

# Configurer et dépanner l'accès d'invité et d'hôte aux espaces CMS

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Configuration](#)

[1\) Configuration à l'aide d'URI différents](#)

[Vérification](#)

[2\) Configuration à l'aide des mêmes URI, mais des codes PIN/mots de passe d'invité et d'hôte non vides](#)

[Vérification](#)

[3\) Configuration à l'aide des mêmes URI avec combinaison du code PIN invité vide et du code PIN hôte non vide](#)

[Vérification](#)

[4\) Un utilisateur hôte est membre de l'espace et autorisé par le biais de la connexion WebRTC, les utilisateurs invités rejoignent la téléconférence avec callID. Le même URI et le même ID d'appel sont utilisés par les participants invités et hôtes avec un code PIN/mot de passe vide ou non vide pour les utilisateurs invités](#)

[Vérification](#)

[Dépannage](#)

[Informations connexes](#)

## Introduction

Ce document décrit comment configurer l'accès d'invité et d'hôte dans des espaces de votre serveur de réunion Cisco (CMS) à l'aide des commandes de l'API.

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Le serveur de réunion Cisco (CMS) avec des espaces définis et la capacité d'y passer des appels
- client API (comme Poster, Postman) ou
- [Guide de l'API CMS](#)

### Components Used

L'information contenue dans le présent document est fondée sur la version 2.1 de CMS

L'information contenue dans ce document a été créée à partir d'un appareil dans des conditions d'essai en laboratoire spécifiques. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

## Informations générales

Le document présente des types de scénarios :

- Différentes URI ou ID d'appels sont utilisées par les participants invités et hôtes
- La même URI est utilisée par les participants invités et hôtes où la différenciation est faite en fonction de la saisie du NIP ou du mot de passe (tous les deux non vides)
- La même URI est utilisée par les participants invités et hôtes où la différenciation est faite en fonction de la saisie du NIP ou du mot de passe (mélange vide et non vide)
- Un utilisateur hôte est membre de l'espace et autorisé par le biais de la connexion WebRTC, les utilisateurs invités rejoignent la téléconférence avec callID. Le même URI et le même ID d'appel sont utilisés par les participants invités et hôtes avec un code PIN/mot de passe vide ou non vide pour les utilisateurs invités

## Configuration

Il existe quatre possibilités de différenciation entre les participants invité et hôte dans CMS, décrites dans les 4 exemples suivants, et sont principalement basées sur différents **callLegProfiles** qui déterminent le comportement en appel des participants qui se connectent à l'espace.

Tout d'abord, la méthode à l'aide d'un **URI** différent (ou ID d'appel) pour les participants invités et hôtes est expliquée, et ensuite elle est ajoutée en utilisant différents codes de passe (ou délai d'attente) sur le même **URI**, pour faire la différenciation entre les participants invités et hôtes. La troisième méthode d'expiration du délai d'attente ou d'entrée de code confidentiel vide pour les utilisateurs invités a été introduite comme nouvelle fonctionnalité sur CMS 2.1, comme indiqué à la section 2.4 des [notes de version](#). La quatrième méthode explique comment configurer l'accès Invité et Hôte sur les espaces dont le propriétaire/les membres sont affectés et faire du membre de l'espace (propriétaire) l'hôte de l'espace.

### 1) Configuration à l'aide d'URI différents

Il s'agit de la configuration de base qui était disponible avant la version 2.1 de CMS et qui est la même que pour une ID d'appels différente. Les étapes suivantes doivent être effectuées pour obtenir la différenciation d'accès invité/hôte sur le même espace :

1. Créer un **callLegProfile invité (needsActivation = true)**
2. Créer un **callLegProfile hôte (needsActivation = false)**
3. Affecter le **callLegProfile invité à un espace nouveau ou existant (étant la méthode d'accès par défaut)**
4. Créer un nouveau **accessMethod sur cet espace même avec une autre URI (et ID d'appels) et lui affecter le callLegProfile hôte**

Étape 1. Créez un **callLegProfile invité (needsActivation = true)**.

Un **callLegProfile** détermine le comportement en appel et, par défaut, vous affectez le comportement en appel de l'invité à l'espace de sorte que vous puissiez plus tard avoir une méthode d'accès différente sur ce même espace, ainsi que pour que l'hôte puisse se joindre à.

**Note:** Vous pouvez également affecter cela au niveau du détenteur (`/api/v1/tenants/<tenant-ID>`) ou du système (`/api/v1/system/profiles`) pour, par exemple, appliquer cela à tous les espaces (ou par détenteur), cependant, ici, cela est affiché sur l'espace lui-même. Tenez compte du fait que l'allocation la plus spécifique du **callLegProfile** est prise en compte pour le comportement en appel.

Le paramètre **needsActivation** est le plus important ici pour le comportement invité/hôte puisque **s'il est réglé à true**, le participant sera dans l'impossibilité de recevoir ou de contribuer à l'audio et la vidéo jusqu'à ce que l'un ou plusieurs participants **intégraux/activateurs (hôte) se joignent**. D'autres paramètres sur le **callLegProfile** se trouvent à la section 8.4.3 du [guide API](#), sous laquelle les paramètres affichés peuvent être pertinents dans cette configuration également (selon vos besoins) :

- `presentationContributionAllowed`
- `rxAudioMute`
- `rxVideoMute`
- `deactivationMode` (`deactive` | `déconnecter` | `stayActivated`) et `deactivationModeTime` [action à effectuer lorsque le dernier activateur quitte l'appel]

Pour créer le **callLegProfile** invité, faites une requête POST sur `/api/v1/callLegProfiles` avec les paramètres et le paramètre d'activation définis sur `true` afin que vous puissiez effectuer une requête **GET** sur cet **callLegProfile-ID** par la suite de ce résultat :

```
needsActivation>true
```

```
needsActivation>
```

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

Notez que l'**callLegProfile-ID** est marqué en gras, car il doit être appliqué à l'espace de l'étape 3 pour l'accès invité (par défaut).

Étape 2. Créez un **callLegProfile** hôte (**fautActivation = false**).

D'une façon semblable, créez le **callLegProfile** hôte pour le comportement en appel de l'hôte. Les mêmes paramètres que ceux mentionnés précédemment s'appliquent, bien que les paramètres puissent être sélectionnés selon vos préférences et exigences. L'élément principal ici est de configurer le paramètre **needsActivation** à **false** pour lui donner le rôle de l'hôte.

Vous le créez par une requête POST sur `/api/v1/callLegProfiles` avec les paramètres et les paramètres d'activation préférés définis sur **false** afin que vous puissiez effectuer une requête **GET** sur cet **callLegProfile-ID** par la suite avec ce résultat par exemple :

```
needsActivation>
```

false

```
needsActivation>
```

Notez que l'**callLegProfile-ID** est marqué en gras, car il doit être appliqué sur l'espace **accessMethod** à l'étape 4 pour l'accès à l'hôte.

Étape 3. Affectez le **callLegProfile** invité à un espace existant ou nouveau (**accessMethod** par défaut).

Effectuez une commande **PUT** sur un espace existant (`/api/v1/coSpaces/<coSpace-ID>`) pour adapter l'espace ou une commande **POST** sur `/api/v1/coSpaces` pour en créer un nouveau avec le paramètre **callLegProfile** invité comme créé à l'étape 1 en tant que comportement en appel pour cet espace. Vous pouvez également configurer les paramètres de l'URI, [du mot de passe et de l'ID d'appels](#) pour cet espace à votre goût selon la section 6.2 du [guide de l'API](#).

Effectuez une requête **GET** sur cet espace (`/api/v1/coSpaces/<coSpace-ID>`) pour vérifier que le **callLegProfile** invité lui est associé, ainsi que l'URI et la valeur de l'ID d'appels. Un exemple de sortie avec cet exemple créé **callLegProfile** invité à l'étape 1 est celui-ci avec une valeur **URI** **guest.space** et **call-ID** de 628821815 (aucun jeu de codes secret) :

```
uricallIdcallLegProfile>
```

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

```
callLegProfile>
```

Notez l'ID d'espace marqué en gras, car il doit être utilisé pour créer la **accessMethod** sur cet espace particulier à l'étape 4.

Étape 4. Créez une nouvelle **accessMethod** sur cet espace avec un **URI** différent (et **call-ID**) et affectez-lui le **callLegProfile** hôte.

Vous voulez créer une autre façon d'accéder à l'espace que l'accès invité qui est actuellement le mode par défaut. Pour ce faire, spécifiez une **accessMethod** sur l'espace lui-même par une commande **POST** sur `/api/v1/coSpaces/<coSpace-ID>/accessMethods` avec ici le **coSpace-ID** étant la valeur marquée en gras à l'étape 3 (`7cc797c9-c0a8-47cf-b519-8dddj c5a01f1ade`) sur laquelle le **callLegProfile** hôte de l'étape 2 est appliqué, ainsi que les différents champs **URI** et **call-ID**.

Après une requête **GET** sur cet espace **accessMethod** (`/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>`), vous devez voir un type de sortie similaire à celui-ci, où vous pouvez voir un **URI** différent (**host.space**) et **call-ID** (888) par opposition à par défaut

**accessMethod** de l'espace ainsi que le **callLegProfile** hôte associé comme configuré à l'étape 2 :

```
uricallIdpasscodecallLegProfile>
```

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

```
callLegProfile>
```

## Vérification

Maintenant, vous pouvez appeler à la même réunion :

- En tant qu'invité
  - en composant à l'URI `guest.space` (suivi du domaine tel que configuré dans vos règles de correspondance d'appel)
  - en entrant la valeur d'ID d'appels 628821815 en rejoignant par IVR ou WebRTC (aucun mot de passe)
- En tant qu'un hôte
  - en composant à l'URI `host.space` (suivi du domaine tel que configuré dans vos règles de correspondance d'appel)
  - en entrant la valeur d'ID d'appels 888 en rejoignant par IVR ou WebRTC (aucun mot de passe)

Lorsqu'il n'y a que des invités qui se joignent à l'espace, ils sont tous placés dans une salle d'attente en attendant que l'hôte s'y joigne. Une fois qu'un hôte se joint, tous les invités et les hôtes sont mis en conférence. S'il n'y a plus d'hôtes joints sur l'espace mais que certains invités sont toujours présents, ils retournent à l'écran du hall d'accueil conformément à la configuration de la **désactivation** sur le paramètre **deactivationMode** sur le **callLegProfile invité** comme indiqué à l'étape 1.

## 2) Configuration à l'aide des mêmes URI, mais des codes PIN/mots de passe d'invité et d'hôte non vides

Cette configuration est similaire à celle de l'exemple précédent, et est également disponible avant la version 2.1 de CMS. Elle nécessite que l'invité et l'hôte entrent un NIP ou un mot de passe non vide afin que la différenciation puisse se faire à partir de cela, comme ils composent à la même **URI**.

Les étapes de configuration sont assez similaires à l'exemple de configuration précédent :

1. Créer un **callLegProfile invité** (**needsActivation = true**)
2. Créer un **callLegProfile hôte** (**needsActivation = false**)
3. Affecter le **callLegProfile invité** à un espace nouveau ou existant en spécifiant un mot de

**pas**se (NIP) d'invité (étant la méthode d'accès par défaut)

4. Créer un nouveau **accessMethod** sur ce même espace avec la même URI (différente ID d'appels) et affecter le **callLegProfile** hôte à celui-ci en incluant un mot de passe de l'hôte (NIP)

Étape 1. Créez un **callLegProfile** invité (**needsActivation = true**).

La même configuration que dans l'exemple 1 précédent et même le même **callLegProfile** invité (**d4bfe12d-68cd-41c0-a671-48395ee170ab**) peut être utilisée comme démontré.

Étape 2. Créer un **callLegProfile** hôte (**needsActivation = false**)

La même configuration que dans l'exemple 1 précédent et même le même **callLegProfile** hôte (**7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912**) peut être utilisée comme démontré.

Étape 3. Affectez le **callLegProfile** invité à un espace existant ou nouveau en spécifiant un code PIN (guest passcode) (en tant que **accessMethod** par défaut).

De la même manière que précédemment, vous pouvez effectuer une opération PUT sur un espace existant (`/api/v1/coSpaces/<cospace-ID>`) ou une opération POST pour créer un nouvel espace (`/api/v1/coSpaces`) avec les paramètres souhaités pour l'URI, le code secret et l'ID d'appel par exemple, ainsi que le profil d'appel invité **callLegProfile** (à partir de l'étape 1) que vous lui avez attribuée conformément à la section 6.2 du [guide de l'API](#).

Si vous effectuez une requête GET sur cet espace, vous devez être en mesure de voir un type de sortie similaire à celui-ci où vous voyez un **URI** de `guestpin.space`, un **call-ID** de 189, notre **callLegProfile** et un **passcode** de 789 :

```
uricallIdcallLegProfile>
```

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

```
callLegProfile<<
```

```
passcode>789
```

```
passcode>
```

Notez l'ID d'espace marqué en gras, car il doit être utilisé pour créer la **accessMethod** sur cet espace particulier à l'étape 4.

Étape 4. Créez une nouvelle **accessMethod** sur cet espace avec la même URI (ID d'appel différent) et affectez le **callLegProfile** hôte à cet espace, y compris un code confidentiel (PIN) hôte.

Dans cet espace vous créez également une autre méthode d'accès pour les hôtes (comme le `callLegProfile` invité est affecté sur l'espace lui-même en tant que l'option par défaut pour se joindre), tout comme dans le premier exemple de configuration. Cela s'effectue en utilisant une commande **POST** sur `/api/v1/coSpaces/<coSpace-ID>/accessMethods` avec la valeur `coSpace-ID` pour notre espace étant `22d9f4ca-8b88-4d11-bba9-e2a2f7428c46` comme indiqué à l'étape précédente. Dans cette commande POST, vous pouvez fournir les différents paramètres tels que l'URI (`guestpin.space`, identique à celui d'origine), l'ID d'appel (889), le `callLegProfile` d'hôte défini à l'étape 2 et le **code secret** ou **PIN** (122).

Si vous effectuez une requête GET sur cette `accessMethod`, vous devez voir un type de sortie similaire montrant le même URI de `guestpin.space`, un **call-ID** de 889, la **référence callLegProfile** de l'hôte et le **PIN** de l'hôte de 1234 :

```
uricallId>passcode>1234
```

```
passcode><
```

```
callLegProfile>
```

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

```
callLegProfile>
```

## Vérification

Maintenant, vous pouvez appeler à la même réunion :

- En tant qu'invité
  - en composant l'**URI** `guestpin.space` (suivi du domaine tel que configuré sur vos règles de correspondance d'appels) et en saisissant le **code PIN** 789
  - en entrant la valeur d'**ID d'appels** 189 en rejoignant par **IVR** ou **WebRTC** avec le **NIP** 789
- En tant qu'un hôte
  - en composant l'**URI** `guestpin.space` (suivi du domaine tel que configuré sur vos règles de correspondance d'appels) et en saisissant le **code PIN** 1234
  - en entrant la valeur d'**ID d'appels** 889 en rejoignant par **IVR** ou **WebRTC** avec le **NIP** 1234

Lorsqu'il n'y a que des invités qui se joignent à l'espace, ils sont tous placés dans une salle d'attente en attendant que l'hôte s'y joigne. Une fois qu'un hôte se joint, tous les invités et les hôtes sont mis en conférence. S'il n'y a plus d'hôtes joints sur l'espace mais que certains invités sont toujours présents, ils retournent à l'écran du hall d'accueil conformément à la configuration de la **désactivation** sur le paramètre `deactivationMode` sur le `callLegProfile` invité comme indiqué à l'étape 1.

### 3) Configuration à l'aide des mêmes URI avec combinaison du code PIN invité vide et du code PIN hôte non vide

Cette configuration est uniquement disponible à partir de la version 2.1 ou ultérieure de CMS en raison de certaines commandes de l'API nouvellement ajoutées, `passcodeMode` et `passcodeTimeout` dans la section `callProfile`. Cela permet un NIP vide pour que les invités se joignent (soit en entrant # ou par délai) pendant que l'hôte a un NIP pour accéder à l'espace et l'activer. Le `callProfile` contrôle l'expérience en appel pour les appels SIP (y compris Lync) et est donc non applicable aux clients CMA (à la fois le client lourd et WebRTC).

Les étapes de configuration sont semblables à celles de l'exemple 2, avec l'ajout du `callProfile` :

1. Créer un `callLegProfile` invité (`needsActivation = true`)
2. Créer un `callLegProfile` hôte (`needsActivation = false`)
3. Créer un `callProfile` avec la configuration `passcodeMode` et `passcodeTimeout` souhaitée
4. Affecter le `callLegProfile` et le `callProfile` invités de l'étape 3 à un espace nouveau ou existant en spécifiant un mot de passe (NIP) d'invité (étant la méthode d'accès par défaut)
5. Créer un nouveau `accessMethod` sur ce même espace avec la même URI (différente ID d'appels) et affecter le `callLegProfile` hôte à celui-ci en incluant un mot de passe de l'hôte (NIP)

Comme les configurations sont assez identiques aux exemples de configuration 1 et 2, il y a des références à ces configurations. En fait, pour le test, le même espace a été utilisé que dans l'exemple 2, mais ajouté avec `callProfile` maintenant.

Étape 1. Créez un `callLegProfile` invité (`needsActivation = true`).

La même configuration que dans l'exemple 1 précédent et même le même `callLegProfile` invité (`d4bfe12d-68cd-41c0-a671-48395ee170ab`) peut être utilisée comme démontré.

Étape 2. Créez un `callLegProfile` hôte (`needsActivation = false`).

La même configuration que dans l'exemple 1 précédent et même le même `callLegProfile` hôte (`7306d2c1-bc15-4dbf-ab4a-1cbdaabd1912`) peut être utilisée comme démontré.

Étape 3. Créez un `callProfile` avec la configuration `passcodeMode` et `passcodeTimeout` souhaitée.

Vous pouvez créer un `callProfile` qui détermine l'expérience en appel pour les appels SIP (y compris Lync). Quelques configurations sont possibles ici, comme permettre l'enregistrement ou la diffusion ou ajuster la limite maximale de participants par exemple, mais on se concentre ici sur les nouveaux ajouts de l'API de CMS 2.1 relatifs à la manipulation du mot de passe. Les autres paramètres peuvent être consultés dans la section 8.2 du [guide de l'API](#).

Deux paramètres déterminent le comportement du mot de passe ici, qui sont :

- ***passcodeMode***

- **required** : l'IVR attend pour toujours qu'un utilisateur entre le code PIN ou le numéro # pour un code PIN vide (pour les invités)

- **timeout** : l'IVR attend un délai *passcodeTimeout* de secondes pour que le participant entre le code PIN et si aucune entrée n'a été effectuée dans ce délai, il suppose qu'un code PIN vide (#) a été saisi.



- ***passcodeTimeout*** : ne doit être défini que lorsque **passcodeMode** est réglé à « timeout » (délai d'attente) et contrôle le temps requis avant d'interpréter le mot de passe comme étant vide

Afin de créer le **callProfile**, exécutez une commande **POST** sur **/api/v1/callProfiles** (ou **PUT** sur **/api/v1/callProfiles/<callProfile-ID>** si vous souhaitez en modifier un existant) avec les paramètres de votre choix pour **passcodeMode** et **passcodeTimeout**. Si vous effectuez une commande **GET** sur ce **callProfile** en particulier, vous devez obtenir un résultat semblable à l'exemple où vous avez configuré le mode à « timeout » (délai d'attente) avec une valeur de délai d'attente de 5 secondes :

```
passcodeMode>timeout
```

```
passcodeMode><
```

```
passcodeTimeout>5
```

```
passcodeTimeout>
```

Notez le **callProfile-ID** comme indiqué en gras, car il doit être utilisé pour attribuer à l'espace pour que ce comportement d'appel se produise à l'étape 4.

Étape 4. Affectez le **callLegProfile** invité et le **callProfile** de l'étape 3 à un espace existant ou nouveau spécifiant un code PIN (guest passcode) (étant la méthode d'accès par défaut).

D'une façon semblable à celle vue plus tôt, vous pouvez soit effectuer une opération **PUT** sur un **espace existant (/api/v1/coSpaces/<cospace-ID>)** ou une opération **POST** pour créer un **nouvel espace (/api/v1/coSpaces)** avec les paramètres de votre choix pour l'URI et l'ID d'appels par exemple ainsi que le **callLegProfile** invité (de l'étape 1). La différence avec les exemples précédents est le **callProfile** de l'étape 3 et le fait qu'aucun mot de passe ne lui est attribué.

Si vous effectuez une requête **GET** sur cet espace, vous devez être en mesure de voir un type de sortie similaire à cet exemple, où vous voyez l'URI de **guestpin.space**, un **call-ID** de 189, le **callLegProfile** et le **callProfile** précédemment créés, comme configuré à l'étape 3 :

```
uricallIdcallLegProfile>
```

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

```
callLegProfile><
```

```
callProfile>
```

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

```
callProfile>
```

Notez l'ID d'espace marqué en gras, car il doit être utilisé pour créer `accessMethod` sur cet espace particulier à l'étape 5.

Étape 5. Créez une nouvelle `accessMethod` sur ce même espace avec le même **URI (ID d'appel différent)** et affectez le `callLegProfile` hôte à celui-ci, y compris un code confidentiel (PIN) hôte.

Dans cet espace vous créez également une autre méthode d'accès pour les hôtes (comme le `callLegProfile invité est affecté sur l'espace lui-même en tant que l'option par défaut pour se joindre`), tout comme dans le premier exemple de configuration. Cela s'effectue en utilisant une commande **POST** sur `/api/v1/coSpaces/<coSpace-ID>/accessMethods` où la valeur `coSpace-ID` est remplacée par la valeur de votre espace étant `22d9f4ca-8b88-4d11-bba9-e2a2f7428c46` comme indiqué à l'étape précédente pour ce cas. Sur cette commande **POST**, vous pouvez fournir les différents paramètres comme l'**URI** (`guestpin.space`, identique à celui d'origine), **call-ID** (889), **host callLegProfile** tel que défini à l'étape 2 et le code secret ou PIN de l'hôte (1234 dans ce cas).

Si vous effectuez une requête **GET** sur cette `accessMethod`, vous devez voir un type de sortie similaire montrant le même URI de `guestpin.space`, un **call-ID** de 889, la **référence callLegProfile** de l'hôte et le **PIN** de l'hôte de 1234 :

```
uricallIdpasscode>1234
```

```
passcode><
```

```
callLegProfile>
```

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

```
callLegProfile>
```

## Vérification

Maintenant, vous pouvez appeler à la même réunion :

- En tant qu'invité
  - en composant l'**URI** `guestpin.space` (suivi du domaine tel que configuré sur vos règles de correspondance d'appels) et en entrant **#** comme **PIN** ou en laissant le délai d'attente après 5 secondes

- en entrant la valeur d'**ID d'appels 189** en rejoignant par IVR ou WebRTC

- En tant qu'un hôte

- en composant l'**URI** `guestpin.space` (suivi du domaine configuré sur vos règles de correspondance d'appels) et en entrant le **code confidentiel** 1234

- en entrant la valeur d'**ID d'appels 889** en rejoignant par IVR ou WebRTC avec le NIP 1234

#### 4) Un utilisateur hôte est membre de l'espace et autorisé par le biais de la connexion WebRTC, les utilisateurs invités rejoignent la téléconférence avec callID. Le même URI et le même ID d'appel sont utilisés par les participants invités et hôtes avec un code PIN/mot de passe vide ou non vide pour les utilisateurs invités

Les étapes suivantes doivent être effectuées pour obtenir la différenciation d'accès invité/hôte sur le même espace pour les membres et les non-membres de l'espace :

1. Créer un `callLegProfile` invité (**needsActivation = true**)
2. Créer un `callLegProfile` hôte (**needsActivation = false**)
3. Affecter le `callLegProfile` invité à un espace nouveau ou existant (étant la méthode d'accès par défaut)
4. Créez une nouvelle **accessMethod** sur ce même espace avec le même **URI** (et ID d'appel) et affectez-lui le **callLegProfile** hôte
5. Affectez le `sownerJID` de l'utilisateur au même espace. (si non attribué)
6. Ajoutez cet ID propriétaire en tant qu'utilisateur membre au même espace et affectez `hostcallLegProfile` à cet utilisateur membre

**Étape 1.** Créez un `callLegProfile` invité (**NeedActivation = true**).

La même configuration que dans l'exemple 1 précédent et le `callLegProfile` invité (**bfe7d07f-c7cb-4e90-a46e-4811bbaf6978**) est utilisée dans cet exemple.

Notez que l'**callLegProfile-ID** est marqué en gras, car il doit être appliqué à l'espace de l'étape 3 pour l'accès invité.

**Étape 2.** Créez un `callLegProfile` hôte (**needsActivation = false**)

La même configuration que dans l'exemple 1 précédent et que le `callLegProfile` hôte (**0e76e943-6d90-43df-9f23-7f1985a74639**) est utilisée dans cet exemple.

Notez que l'**callLegProfile-ID** est marqué en gras, car il doit être appliqué sur l'espace **accessMethod** à l'étape 4 pour l'accès hôte et sur le membre `coSpace` à l'étape 6.

**Étape 3.** Affectez le **callLegProfile** invité à un espace existant ou nouveau (en tant que `accessMethod` par défaut).

Exécutez une commande PUT sur un espace existant (`/api/v1/coSpaces/<coSpace-ID>`) pour adapter l'espace ou une commande POST sur `/api/v1/coSpaces` pour en créer un nouveau avec le paramètre `callLegProfile` invité créé à l'étape 1 comme comportement d'appel pour cet espace. Vous pouvez également définir les paramètres **URI** et **ID d'appel** pour cet espace ainsi que votre désir conformément à la section 6.2 du [guide API](#).

Effectuez une requête GET sur cet espace (`/api/v1/coSpaces/<coSpace-ID>`) pour vérifier que le **callLegProfile** invité est associé à cet espace, ainsi que la **valeur URI** et **call-ID**. Un exemple de sortie avec cet exemple créé callLegProfile invité à l'étape 1 est celui-ci avec une valeur URI **globale** et **call-ID** de 1234 (aucun jeu de codes), nonMemberAccessset **sur true** :

```
<?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>0w402zTTF0WdL4ymF8D0_A</secret>
  <defaultLayout>allEqual</defaultLayout>
</coSpace>
```

Notez l'ID d'espace marqué en gras, car il doit être utilisé pour créer la **accessMethod** sur cet espace particulier à l'étape 4.

**Étape 4.** Créez une nouvelle **accessMethod** sur cet espace avec le même URI (et ID d'appel) et affectez-lui le **callLegProfile** hôte.

Vous voulez créer une autre façon d'accéder à l'espace que l'accès invité qui est actuellement le mode par défaut. Pour ce faire, spécifiez une **accessMethod** sur l'espace lui-même par une commande POST sur `/api/v1/coSpaces/<coSpace-ID>/accessMethods` avec ici **coSpace-ID** étant la valeur marquée en gras à l'étape 3 (**96d28acb-86c6-478d-b81a**). **a37ffb0adafc**) sur lequel le **callLegProfile** de l'étape 2 est appliqué, ainsi que le même URI et call-ID. Vous pouvez ajouter un code secret non vide pour les hôtes qui se connectent via callID (sans être connectés en tant qu'utilisateur via webRTC).

Après une requête GET sur cet espace **accessMethod** (`/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>`), vous devez être en mesure de voir un type de sortie similaire à celui-ci, où vous pouvez voir le **même URI** (global) et l'**ID d'appel** (1234) ainsi que spécialement associé **host callLegProfile** tel que configuré à l'étape 2 et **host passcode**(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>
```

**Étape 5.** Affectez **Jidto** au propriétaire de l'utilisateur. (si non attribué). Ajoutez le **ownerJID** à l'espace en **spécifiant ingownerJid** ([user1@evacanoalone.net](mailto:user1@evacanoalone.net)) sur l'espace par une commande PUT sur `/api/v1/coSpaces/<coSpace-ID>`

Après une requête GET sur cet espace, vous devez être en mesure de voir que **ownerJid** and **ownerJidare** a été affecté à cet espace :

```
<?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>

```

Notez l'ID du propriétaire (1d942281-413e-4a2a-b776-91a674c3a5a9).

**Étape 6.** Ajoutez cet ID de propriétaire (1d942281-413e-4a2a-b776-91a674c3a5a9) à partir de l'étape 5 en tant qu'utilisateur membre à l'espace et affectez `hostcallLegProfile` à cet utilisateur membre. Pour ce faire, spécifiez `userJid` et `hostcallLegProfile` sur l'espace lui-même (spécifiez `coSpaceID`) par une commande POST (`/api/v1/coSpaces/<coSpaceID>/coSpaceUsers`). D'autres paramètres sur `coSpaceUsers` peuvent être se trouve à la section 6.4.2 du [guide de l'API](#), sous laquelle les illustrés peuvent également être pertinents dans cette configuration :

```

<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>

```

Vérifier que l'utilisateur membre a été ajouté à l'espace par aGET command (`/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>

```

Notez l'ID utilisateur (si différent de l'étape 5 du formulaire ID propriétaire du formulaire). Vérifiez que `callLegProfile` a été attribué à `coSpaceUser` par une requête GET **spécifiant** `coSpaceID` et `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>
```

## Vérification

Maintenant, vous pouvez appeler à la même réunion :

- En tant qu'invité

- en composant l'URI (suivi du domaine configuré sur vos règles de correspondance d'appels)
- en entrant la valeur d'ID d'appels 1234 en rejoignant par IVR ou WebRTC (aucun mot de passe)

- En tant qu'un hôte

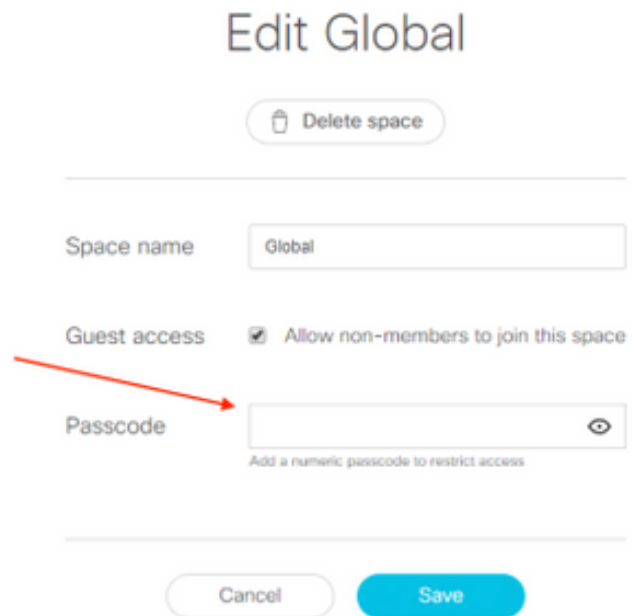
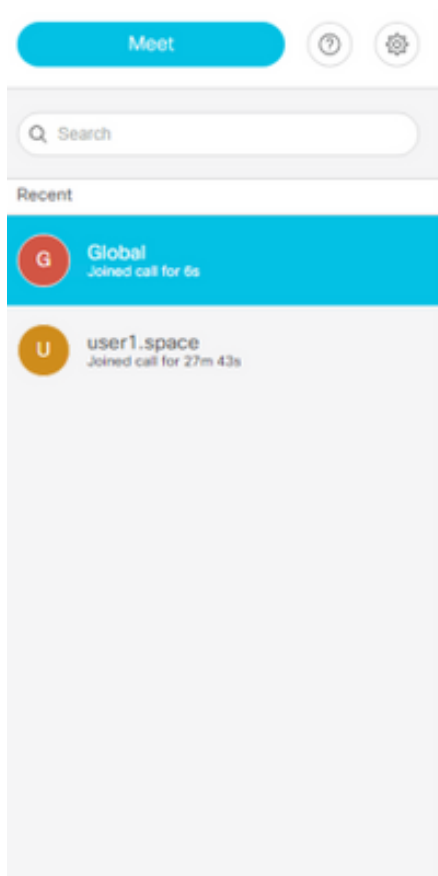
En se connectant en tant qu'utilisateur (membre de l'espace avec "hôte" callLegProfile assigné, avec user1@evacanoalone.net dans ce scénario) via webRTC et rejoindre l'espace ("URI" global).

- en composant "URI" global (suivi du domaine tel que configuré sur vos règles de correspondance d'appels) et le code secret 12345.

- en entrant la valeur call-ID 1234 via une jointure IVR ou WebRTC (avec un code secret d'hôte 12345)

Lorsqu'il n'y a que des invités qui se joignent à l'espace, ils sont tous placés dans une salle d'attente en attendant que l'hôte s'y joigne. Une fois qu'un hôte se joint, tous les invités et les hôtes sont placés dans la conférence. S'il n'y a plus d'hôtes joints sur l'espace mais que certains invités sont toujours présents, ils retournent à l'écran du hall d'accueil conformément à la configuration de la **désactivation** sur le paramètre **deactivationMode** sur le **callLegProfile invité** comme indiqué à l'étape 1.

L'hôte (propriétaire/membre) peut définir (modifier/supprimer) un mot de passe pour les invités directement dans l'application webRTC ou désactiver complètement un accès non membre (invité) pour l'espace :



## Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

La journalisation de CMS nous l'affiche bel et bien brièvement lorsque vous vous joignez en tant qu'invité ou lorsque le premier hôte se joint, mais il est préférable de vérifier à l'aide de requêtes GET le callProfile ainsi que les définitions des callLegProfile hôte et invité et leur répartition sur leurs accessMethod respectifs (ou la méthode d'accès par défaut) ou au niveau supérieur (niveau global ou du détenteur).

Vous pouvez suivre cette structure pour obtenir toutes les informations :

1. **GET** sur `/api/v1/callProfiles` (si vous l'utilisez avec `passcodeMode`)  
>vérifier en détail l'ID de profil d'appel souhaité à l'aide de **GET** on `/api/v1/callProfiles/<callProfile-ID>`
2. **GET** sur `/api/v1/callLegProfiles`  
>vérifier en détail les ID callLegProfile souhaités de l'invité et de l'hôte à l'aide de **GET** on `/api/v1/callProfiles/<callProfile-ID>`
3. **GET** sur `/api/v1/coSpaces`  
>vérifiez en détail l'ID d'espace souhaité à l'aide de **GET** on `/api/v1/coSpaces/<coSpace-ID>`  
>vérifier si l'ID callProfile-ID souhaité (étape 1) et le callLegProfile invité (étape 2) sont associés à cet espace  
[s'ils ne sont pas là, vérifiez les éléments moins spécifiques comme au niveau du détenteur (`/api/v1/tenants/<tenant-ID>`) ou global (`/api/v1/system/profiles`)]
4. **GET** sur `/api/v1/coSpaces/<coSpace-ID>/accessMethods`

>vérifiez en détail l'accessMethod souhaité à l'aide de **GET** on **/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>** pour vérifier si le **callLegProfile** hôte est affecté

Dans la journalisation CMS illustrée dans cet exemple, deux premiers participants invités arrivent (*appelez 38* à partir de 2000@steven.lab et *appelez 39* à partir de 1060@steven.lab) qui arrivent en dehors de l'espace `guestpin.space@acano.steven.lab`, puis l'hôte se joint. Vous pouvez voir dans l'extrait que, pour les invités, il nous indique ce qui doit être fait (**être désactivé**) et vous pouvez voir ce comportement changer pour ces appels lorsque l'hôte (`stejanss.movi@steven.lab`) se joint à l'espace (**cesser d'être désactivé**). De même, vous pouvez revoir la même journalisation lorsque les invités retournent à la salle d'attente dès qu'il n'y a plus d'hôte sur l'espace (**être désactivé**).

```
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
```

## Informations connexes

- [Support et documentation techniques - Cisco Systems](#)
- [Documentation CMS](#)