie um novo **accessMethod** nesse mesmo espaço com o mesmo **URI** (e call-ID) e atribua o host **callLegProfile** a ele
5. Atribua sownerJID ao mesmo espaço. (se não atribuído)
6. Adicione esse ownerId como o usuário membro ao mesmo espaço e atribua host callLegProfile esse usuário membro

**Etapa 1.** Crie um convidado callLegProfile (needsActivation = true).

A mesma configuração do exemplo anterior 1 e o convidado callLegProfile (**bfe7d07f-c7cb-4e90-a46e-4811bbaf6978**) é usado neste exemplo.

Anote a **callLegProfile-ID**, conforme marcado em negrito, pois ela deve ser aplicada no espaço na etapa 3 para o acesso do convidado.

**Etapa 2.** Crie um host callLegProfile (needsActivation = false)

A mesma configuração do exemplo anterior 1 e o host callLegProfile (**0e76e943-6d90-43df-9f23-7f1985a74639**) é usado neste exemplo.

Anote a **callLegProfile-ID** conforme marcado em negrito, pois ela deve ser aplicada no espaço **accessMethod** na etapa 4 para o acesso do host e no membro do coSpace na etapa 6.

**Etapa 3.** Atribua o convidado **callLegProfile** para um espaço novo ou existente (sendo o accessMethod padrão).

Execute um comando PUT em um espaço existente (**/api/v1/coSpaces/<coSpace-ID>**) para adaptar o espaço ou um comando POST em **/api/v1/coSpaces** para criar um novo com o parâmetro convidado **callLegProfile** criado na etapa 1 como o comportamento no espaço para isso. Você também pode definir os parâmetros **URI** e **call-ID** para esse espaço, bem como sua vontade, conforme a seção 6.2 do [guia de API](#).

Execute uma solicitação GET nesse espaço (**/api/v1/coSpaces/<coSpace-ID>**) para verificar se o convidado **callLegProfile** está associado a ele, bem como o **URI** e o valor de ID de chamada. Um exemplo de saída com este exemplo criado callLegProfile na etapa 1 é este com um valor **URI global** e **call-ID** de 1234 (sem conjunto de senhas), nonMemberAccessset **true**:

```
<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>
```

Anote a ID de espaço como marcada em negrito, pois ela deve ser usada para criar o **accessMethod** nesse espaço específico na etapa 4.

**Etapa 4.** Crie um novo **accessMethod** nesse espaço com o mesmo **URI** (e **call-ID**) e atribua o **host callLegProfile** a ele.

Você deseja criar uma maneira diferente de acessar o espaço do que o acesso de convidado, que atualmente é o padrão. Isso é feito especificando um **accessMethod** no próprio espaço por um comando POST em **/api/v1/coSpaces/<coSpace-ID>/accessMethods** com aqui o **coSpace-ID** sendo o valor marcado em negrito na etapa 3 (**96d28acb-86c6-478d-b81a -a37ffb0adafc**) na qual o **host callLegProfile** da etapa 2 é aplicado, bem como o mesmo **URI** e **call-ID** campo. Você pode adicionar uma senha não vazia para os hosts que se conectam via **callID** (sem fazer logon como um usuário via WebRTC).

Depois de uma solicitação GET nesse espaço **accessMethod** (**/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>**), você deve ser capaz de ver um tipo de saída semelhante a esta, onde você pode ver o **mesmo URI** (global) e **call-ID** (1234) assim como **callLegProfile** do **host** especialmente associado conforme configurado na etapa 2 e **senha do host**(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>kff0lzTTE0feL4fsdf43w_B </secret>
</accessMethod>
```

**Etapa 5.** Atribua **ownerJid** ao espaço. (se não atribuído). Adicione **ownerJID** ao espaço especificando **ownerJid** ([user1@evacanoalone.net](mailto:user1@evacanoalone.net)) no espaço por um comando PUT em **/api/v1/coSpaces/<coSpace-ID>**

Depois de uma solicitação GET nesse espaço, você deve conseguir ver que **ownerId** and **ownerJid** have foi atribuído ao espaço:

```
<?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>0w4O2zTTF0WdL4ymF8D0_A</secret>
```

```
<numAccessMethods>1</numAccessMethods>
<defaultLayout>allEqual</defaultLayout>
</coSpace>
```

Anote o ownerId (1d942281-413e-4a2a-b776-91a674c3a5a9).

**Etapa 6.** Adicione a ID do proprietário (1d942281-413e-4a2a-b776-91a674c3a5a9) da etapa 5 como usuário membro ao espaço e atribua hostCallLegProfile a esse usuário membro. Isso é feito especificando-se userJid e hostCallLegProfile no próprio espaço (especificando coSpaceID) por um comando POST (/api/v1/coSpaces/<coSpaceID>/coSpaceUsers). Outros parâmetros no coSpaceUsers podem ser encontrados na seção 6.4.2 do [guia API](#), sob a qual os mostrados também podem ser relevantes nesta configuração:

```
<canDestroy>verdadeiro</canDestroy>

<canAddRemoveMember>verdadeiro</canAddRemoveMember>

<canChangeName>verdadeiro</canChangeName>

<canChangeUri>falso</canChangeUri>

<canChangeCallId>false</canChangeCallId>

<canChangePasscode>true</canChangePasscode>

<canPostMessage>true</canPostMessage>

<canDeleteAllMessages>falso</canDeleteAllMessages>

<canRemoveSelf>falso</canRemoveSelf>
```

Verifique se o usuário membro foi adicionado ao espaço por um comando GET (/api/v1/coSpaces/<coSpaceID>/coSpaceUsers?)

```
<?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>
```

Anote a ID de usuário (se diferente da etapa 5 do formulário ownerId). Verifique se o maior callLegProfile foi atribuído a coSpaceUser por uma solicitação GET especificando coSpaceID e userID (/api/v1/coSpaces/<coSpaceID>/coSpaceUsers/<userID>)

```
<?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>
```

## Verificar

Agora você pode fazer a discagem para a mesma reunião:

- Como um convidado

- discando para URI (seguido pelo domínio configurado nas regras de correspondência de chamadas)

- inserindo call-ID de valores 1234 via participação por IVR ou WebRTC (nenhuma senha)

- Como um host

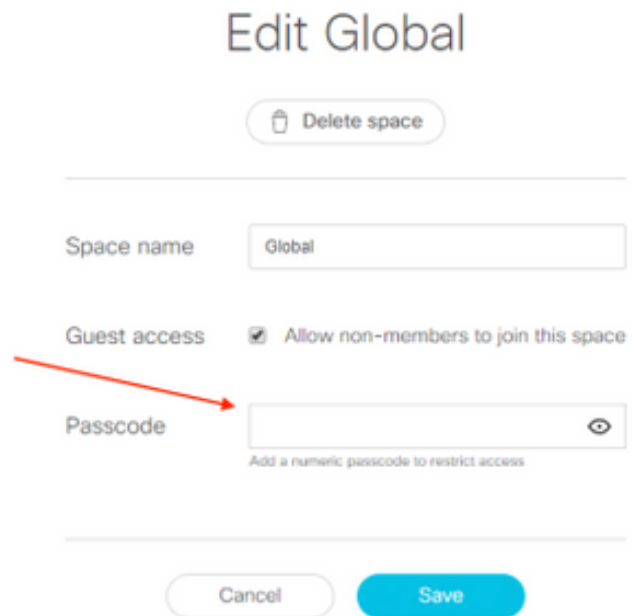
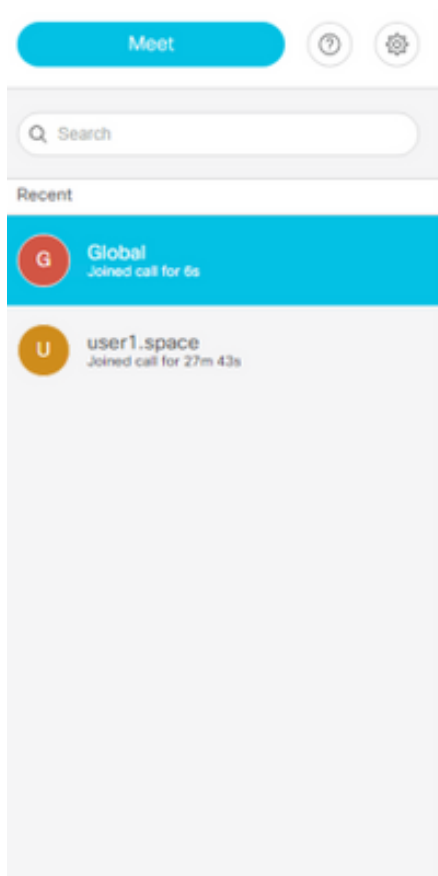
Fazendo login como um usuário (um membro do espaço com callLegProfile atribuído como "host", com user1@evacanoalone.net neste cenário) via webRTC e ingresse no espaço (URI "global").

- discando para URI "global" (seguido pelo domínio configurado nas regras de correspondência de chamadas) e pela senha 12345.

- inserindo o valor de call-ID 1234 por meio de participação IVR ou WebRTC (com uma senha de host 12345)

Quando há apenas convidados que entram espaço, eles são colocados em uma sala de lobby, esperando a entrada do host. Quando um host se junta, todos os convidados e hosts são colocados na conferência. Se não houver mais hosts unidos no espaço, mas ainda houver alguns convidados, eles retornarão à tela do lobby de acordo com a configuração de **desativar** no parâmetro **deactivationMode** no convidado **callLegProfile**, como mostrado na Etapa 1.

O host (proprietário/membro) pode definir (editar/remover) uma senha para convidados diretamente no aplicativo WebRTC ou desativar completamente um acesso de não membro (convidado) para o espaço:



## Troubleshoot

Esta seção disponibiliza informações para a solução de problemas de configuração.

O registro do CMS nos mostra brevemente quando você entra como convidado ou quando o primeiro host entra, mas é melhor verificar usando as solicitações GET do callProfile, bem como as definições de convidado e host callLegProfile e a alocação delas nos respectivos accessMethods (ou o método de acesso padrão) ou em nível superior (nível global ou nível de locatário).

Você pode seguir esta estrutura para obter todas as informações:

1. **GET em /api/v1/callProfiles** (se o estiver usando com o **passcodeMode**)  
>verifique em detalhes a callProfile-ID desejada usando **GET** em **/api/v1/callProfiles/<callProfile-ID>**
2. **GET em /api/v1/callLegProfiles**  
>verificar em detalhes as callLegProfile-IDs desejadas de convidado e host usando **GET** em **/api/v1/callProfiles/<callProfile-ID>**
3. **GET em /api/v1/coSpaces**  
>verifique em detalhes a ID de espaço desejada usando **GET** em **/api/v1/coSpaces/<coSpace-ID>**  
>verifique se o callProfile-ID (etapa 1) e o convidado callLegProfile (etapa 2) estão associados a este espaço  
[se não estiverem lá, verifique os elementos menos específicos como locatário (**/api/v1/espacos/<tenant-ID>**) ou nível global (**/api/v1/system/profile**)]

#### 4. GET em /api/v1/coSpaces/<coSpace-ID>/accessMethods

>verifique em detalhes o accessMethod desejado usando GET em /api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID> para verificar se o host callLegProfile está atribuído

No registro do CMS mostrado neste exemplo, você tem dois primeiros participantes convidados entrando (*ligue para 38* do 2000@steven.lab e *ligue para 39* do 1060@steven.lab), que atingem o espaço guestpin.space@acano.steven.lab e depois o host entra. Com o snippet, você pode ver que, para os convidados, ele informa sobre o que deve ser feito com ele (**para ser desativado**) e você pode ver esse comportamento para as alterações de chamadas quando o host (stejanss.movi@steven.lab) ingressa em um espaço (**interrupção da desativação**). Da mesma forma você pode rever o mesmo registro quando os convidados são transferidos novamente para o lobby, assim como a ausência de hosts em um espaço (**a ser desativado**).

```
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
```

## Informações Relacionadas

- [Suporte Técnico e Documentação - Cisco Systems](#)



- [documentação CMS](#)