

Configurazione e risoluzione dei problemi di accesso guest e host in CMS Spaces

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[1\) Configurazione con URI diversi](#)

[Verifica](#)

[2\) Configurazione con gli stessi URI ma PIN/passcode host e non vuoti](#)

[Verifica](#)

[3\) Configurazione utilizzando gli stessi URI con combinazione di PIN guest vuoto e PIN host non vuoto](#)

[Verifica](#)

[4\) Un utente ospite è membro dello spazio e autorizzato tramite accesso webRTC, gli utenti ospiti si uniscono al meeting con callID. Lo stesso URI e lo stesso callID vengono utilizzati dai partecipanti guest e host con PIN/passcode vuoti o non vuoti per gli utenti guest](#)

[Verifica](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

Introduzione

In questo documento viene descritto come configurare l'accesso guest e host negli spazi del Cisco Meeting Server (CMS) utilizzando i comandi API.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Cisco Meeting Server (CMS) con spazi impostati e in grado di effettuare chiamate
- Client API (come Poster, Postman) o
- [Guida API CMS](#)

Componenti usati

Le informazioni di questo documento si basano sulla versione 2.1 di CMS

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Il documento delinea i tipi di scenari:

- I partecipanti guest e host utilizzano URI o ID chiamata diversi
- Lo stesso URI viene utilizzato dai partecipanti guest e host in cui la differenziazione viene effettuata in base al PIN o al passcode (entrambi non vuoti)
- Lo stesso URI viene utilizzato dai partecipanti guest e host in cui la differenziazione viene effettuata in base al PIN o all'immissione del passcode (combinazione di vuoto/non vuoto)
- Un utente host è membro dello spazio e autorizzato tramite accesso a webRTC, gli utenti guest partecipano alla riunione con callID. Lo stesso URI e lo stesso callID vengono utilizzati dai partecipanti guest e host con PIN/passcode vuoti o non vuoti per gli utenti guest

Configurazione

Ci sono quattro possibilità di differenziazione tra Guest e Host partecipanti in CMS, descritti nei successivi 4 esempi, e sono principalmente basati su diversi **callLegProfiles** che determinano il comportamento in-call per quei partecipanti che si uniscono nello spazio.

In primo luogo, viene spiegato il metodo utilizzando un **URI** (o call-ID) diverso per i partecipanti ospiti e gli ospiti, e successivamente viene aggiunto utilizzando passcode (o timeout) diversi sullo stesso **URI**, per fare la distinzione tra i partecipanti ospiti e gli ospiti. Il terzo metodo di immissione di un timeout o di un PIN vuoto per gli utenti guest è stato introdotto come nuova funzionalità in CMS 2.1, come mostrato nella sezione 2.4 delle [note di rilascio](#). Il quarto metodo spiega come impostare l'accesso come Guest e Host sugli spazi con proprietari/membri assegnati e come rendere il membro dello spazio (proprietario) l'host dello spazio.

1) Configurazione con URI diversi

Questa è la configurazione di base disponibile prima della versione CMS 2.1 ed è la stessa di un ID chiamata diverso. Per differenziare l'accesso di un ospite o di un host sullo stesso spazio, è necessario eseguire i passaggi successivi:

1. Crea un **callLegProfile** guest (**needActivation = true**)
2. Creare un **callLegProfile** host (**needActivation = false**)
3. Assegna il **callLegProfile** guest a uno spazio nuovo o esistente (metodo di accesso predefinito)
4. Creare un nuovo **accessMethod** nello stesso spazio con un **URI** diverso (e call-ID) e assegnarvi l'host **callLegProfile**

Passaggio 1. Creare un **callLegProfile** guest (**needActivation = true**).

Un **callLegProfile** determina il comportamento durante la chiamata e per impostazione predefinita viene assegnato il comportamento ospite durante la chiamata allo spazio in modo che sia possibile in seguito disporre di un metodo di accesso diverso nello stesso spazio e consentire

all'host di partecipare.

Nota: È inoltre possibile assegnare questo valore a livello di tenant (`/api/v1/tenants/<tenant-ID>`) o a livello di sistema (`/api/v1/system/profiles`), ad esempio per applicare questo valore a tutti gli spazi (o per tenant), indipendentemente da come viene visualizzato nello spazio stesso. Tenere presente che l'allocazione più specifica di **callLegProfile** viene presa in considerazione per il comportamento in chiamata.

Il parametro **needActivation** è il più importante per il comportamento ospite/ospite poiché, se impostato su **true**, il partecipante non è in grado di ricevere o contribuire con audio e video fino a quando uno o più partecipanti **full/activator** (host) non si uniscono. Altri parametri su **callLegProfile** sono disponibili nella sezione 8.4.3 della [guida API](#), in cui possono essere rilevanti anche quelli mostrati in questa configurazione (a seconda dei requisiti):

- `presentazioneContributoConsentito`
- `Disattiva audio rx`
- `rxVideoMute`
- `deactivationMode` (disattiva Disconnessione | | `keepActivated`) e `deactivationModeTime` [azione da eseguire quando l'ultimo attivatore esce dalla chiamata]

Per creare il **callLegProfile** guest, eseguire una richiesta POST in `/api/v1/callLegProfiles` con i parametri preferiti e il parametro **needActivation** impostato su **true** in modo da poter eseguire una richiesta **GET** su tale **callLegProfile-ID** in seguito con questo risultato, ad esempio:

```
needsActivation>true
```

```
needsActivation>
```

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

Prendere nota di **callLegProfile-ID** come indicato in grassetto in quanto deve essere applicato allo spazio del passaggio 3 per l'accesso guest (predefinito).

Passaggio 2. Creare un host **callLegProfile** (**needActivation = false**).

In modo analogo, creare l'host **callLegProfile** per il comportamento in-call dell'host. Vengono applicati gli stessi parametri indicati in precedenza, sebbene sia possibile selezionarli in base alle proprie preferenze e requisiti. L'elemento principale, in questo caso, è l'impostazione del parametro **needActivation** su **false** per assegnargli il ruolo host.

È possibile crearlo tramite una richiesta POST in `/api/v1/callLegProfiles` con i parametri preferiti e il parametro **needActivation** impostato su **false** in modo che sia possibile eseguire una richiesta **GET** in tale **callLegProfile-ID** successivamente con il risultato seguente, ad esempio:

```
needsActivation>
```

```
false
```

```
needsActivation>
```

Prendere nota di **callLegProfile-ID** come indicato in grassetto in quanto deve essere applicato allo spazio **accessMethod** al passaggio 4 per l'accesso host.

Passaggio 3. Assegnare il **callLegProfile** guest a uno spazio nuovo o esistente (**accessMethod** predefinito).

Eseguire un comando PUT su uno spazio esistente (`/api/v1/coSpaces/<coSpace-ID>`) per adattare lo spazio o un comando POST su `/api/v1/coSpaces` per crearne uno nuovo con il **parametro** guest **callLegProfile** creato nel passaggio 1 come comportamento in-call per tale spazio. È inoltre possibile impostare i parametri **URI**, **passcode** e **call-ID** per tale spazio come indicato nella sezione 6.2 della [guida API](#).

Eseguire una richiesta GET su tale spazio (`/api/v1/coSpaces/<coSpace-ID>`) per verificare che **callLegProfile** guest sia associato a tale spazio, nonché il valore **URI** e **call-ID**. Un output di esempio con questo esempio creato **callLegProfile** guest nel passaggio 1 è questo con un valore **URI** `guest.space` e **call-ID** di 628821815 (senza passcode impostato):

```
uricallIdcallLegProfile>
```

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

```
callLegProfile>
```

Prendere nota dello spazio-ID contrassegnato in grassetto in quanto deve essere utilizzato per creare l'**accessMethod** su quello spazio specifico nel passaggio 4.

Passaggio 4. Creare un nuovo **accessMethod** in tale spazio con un **URI** (e un **call-ID**) diverso e assegnarvi l'host **callLegProfile**.

Si desidera creare un modo diverso per accedere allo spazio rispetto all'accesso guest, che è attualmente quello predefinito. A tale scopo, è necessario specificare un **accessMethod** sullo spazio stesso mediante un comando POST su `/api/v1/coSpaces/<coSpace-ID>/accessMethods` dove `coSpace-ID` è il valore contrassegnato in grassetto al passaggio 3 (`7cc797c9-c0a8-47cf-b519-8dc5a01f1ade`) al quale viene applicato il **callLegProfile** del passaggio 2, nonché i diversi **URI** e **call-ID**.

Dopo una richiesta GET per il metodo di accesso dello spazio (`/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>`), è necessario essere in grado di visualizzare un tipo di output simile a questo, in cui è possibile visualizzare un **URI** (`host.space`) e un **call-ID** (888) diversi dal **metodo di accesso** predefinito dello spazio e dall'host **callLegProfile** appositamente associato impostato nel passaggio 2:

```
uricallIdpasscodecallLegProfile>
```

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

```
callLegProfile>
```

Verifica

Ora è possibile accedere alla stessa riunione:

- Come ospite
 - chiamando l'URI `guest.space` (seguito dal dominio configurato nelle regole di corrispondenza chiamate)
 - immettendo il valore call-ID 628821815 tramite IVR o WebRTC join (senza passcode)
- Come host
 - chiamando l'URI `host.space` (seguito dal dominio configurato nelle regole di corrispondenza chiamate)
 - immettendo il valore 888 dell'ID chiamata tramite un join IVR o WebRTC (senza passcode)

Quando ci sono solo ospiti uniti allo spazio, tutti sono messi in una sala di attesa in attesa che l'host si unisca a loro. Una volta che un ospite si unisce, tutti gli ospiti e gli ospiti sono messi in conferenza. Se non ci sono più host uniti nello spazio ma ancora alcuni ospiti, questi tornano alla schermata di benvenuto come da configurazione di **deactivate** sul parametro **deactivationMode** sul guest **callLegProfile** come mostrato nel Passo 1.

2) Configurazione con gli stessi URI ma PIN/passcode host e non vuoti

Questa configurazione è simile a quella dell'esempio precedente ed è disponibile anche prima di CMS 2.1. Richiede sia al guest che all'host di immettere un PIN o un passcode non vuoto in modo che sia possibile effettuare una differenziazione in base al numero di caratteri che compongono lo stesso **URI**.

I passaggi di configurazione sono molto simili all'esempio di configurazione precedente:

1. Crea un **callLegProfile** guest (**needActivation = true**)
2. Creare un **callLegProfile** host (**needActivation = false**)
3. Assegnare il **callLegProfile** guest a uno spazio nuovo o esistente specificando un codice di accesso guest (PIN) (metodo di accesso predefinito)
4. Creare un nuovo **accessMethod** nello stesso spazio con lo stesso **URI** (call-ID diverso) e assegnargli il **callLegProfile** host includendo un passcode (PIN)

Passaggio 1. Creare un **callLegProfile** guest (**needActivation = true**).

È possibile utilizzare la stessa configurazione dell'esempio precedente 1 e persino la stessa

chiamata **LegProfile** guest (d4bfe12d-68cd-41c0-a671-48395ee170ab).

Passaggio 2. Creare un **callLegProfile** host (**needActivation = false**)

È possibile usare la stessa configurazione dell'esempio precedente 1 e anche la stessa configurazione host **callLegProfile** (7306d2c1-bc15-4dbf-ab4a-1cbdabd1912).

Passaggio 3. Assegnare il **callLegProfile** guest a uno spazio nuovo o esistente specificando un codice di accesso guest (PIN) (che corrisponde al **metodo di accesso** predefinito).

Analogamente, è possibile eseguire un'operazione PUT su uno spazio esistente (/api/v1/coSpaces/<cospace-ID>) o un'operazione POST per creare un nuovo spazio (/api/v1/coSpaces) con i parametri desiderati per l'URI, il passcode e il call-ID, ad esempio, nonché il **callLegProfile** guest (dal passaggio 1) assegnato a tale spazio in base alla sezione 6.2 della [guida API](#).

Se si esegue una richiesta GET su tale spazio, è necessario essere in grado di visualizzare un tipo di output simile a quello riportato di seguito in cui vengono visualizzati un **URI** di guestpin.space, un **call-ID** di 189, il **callLegProfile** guest creato in precedenza e un **passcode** di 789:

```
uricallIdcallLegProfile>
```

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

```
callLegProfile><
```

```
passcode>789
```

```
passcode>
```

Prendere nota dello spazio-ID contrassegnato in grassetto in quanto deve essere utilizzato per creare l'**accessMethod** su quello spazio specifico nel passaggio 4.

Passaggio 4. Creare un nuovo **accessMethod** in tale spazio con lo stesso **URI** (**call-ID** diverso) e assegnargli il **callLegProfile** dell'host includendo un passcode (PIN) dell'host.

In questo spazio è anche possibile creare un diverso metodo di accesso per gli host (poiché il **callLegProfile** guest è assegnato allo spazio stesso come opzione di join predefinita), proprio come nel primo esempio di configurazione. A tale scopo, è necessario utilizzare un comando **POST** su /api/v1/coSpaces/<coSpace-ID>/accessMethods con il valore **coSpace-ID** per il nostro spazio **22d9f4ca-8b88-4d11-bba9-e2a2f7428c46** evidenziato nel passaggio precedente. Con questo comando POST è possibile fornire parametri diversi, quali URI (guestpin.space, lo stesso di quello originale), call-ID (889), host **callLegProfile** come definito nel passaggio 2 e **passcode**

host o **PIN** (in questo caso 1234).

Se si esegue una richiesta GET su tale `accessMethod`, è necessario essere in grado di visualizzare un tipo di output simile che mostri lo stesso URI di `guestpin.space`, un **call-ID** di 889, il riferimento **callLegProfile** dell'host e il **PIN** dell'host di 1234:

```
uricallIdpasscode>1234
```

```
passcode><
```

```
callLegProfile>
```

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

```
callLegProfile>
```

Verifica

Ora è possibile accedere alla stessa riunione:

- Come ospite
 - chiamando l'**URI** `guestpin.space` (seguito dal dominio come configurato nelle regole di corrispondenza chiamate) e immettendo il **PIN** 789
 - immettendo il valore **call-ID** 189 tramite IVR o WebRTC join con **PIN** 789
- Come host
 - chiamando l'**URI** `guestpin.space` (seguito dal dominio come configurato nelle regole di corrispondenza chiamate) e immettendo il **PIN** 1234
 - immettendo il valore **call-ID** 889 tramite IVR o WebRTC join con **PIN** 1234

Quando ci sono solo ospiti uniti allo spazio, tutti sono messi in una sala di attesa in attesa che l'host si unisca a loro. Una volta che un ospite si unisce, tutti gli ospiti e gli ospiti sono messi in conferenza. Se non ci sono più host uniti nello spazio ma ancora alcuni ospiti, questi tornano alla schermata di benvenuto come da configurazione di **deactivate** sul parametro **deactivationMode** sul guest **callLegProfile** come mostrato nel Passo 1.

3) Configurazione utilizzando gli stessi URI con combinazione di PIN guest vuoto e PIN host non vuoto

Questa configurazione è disponibile solo a partire dalla versione 2.1 di CMS in avanti a causa di alcuni comandi API appena aggiunti di `passcodeMode` e `passcodeTimeout` nella sezione **callProfile**. In questo modo, è possibile inserire un PIN vuoto (immettendo # o timeout) mentre

l'host dispone di un PIN per accedere allo spazio e attivarlo. Il **callProfile** controlla l'esperienza in chiamata per le chiamate SIP (incluso Lync) e pertanto non è applicabile ai client CMA (sia thick client che WebRTC).

I passaggi di configurazione sono simili a quelli dell'esempio 2, con l'aggiunta del profilo di chiamata:

1. Crea un **callLegProfile** guest (**needActivation = true**)
2. Creare un **callLegProfile** host (**needActivation = false**)
3. Creare un **callProfile** con la configurazione **passcodeMode** desiderata e **passcodeTimeout**
4. Assegnare **callLegProfile** e **callProfile** guest del passaggio 3 a uno spazio nuovo o esistente specificando un codice di accesso guest (PIN) (metodo di accesso predefinito)
5. Creare un nuovo **accessMethod** nello stesso spazio con lo stesso **URI (call-ID diverso)** e assegnargli il **callLegProfile** host includendo un passcode (PIN)

Poiché le configurazioni sono identiche agli esempi di configurazione 1 e 2, esistono riferimenti a tali configurazioni. In realtà per il test è stato utilizzato lo stesso spazio dell'esempio 2, ma aggiunto con callProfile.

Passaggio 1. Creare un **callLegProfile** guest (**needActivation = true**).

È possibile utilizzare la stessa configurazione dell'esempio precedente 1 e persino la stessa funzione guest callLegProfile (**d4bfe12d-68cd-41c0-a671-48395ee170ab**).

Passaggio 2. Creare un host **callLegProfile** (**needActivation = false**).

Come dimostrato, è possibile utilizzare la stessa configurazione dell'esempio precedente 1 e anche lo stesso host callLegProfile (**7306d2c1-bc15-4dbf-ab4a-1cbdabd1912**).

Passaggio 3. Creare un **callProfile** con la configurazione **passcodeMode** e **passcodeTimeout** desiderata.

È possibile creare un callProfile che determina l'esperienza in chiamata per le chiamate SIP (incluso Lync). Qui ci sono alcune possibili configurazioni, come ad esempio consentire la registrazione o lo streaming o il limite massimo di partecipanti, ma l'attenzione qui è concentrata sulle nuove API aggiunte da CMS 2.1 relative alla gestione del passcode. Gli altri parametri sono disponibili nella sezione 8.2 della [guida API](#).

Il comportamento del passcode è determinato da due parametri:

- **passcodeMode**

- **obbligatorio** : l'IVR attende per sempre che un utente immetta il PIN o il numero per un PIN vuoto (per gli ospiti)

- **timeout** : l'IVR attende un periodo di **passcodeTimeout** di secondi prima che il partecipante immetta il PIN e, se non è stata effettuata alcuna immissione entro tale periodo, presuppone che sia stato immesso un PIN vuoto (#)

- **passcodeTimeout**: deve essere impostato solo quando **passcodeMode** è impostato su timeout e controlla il periodo di tempo prima di interpretare il passcode come vuoto

Per creare callProfile, eseguire un comando POST su `/api/v1/callProfiles` (o PUT su `/api/v1/callProfiles/<callProfile-ID>` se si desidera modificarne uno esistente) con i parametri

desiderati per **passcodeMode** e **passcodeTimeout**. Se si esegue un comando **GET** su **callProfile** specifico, è necessario ottenere un tipo di risultato simile, ad esempio se la modalità è stata impostata come timeout e un valore di timeout di 5 secondi:

```
passcodeMode>timeout
```

```
passcodeMode><
```

```
passcodeTimeout>5
```

```
passcodeTimeout>
```

Prendere nota di **callProfile-ID** come indicato in grassetto perché deve essere utilizzato per assegnare allo spazio per avere questo comportamento durante la chiamata nel passaggio 4.

Passaggio 4. Assegnare i parametri **callLegProfile** e **callProfile** guest del passaggio 3 a uno spazio nuovo o esistente specificando un codice di accesso guest (PIN) (che è il metodo di accesso predefinito).

Analogamente, è possibile eseguire un'operazione PUT su uno spazio esistente (/api/v1/coSpaces/<cospace-ID>) o un'operazione POST per creare un nuovo spazio (/api/v1/coSpaces) con i parametri desiderati per l'URI e **call-ID**, ad esempio, nonché per l'elemento guest **callLegProfile** (dal passaggio 1). La differenza rispetto agli esempi precedenti è rappresentata dal **callProfile** del passaggio 3 e dal fatto che non è stato assegnato alcun passcode.

Se si esegue una richiesta GET su tale spazio, è necessario essere in grado di visualizzare un tipo di output simile a quello dell'esempio, in cui viene visualizzato l'URI di guestpin.space, un **call-ID** di 189, il **callLegProfile** guest creato in precedenza e il **callProfile** impostato nel passaggio 3:

```
uricallIdcallLegProfile>
```

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

```
callLegProfile><
```

```
callProfile>
```

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

```
callProfile>
```

Prendere nota dell'ID dello spazio contrassegnato in grassetto in quanto deve essere utilizzato per creare l'accessMethod su quello spazio specifico nel passaggio 5.

Passaggio 5. Creare un nuovo **accessMethod** nello stesso spazio con lo stesso **URI** (**call-ID** diverso) e assegnarvi l'host **callLegProfile** includendo un passcode host (PIN).

In questo spazio è anche possibile creare un diverso metodo di accesso per gli host (poiché il **callLegProfile** guest è assegnato allo spazio stesso come opzione di join predefinita), proprio come nel primo esempio di configurazione. A tale scopo, viene utilizzato un comando **POST** su **/api/v1/coSpaces/<coSpace-ID>/accessMethods** in cui il valore **coSpace-ID** viene sostituito con il valore **22d9f4ca-8b88-4d11-bba9-e2a2f7428c46** come evidenziato nel passaggio precedente per questo caso. Con questo comando **POST**, è possibile fornire parametri diversi come **URI** (guestpin.space, lo stesso di quello originale), **call-ID** (889), host **callLegProfile** come definito nel passaggio 2 e il passcode host o PIN (1234 in questo caso).

Se si esegue una richiesta GET su tale accessMethod, è necessario essere in grado di visualizzare un tipo di output simile che mostri lo stesso URI di guestpin.space, un **call-ID** di 889, il riferimento **callLegProfile** dell'host e il **PIN** dell'host di 1234:

```
uricallIdpasscode>1234
```

```
passcode><
```

```
callLegProfile>
```

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

```
callLegProfile>
```

Verifica

Ora è possibile accedere alla stessa riunione:

- Come ospite
 - chiamando l'**URI** guestpin.space (seguito dal dominio come configurato nelle regole di corrispondenza chiamate) e immettendo **#** come **PIN** o lasciandolo scadere dopo 5 secondi
 - immettendo il valore **call-ID** 189 tramite IVR o WebRTC join
- Come host
 - chiamando l'**URI** guestpin.space (seguito dal dominio come configurato nelle regole di corrispondenza chiamate) e immettendo il **PIN** 1234

- immettendo il valore **call-ID** 889 tramite IVR o WebRTC join con **PIN** 1234

4) Un utente ospite è membro dello spazio e autorizzato tramite accesso webRTC, gli utenti ospiti si uniscono al meeting con callID. Lo stesso URI e lo stesso callID vengono utilizzati dai partecipanti guest e host con PIN/passcode vuoti o non vuoti per gli utenti guest

Per differenziare l'accesso Guest/Host sullo stesso spazio per i membri e i non membri dello spazio, è necessario eseguire i passi successivi:

1. Crea un **callLegProfile** guest(**needActivation = true**)
2. Creare un **callLegProfile** host(**needActivation = false**)
3. Assegna il **callLegProfile** del guest a uno spazio nuovo o esistente (metodo di accesso predefinito)
4. Creare un nuovo **accessMethod** nello stesso spazio con lo stesso **URI** (e call-ID) e assegnarvi l'host **callLegProfile**
5. Assegnare sownerJID dell'utente allo stesso spazio. (se non assegnato)
6. Aggiungere l'ID proprietario come utente membro allo stesso spazio e assegnare **hostcallLegProfile** a tale utente membro

Passaggio 1. Creare un **callLegProfile** guest (**needActivation = true**).

Nell'esempio viene utilizzata la stessa configurazione dell'esempio precedente 1 e la funzione guest **callLegProfile** (**bfe7d07f-c7cb-4e90-a46e-4811bbaf6978**).

Prendere nota di **callLegProfile-ID** come indicato in grassetto in quanto deve essere applicato allo spazio del passaggio 3 per l'accesso guest.

Passaggio 2. Creare un **callLegProfile** host (**needActivation = false**)

Nell'esempio viene utilizzata la stessa configurazione dell'esempio precedente 1 e dell'host **callLegProfile** (**0e76e943-6d90-43df-9f23-7f1985a74639**).

Prendere nota di **callLegProfile-ID** come indicato in grassetto in quanto deve essere applicato allo spazio **accessMethod** al passaggio 4 per l'accesso host e al membro di **coSpace** al passaggio 6.

Passaggio 3. Assegnare il **callLegProfile** guest a uno spazio nuovo o esistente (essendo il metodo di accesso predefinito).

Eseguire un comando PUT su uno spazio esistente (**/api/v1/coSpaces/<coSpace-ID>**) per adattare lo spazio o un comando POST su **/api/v1/coSpaces** per crearne uno nuovo con il parametro guest **callLegProfile** creato nel passaggio 1 come comportamento in-call per tale spazio. È inoltre possibile impostare i parametri **URI** e **call-ID** per tale spazio in base alle proprie esigenze, come indicato nella sezione 6.2 della [guida alle API](#).

Eseguire una richiesta GET su tale spazio (**/api/v1/coSpaces/<coSpace-ID>**) per verificare che **callLegProfile** guest sia associato a tale spazio, nonché il valore **URI** e **call-ID**. Un output di esempio con questo esempio creato in guest **callLegProfile** nel passaggio 1 è questo con un valore **URI** di **global** e **call-ID** di 1234 (senza passcode impostato), **nonMemberAccessset to 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>0w4O2zTTF0WdL4ymF8D0_A</secret>
  <defaultLayout>allEqual</defaultLayout>
</coSpace>

```

Prendere nota dello spazio-ID contrassegnato in grassetto in quanto deve essere utilizzato per creare l'**accessMethod** su quello spazio specifico nel passaggio 4.

Passaggio 4. Creare un nuovo **accessMethod** nello spazio con lo stesso **URI** (e **call-ID**) e assegnarvi l'host **callLegProfile**.

Si desidera creare un modo diverso per accedere allo spazio rispetto all'accesso guest, che è attualmente quello predefinito. A tale scopo, specificare un **accessMethod** sullo spazio stesso tramite un POSTcommand on **/api/v1/coSpaces/<coSpace-ID>/accessMethods**. In questo caso, il **coSpace-ID** rappresenta il valore in grassetto contrassegnato nel passaggio 3 (**96d28acb-86c6-478d-b81a-a37ffb0adafc**) a cui è applicato l'host **callLegProfile** del passaggio 2, nonché lo stesso **URI** e **call-ID**. È possibile aggiungere un passcode non vuoto per gli host che si connettono tramite **callID** (senza aver eseguito l'accesso come utente tramite **webRTC**).

Dopo una richiesta GET per il metodo di accesso allo spazio (**/api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID>**), è necessario essere in grado di visualizzare un tipo di output simile a questo, in cui è possibile visualizzare lo **stesso URI** (globale) e **call-ID** (1234) nonché l'host **callLegProfile** appositamente associato impostato nel passaggio 2 e il **passcode 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>kff01zTTE0feL4fsdf43w_B </secret>
</accessMethod>

```

Passaggio 5. Assegnare il proprietarioJid dello spazio. (se non assegnato). Aggiungere **ownerJID** allo spazio specificando **ownerJid** (user1@evacanoalone.net) sullo spazio con aPUTcommand on **/api/v1/coSpaces/<coSpace-ID>**

Dopo una richiesta GET su quello spazio, è necessario poter verificare che **ownerId** e **ownerJid** sono stati assegnati allo spazio:

```

<?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>
```

Annotare l'ID proprietario (1d942281-413e-4a2a-b776-91a674c3a5a9).

Passaggio 6. Aggiungere l'ID proprietario (1d942281-413e-4a2a-b776-91a674c3a5a9) dal passaggio 5 come utente membro allo spazio e assegnare hostcallLegProfil a tale utente membro. A tale scopo, specificare userJidandhost callLegProfilsullo spazio stesso (specificando coSpaceID) con un comando POST (/api/v1/coSpaces/<coSpaceID>/coSpaceUsers). Altri parametri su **coSpaceUsers** sono disponibili nella sezione 6.4.2 della [guida API](#), in cui possono essere rilevanti anche quelli visualizzati in questa configurazione:

```
<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>
```

Verificare che l'utente membro sia stato aggiunto allo spazio tramite aGETcommand (/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>
```

Prendere nota dell'ID utente (se diverso da ownerID modulo passaggio 5). Verificare che l'host callLegProfile sia stato assegnato a coSpaceUser da una richiesta GET che specifica coSpaceIDanduserID (/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>
```

Verifica

Ora è possibile accedere alla stessa riunione:

- Come ospite

- chiamando l'URI (seguito dal dominio configurato nelle regole di corrispondenza delle chiamate)

- immettendo il valore call-ID 1234 tramite IVR o WebRTC join (senza passcode)

- Come host

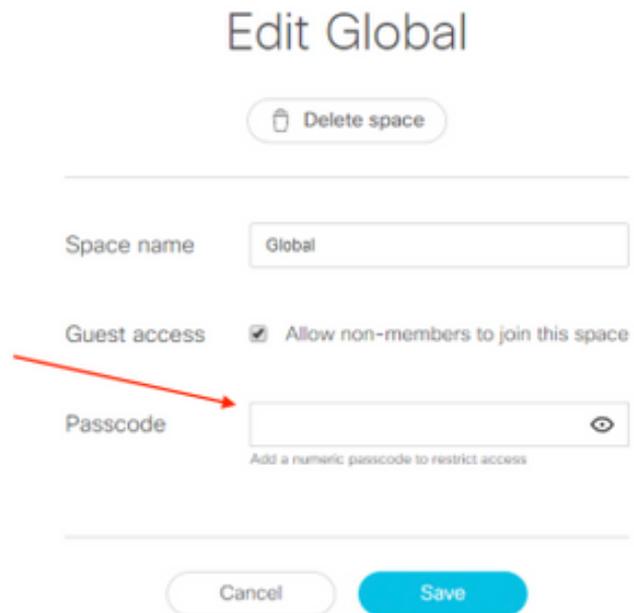
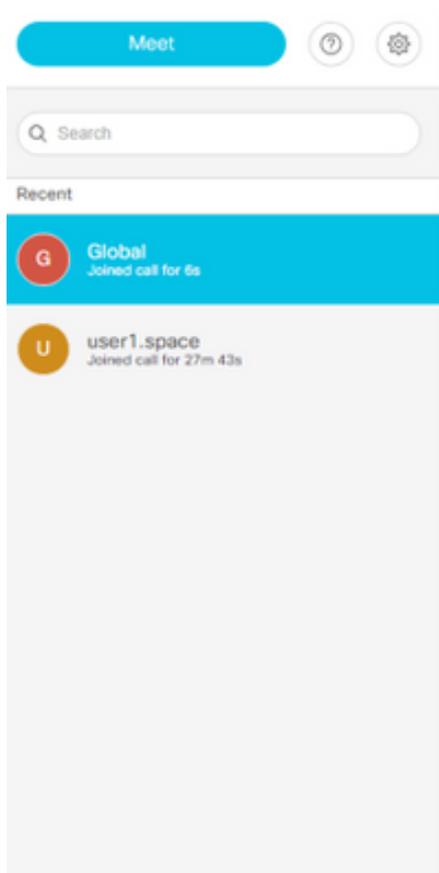
Effettuando l'accesso come utente (membro dello spazio con "host" assegnato `callLegProfile`, con `user1@evacanoalone.net` in questo scenario) tramite webRTC e unirsi allo spazio (URI "globale").

- chiamando l'URI "globale" (seguito dal dominio configurato nelle regole di corrispondenza delle chiamate) e il passcode 12345.

- immettendo il valore call-ID 1234 tramite IVR o WebRTC join (con passcode host 12345)

Quando ci sono solo ospiti uniti allo spazio, tutti sono messi in una sala di attesa in attesa che l'host si unisca a loro. Una volta che un ospite si unisce, tutti gli ospiti e gli ospiti vengono messi nella conferenza. Se non ci sono più host uniti nello spazio ma ancora alcuni ospiti, questi tornano alla schermata di benvenuto come da configurazione di **deactivate** sul parametro **deactivationMode** sul guest **callLegProfile** come mostrato nel Passo 1.

L'host (proprietario/membro) può impostare (modificare/rimuovere) una password per gli ospiti direttamente nell'app WebRTC o disabilitare completamente l'accesso non membro (ospite) per lo spazio:



Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione.

La registrazione di CMS mostra brevemente quando si partecipa come guest o quando il primo host si unisce, ma è meglio verificare utilizzando GET le richieste callProfile, nonché le definizioni callLegProfile guest e host e la relativa allocazione nei rispettivi metodi di accesso (o metodo di accesso predefinito) o a un livello superiore (livello globale o livello tenant).

È possibile seguire questa struttura per ottenere tutte le informazioni:

1. **GET su /api/v1/callProfiles** (se viene utilizzato con **passcodeMode**)
>verificare in dettaglio l'ID profilo chiamata desiderato utilizzando **GET** on **/api/v1/callProfiles/<callProfile-ID>**
2. **GET on /api/v1/callLegProfiles**
>verificare in dettaglio gli ID callLegProfile desiderati di guest e host utilizzando **GET** on **/api/v1/callProfiles/<callProfile-ID>**
3. **GET on /api/v1/coSpaces**
>verificare in dettaglio l'ID spazio desiderato utilizzando **GET** on **/api/v1/coSpaces/<coSpace-ID>**
>verificare che l'callProfile-ID (passaggio 1) e l'guest callLegProfile (passaggio 2) desiderati siano associati a questo spazio
[in caso contrario, controllare gli elementi meno specifici come il livello tenant (**/api/v1/tenants/<tenant-ID>**) o globale (**/api/v1/system/profiles**)]

4. GET on /api/v1/coSpaces/<IDcoSpace>/accessMethods

>verificare in dettaglio l'accessMethod desiderato utilizzando GET on /api/v1/coSpaces/<coSpace-ID>/accessMethods/<accessMethod-ID> per verificare se l'host callLegProfile è assegnato

Nel log CMS mostrato in questo esempio, arrivano i primi due partecipanti ospiti (*chiamando 38* da 2000@steven.lab e *chiamando 39* da 1060@steven.lab) che si recano nello spazio guestpin.space@acano.steven.lab e quindi l'host si unisce. Si può vedere dal frammento che per gli ospiti ci informa su cosa deve essere fatto con esso (**da disattivare**) e si può vedere questo comportamento per quelle chiamate cambiare quando l'host (stejanss.movi@steven.lab) si unisce allo spazio (**cessando di essere disattivato**). Analogamente, è possibile visualizzare di nuovo la stessa registrazione quando gli ospiti si spostano di nuovo nella hall non appena non ci sono più host nello spazio (**da disattivare**).

```
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
```

Informazioni correlate

- [Documentazione e supporto tecnico – Cisco Systems](#)

- [Documentazione CMS](#)