

# Cisco Meeting Server

## Cisco Meeting Server リリース 3.7 以降 イベントガイド

2024 年 9 月 27 日

# 目次

変更履歴 .....	3
1 はじめに .....	4
1.1 サブスクリプションの概要 .....	4
1.1.1 サブスクリプションの概要 .....	4
2 イベントリソースと購読可能な要素 .....	6
2.0.1 イベントリソースと購読可能な要素 .....	6
2.1 callInfo .....	6
2.2 callRoster .....	8
2.3 通話 .....	10
3 Call Bridge のグループとクラスタ .....	11
3.0.1 Call Bridge のグループとクラスタ .....	11
4 認証フローの例 .....	12
4.0.1 認証フローの例 .....	12
5 メッセージ フローの例 .....	15
5.0.1 メッセージ フローの例 .....	15
6 WebSocket の仕様 .....	22
Cisco の法的情報 .....	23
Cisco の商標または登録商標 .....	24

## 変更履歴

日付	変更の概要
2023年3月16日	3.7の新しいバージョンです。
2022年8月23日	3.6の新しいバージョンです。
2022年4月20日	3.5の新しいバージョンです。
2021年8月24日	3.3の新しいバージョンです。
2021年4月9日	3.2の新しいバージョンです。

# 1 はじめに

ミーティングサーバは、ミーティングサーバ上で発生する変更をリアルタイムで「イベントクライアント」に通知することができます。 Meeting Server は、イベントのためのサーバーとして機能し、イベントクライアントは、例えば、ウェブベースの管理アプリケーションである場合があります。 Cisco Meeting Management はイベントクライアントとして機能します。

---

**メモ:** API クライアントの構築と同様に、独自のイベント クライアントを構築できます。 イベントクライアントは、HTTP および WebSocket ライブラリをサポートする必要があります、どちらも Python などの一般的なスクリプト言語で使用できます。 Meeting Server のイベントポートはウェブ管理で設定したものと同じです。通常はインタフェース A の TCP ポート 443 です。

---

Meeting Server の API リソースを継続的にポーリングする代わりに、イベントクライアントはイベントリソースをサブスクライブして更新を受け取ることができます。例えば、イベントクライアントと Meeting Server 間の WebSocket 接続を確立した後、イベントクライアントはイベントリソース `callRoster` をサブスクライブし、アクティブな会議の参加者リストの更新を受け取り、新しい参加者が参加したときや既存の参加者がレイアウトを変更したときなどを確認することができます。

ミーティングサーバは、以下の 3 つの登録可能なイベントリソースをサポートしています。

- `callInfo` は特定の電話会議についての情報を提供し、
- `callRoster` は、電話会議の各参加者に関する情報を提供し、
- `calls` は、アクティブな電話会議に関する情報を提供します。

登録可能な各イベントリソースには、登録可能な要素があります。 [第 2 項](#)を参照してください。

## 1.1 サブスクリプションの概要

### 1.1.1 サブスクリプションの概要

イベントクライアントが Meeting Server に登録する際、登録にはイベントクライアントが登録を希望するリソースの一覧が表示されます。これは、アクティブなサブスクリプションを希望するリソースの完全なリストである必要があります。イベントクライアントが既存のサブスクリプションのセットに新しいリソースを追加する必要がある場合、既存のサブスクリプション先リソースすべてを新しいリクエストに含める必要があります。すべてのアクティブなサブスクリプションを停止するには、イベントクライアントは空のリクエストを提供する必要があります（これは WebSocket 接続が切断された場合にも発生します）。

サブスクリプションリクエストで登録される各リソースには、イベントクライアントによって固有の番号が割り当てられるため、Meeting Server が更新を提供する際に、クライアントはそのサブスクリプションが参照されているかを把握できます。 イベントクライアントが以前と同じリソースをサブスクライブしているが、識別子の数字が異なる場合、Meeting Server はこれを新しいサブスクリプションとして扱い、古いサブスクリプション リクエストは効果的に破棄されます。

Meeting Server が新規のサブスクリプション リクエストを受け取ると、まず簡単な「ack」を返信し、サブスクリプション リクエストを受け取り、現在処理中であることを伝えます。 この段階での肯定的な応答は、Meeting Server がリクエストを受け取ったことを意味するだけです。サブスクリプション リクエストの各要素の更新されたステータスを示す 1 つ以上の追跡情報があり、その後、そのリソースの実際の更新が行われます。

例えば、イベントクライアントが、アクティブな電話会議に関する情報と特定の電話会議の参加者リストのサブスクライブを要求する場合、まず「ack」を受け取ります、次に両方のサブスクリプションが「保留中」とであると伝える「subscriptionUpdate」メッセージを返します。これは Meeting Server がまだサブスクリプションを設定中であることを意味します。しばらくして、イベント クライアントは、アクティブな電話会議リストのサブスクリプションが「アクティブ」で、参加者リストのサブスクリプションがまだ「保留中」という更新情報を取得する場合があります。 Meeting Server は、この段階でアクティブな電話会議リストの更新情報も提供し始めます。サブスクライバにサブスクリプションが「アクティブ」とすると通知されるまで、サブスクライブしたリソースに関する実際の更新は行われません。 イベントクライアントがサブスクライブした参加者リストの電話会議が存在する場合（つまり、指定された GUID がまだアクティブな電話会議に対応している場合）、イベントクライアントは、アクティブな電話会議のサブスクリプションと参加者リストのサブスクリプションの両方が「アクティブ」であることを示すサブスクリプションの更新を受け取ります。 イベントクライアントはこの時点から参加者リストの更新を受け取り始めます。 参加者リストサブスクリプションの会議の GUID がアクティブな会議に正常に解決されなかった場合、サブスクリプションのステータスは「無効化」になります。 これはまた、電話会議が終了したとき、またはイベントクライアントがサブスクリプションを解除したときに、後続の「subscriptionUpdate」メッセージでサブスクリプションが変更される状態でもあります。

イベントリソースにサブスクライブする際のメッセージの流れについての詳細は、[第 5 項](#)を参照してください。

## 2 イベントリソースと購読可能な要素

### 2.0.1 イベントリソースと購読可能な要素

バージョン 2.4 から、Meeting Server は次のイベントリソースの「イベントクライアント」にリアルタイム情報を提供するために、イベントの使用をサポートします。

- [callInfo](#) は、特定の電話会議に関する情報を提供し、
- [callRoster](#) は、電話会議の各参加者に関する情報を提供し、
- [calls](#) は、アクティブな電話会議に関する情報を提供します。

## 2.1 callInfo

表 1: サブスクリブ可能なイベントリソースを使用して利用できる特定の電話会議に関する情報  
callInfo

名前	値	説明
要求パラメータ		
call	ID	更新情報を受け取る電話会議の ID。
購読可能な要素		
name	文字列	電話会議の名前。
participants	数値	現在電話会議に参加している参加者数。
distributedInstances	数値	Call Bridge クラスタ全体に存在する、この電話会議の分散インスタンスの数（非クラスタ電話会議の場合は 0）。
recording	active inactive	次のいずれか: <i>active</i> : この会議は現在録画中です <i>inactive</i> : この会議は現在録画されていません。
endpointRecording	active inactive	次のいずれか: <i>active</i> : この電話会議は現在端末により外部から録画されます (Lync クライアント) <i>inactive</i> : この電話会議は現在端末により外部から録画されていません (Lync クライアント)。

名前	値	説明
streaming	active  inactive	次のいずれか:  <i>active</i> : この会議は現在ストリーミング中です  <i>inactive</i> : この会議は現在ストリーミングされていません。
ロック状態	locked  unlocked	次のいずれか:  <i>locked</i> : この会議は現在ロックされています  <i>unlocked</i> : この会議は現在ロック解除状態です。
callType	coSpace  adHoc  lyncConferencing  forwarding	次のいずれか:  <i>coSpace</i> : この呼び出しは、coSpace のインスタンス化です  <i>adHoc</i> : アドホックの多者通話です  <i>lyncConferencing</i> - この通話はミーティングサーバから Lync で主催されている電話会議への接続です  <i>forwarding</i> : 転送された通話、つまり「ゲートウェイ」通話。
callCorrelator	ID	この値を使用して、複数の Call Bridges に分散されるが、すべてが同じ coSpace 内の同じ通話またはアドホック通話のいずれかの通話セッションを識別することができます。  注 : coSpace内の通話の場合、callCorrelator 値は coSpace の有効期間中は同じになります。アドホック コールごとに、値が動的に生成されます。
joinAudioMuteOverride	true  false	次のいずれか:  <i>true</i> : 会議に参加する新しい参加者はミュートされます  <i>false</i> : 会議に参加する際、新しい参加者はミュートされません。設定されていない場合、これが既定となります。

## 2.2 callRoster

表 2 : サブスクライブ可能なイベントリソース `callRoster` を使用して取得できる電話会議の各参加者に関する情報

名前	値	説明
<b>要求パラメータ</b>		
<code>call</code>	ID	参加者の更新情報を受信する電話会議の ID。
<b>購読可能な要素</b>		
<code>name</code>	文字列	参加者の表示名。
<code>uri</code>	URI ユーザー部分	この参加者に関連する URI。
<code>state</code>	initial  ringing  connected  onHold	この参加者の現在の信号状態を反映します。
<code>direction</code>	incoming  outgoing	次のいずれか:  <i>incoming</i> : この参加者は電話会議にダイヤルイン (リモート SIP デバイスが Meeting Server への接続を開始)  <i>outgoing</i> : この参加者は電話会議に参加するためにダイヤルアウト (Meeting Server からリモート SIP デバイスへのコールレグが確立)。
<code>audioMuted</code>	true  false	次のいずれか :  <i>true</i> : Meeting Server はこの参加者の音声をミュートにしている  <i>false</i> : Meeting Server はこの参加者の音声をミュートにしていない。
<code>videoMuted</code>	true  false	次のいずれか :  <i>true</i> : Meeting Server はこの参加者のビデオをミュートにしている  <i>false</i> : Meeting Server はこの参加者のビデオをミュートにしていない。
<code>importance</code>	数値	この参加者の重要性の値。重要度が設定されていない、または設定解除に変更された場合は NULL。

名前	値	説明
レイアウト	allEqual  speakerOnly  telepresence  stacked  allEqualQuarters  allEqualNinths  allEqualSixteenths  allEqualTwentyFifths  onePlusFive  onePlusSeven  onePlusNine  automatic  onePlusN	この参加者が現在使用しているレイアウトです。
アクティブなスピーカー	true  false	次のいずれか:  <i>true</i> : この参加者は現在この電話会議のアクティブスピーカーと見なされている  <i>false</i> : この参加者は現在この電話会議のアクティブスピーカーと見なされていない
presenter	true  false	次のいずれか:  <i>true</i> : この参加者はこの電話会議でプレゼンテーションを行っている (画面を共有している)  <i>false</i> : この参加者は現在会議でプレゼンテーションを行っていない。
エンドポイント録画	アクティブ  非アクティブ	次のいずれか:  <i>active</i> : この参加者は現在、会議を録音している  <i>inactive</i> : この参加者は現在、会議を録音していない。
移動できる	true false	この参加者を movedParticipant API コマンドを使用して移動できるかどうかを示します。(バージョン 2.6 以降)
canMoveToLobby	true false	この参加者をロビーに移動できるかどうかを示します。(バージョン 3.5 から)
移動した参加者	ID	この参加者が参加者の移動の一部として作成された場合、ID は移動元の参加者の GUID です。(バージョン 2.6 以降)
movedParticipantCallBridge	ID	この参加者が参加者の移動の一部として作成された場合、ID はこの参加者の移動元の電話会議をホストしている Call Bridge の GUID です。(バージョン 2.6 以降)

## 2.3 通話

表 3 : サブスクリブ可能なイベントリソース `calls` を使用して利用可能なアクティブな電話会議に関する情報

名前	値	説明
<b>必須の応答要素</b>		
<code>call</code>	ID	要素が更新された電話会議の ID です。
<b>購読可能な要素</b>		
<code>name</code>	String	電話会議の名前。
参加者	数値	現在電話会議に参加している参加者数。
分散インスタンス	数値	Call Bridge クラスタ全体に存在する、この電話会議の分散インスタンスの数（非クラスタ電話会議の場合は 0）。
レコーディング	アクティブ  非アクティブ	次のいずれか:  <i>active</i> : この会議は現在録画中です  <i>inactive</i> : この会議は現在録画されていません。
エンドポイント録画	アクティブ  非アクティブ	次のいずれか:  <i>active</i> : この電話会議は現在端末により外部から録画されています (Lync クライアント)  <i>inactive</i> : この電話会議は現在端末により外部から録画されていません (Lync クライアント)。
ストリーミング	アクティブ  非アクティブ	次のいずれか:  <i>active</i> : この会議は現在ストリーミング中です  <i>inactive</i> : この会議は現在ストリーミングされていません。
ロック状態	locked  notLocked	次のいずれか:  <i>locked</i> : この会議は現在ロックされている  <i>notLocked</i> : この会議は現在ロックされていない。
<code>callType</code>	coSpace  forwarding  adHoc  lyncConferencing	次のいずれか:  <i>coSpace</i> : この会議は、coSpace のインスタンス  <i>forwarding</i> : 転送された「ゲートウェイ」通話 アドホック - アドホックの多者通話です  <i>lyncConferencing</i> : このコールレグは Lync 電話会議に参加。
<code>callCorrelator</code>	ID	コールのすべての分散インスタンスで同じ相関関係子 GUID。

## 3 Call Bridge のグループとクラスタ

### 3.0.1 Call Bridge のグループとクラスタ

電話会議が Call Bridge グループ（またはクラスタ）全体で主催される場合、各 Meeting Server がアクティブな電話会議とそれが 1 つ以上の参加者を持つ任意の会議の名簿リスト情報を知るために、グループ内の Call Bridges 間でメッセージがやり取りされます。

サブスクリプションにより、電話会議のローカルおよびリモートの参加者の更新情報を受け取りますが、リモートの参加者が電話会議を退席すると、パラメータ「reason」は提供されません。電話会議を主催している Call Bridge のサブスクリプションのみが、参加者が電話会議を退席した理由を受け取ります。Call Bridge グループまたはクラスタ全体のアクティブな電話会議すべての情報を受信するには、イベントクライアントはすべての Meeting Server でイベントリソース `calls` をサブスクライブする必要がありますが、イベントリソース `callInfo` または `callRoster` を見るためには、電話会議を報告する Meeting Server の 1 つにだけサブスクライブする必要があります。

その Call Bridge 上でホストされている参加者が会議から退出したため、Meeting Server が電話会議から参加しなくなった場合、Call Bridge はサブスクリプションを非アクティブ化し、イベントクライアントは別の Meeting Server を選ぶ必要があります。

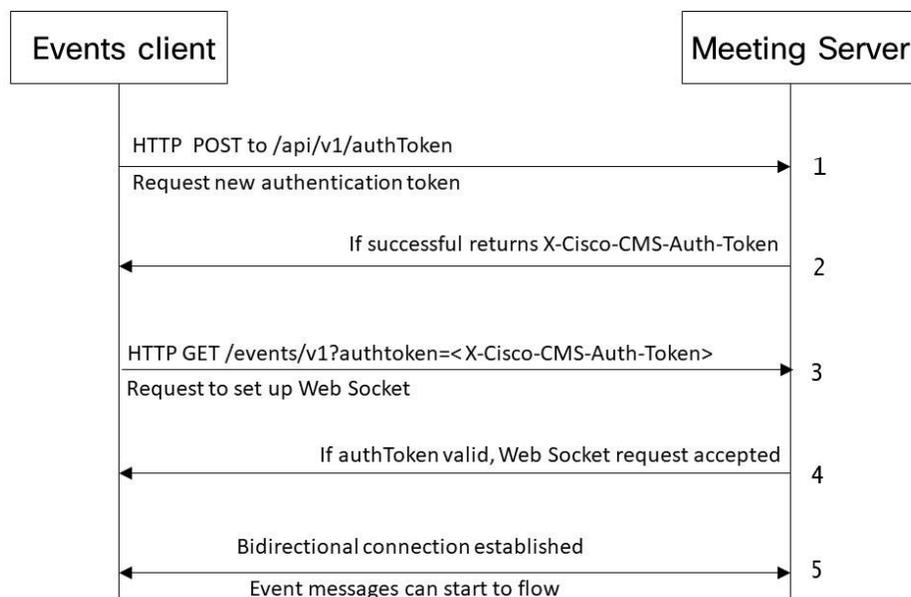
## 4 認証フローの例

### 4.0.1 認証フローの例

イベントクライアントは、Meeting Server 上のイベントのサブスクライブを許可される前に、Meeting Server によって認証され、イベントクライアントと Meeting Server 間に [WebSocket プロトコル](#) を使用した接続が確立される必要があります。この接続により、イベントクライアントはミーティングサーバの API を頻繁にポーリングする必要なく、イベント情報を受信できます。

図 1 は、WebSocket を確立するために必要なイベントクライアントと Meeting Server 間のコールフローを示しています。イベントクライアントが使用する HTTP POST および GET をミーティングサーバに送信するには、Python または類似のコーディング言語を使用する必要があります。

図 1 : イベントクライアントと Meeting Server 間で WebSocket を確立するコールフローの概要



1. イベントクライアントは、新しい認証トークンを発行するために、`/api/v1/authTokens` に POST します。例えば、リクエストは次のようになります。

```
POST /api/v1/authTokens HTTP/1.1\r\n
Origin: http://xx.xxx.xxx.xxx:8080\r\n
Content-length: 0\r\n
Host: xx.xxx.xxx.xxx:8080\r\n
Accept: */*\r\n
Connection: keep-alive\r\n
Authorization: Basic Ym9iOmJ1aWxkZXI=\r\n
\r\n
```

- Meeting Server からの正常な応答は、「権限」が十分な権限を持つユーザーアカウントに関連していると想定すると、次の形式になります。

```
HTTP/1.1 200 OK\r\n
X-Cisco-CMS-Auth-Token: 7174c102-61b3-47a6-8ff2-b86256cca958\r\n
Connection: close\r\n
\r\n
```

返された「X-Cisco-CMS-Auth-Token」は、次の段階である WebSocket 接続自体で使用されます。

- クライアントは WebSocket 接続をセットアップするために、次の形式の別の HTTP リクエストを行います。

```
GET /events/v1?authToken=7174c102-61b3-47a6-8ff2-b86256cca958 HTTP/1.1\r\n/
Host: xx.xxx.xxx.xxx\r\n/
接続: アップグレード\r\n/

プラグマ: no-cache\r\n/

Cache-Control: no-cache\r\n/
アップグレード: websocket\r\n/

オリジン: http://xx.xxx.xxx.xxx:8080\r\n/

Sec-WebSocket-Version: 13\r\n/
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6\r\n
Sec-WebSocket-Key: lGaahHe/KdA9lPdPxAlZfw==\r\n
Sec-WebSocket-Extensions: permmessage-deflate; client_max_window_bits\r\n
\r\n
```

---

注：X-Cisco-CMS-Auth-Token の値は、「authToken」URI パラメータとしてミーティングサーバーに送信する必要があります。Sec-WebSocket-Key の値（およびフォロアアップの Sec-WebSocket-Accept）の値は、[RFC6455 に準拠しています](#)。

---

- Meeting Server が WebSocket 接続を受け入れる場合（例えば、authToken が有効）、成功時の応答は次のようになります。

```
HTTP/1.1 101 プロトコルの切り替え\r\n
```

```
アップグレード: websocket\r\n
```

```
接続: アップグレード\r\n
```

```
Sec-WebSocket-Accept: ZISmDfOsp675RM7TQKa0LbQKCqk=\r\n
```

```
\r\n
```

5. 双方向 WebSocket 接続は「使用可能」で、イベントメッセージのフローを開始できます。

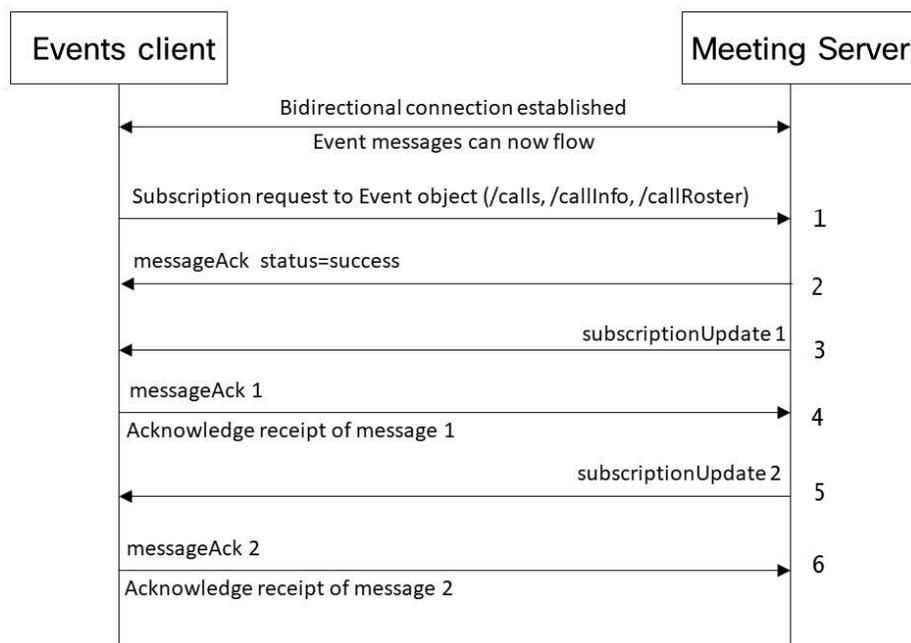
## 5 メッセージフローの例

### 5.0.1 メッセージフローの例

イベントクライアントと Meeting Server 間の WebSocket が確立されると、イベントクライアントは Meeting Server に登録して、イベントタイプの更新を取得できます。

図 2 は、ウェブソケット接続を介したイベントクライアントとミーティングサーバ間のコールフローを示しています。イベントクライアントからのサブスクリプションは、JSON ファイルの形式である必要があります。通話フローの詳細については、図 2 を参照してください。

図 2: イベントクライアントとミーティングサーバ間を流れるイベントメッセージの概要



1. イベントクライアントから Meeting Server への最初のサブスクリプション。コールリスト（アクティブな電話会議リスト）などのイベントタイプのサブスクリプションをリクエストします。

```

{
  "type": "message",
  "message": {
    "messageId": 8,
    "type": "subscribeRequest",
    "subscriptions": [
      {
        "index": 3,
        "type": "calls",
        "elements": [
          "name",
          "participants"
        ]
      }
    ]
  }
}

```

上記の例では、クライアントはインデックス 8 のメッセージングとしてサブスクライブリクエストを送信し、サブスクリプションと呼ばれるセット（配列）をサブスクライブするための単一のリソースを提供しています。このサブスクリプションに対してタグ「3」のインデックスを使用しました。タイプは「calls」で、これはアクティブな通話リストを指します。このサブスクリプション内で、関心のある要素「name」（電話会議名）および「participants」（アクティブな参加者数）のセットが指定されています。この要素のセットには、2 つの主な影響があります。

- どの変更がミーティングサーバ→クライアントの更新をトリガーするかを決定します（この場合、名前または参加者数に変更されると更新が送信されます）。
- Meeting Server からクライアントへの更新にどの要素を含めるかを決定します（この場合、クライアントに送信されるレコードには、名前と参加者数だけが含まれます）。

2. Meeting Server は、サブスクライブリクエストに対して ACK で応答します。

```

{
  "type": "messageAck",
  "messageAck": {
    "messageId": 8,
    "status": "success"
  }
}

```

この ACK は「messageId」が 8 で、どのメッセージが確認応答されているかをクライアントに示すために、およびメッセージが正常に理解され、実行されたかどうかを示す「ステータス」コードがあります。

3. Meeting Server はまた、アクティブなサブスクリプションの状況に関して、リクエストを送信したクライアントに最新情報を提供します。これは、例えば次のように、クライアントに送信される `subscriptionUpdate` の形式になります。

```
{
  "type": "message",
  "message": {
    "messageId": 1,
    "type": "subscriptionUpdate",
    "subscriptions": [
      {
        "index": 3,
        "state": "pending"
      }
    ]
  }
}
```

これは、インデックス「3」のサブスクリプションが保留中であることをクライアントに伝えます。セットアップの処理中ですが、まだアクティブではありません。Meeting Server はこのメッセージに 1 の `messageId` のタグを付けています。Meeting Server が後の更新を送信するために、クライアントはこのメッセージに対して ACK 応答する必要があります。

4. クライアントからミーティングサーバへの対応する `messageAck` は以下のようになります。

```
{
  "type": "messageAck",
  "messageAck": {
    "messageId": 1,
    "status": "success"
  }
}
```

Meeting Server とクライアントは、`messageId` 値に独自の個別の番号スペースを使用します。それらが異なる必要はなく、連続する必要もありません。

5. Meeting Server がリクエストされたサブスクリプションのセットアップに成功すると、さらに次の `subscriptionUpdate` を送信する場合があります。例：

```

{
  "type": "メッセージ",
  "メッセージ": {
    "messageId": 2,
    "type": "subscriptionUpdate",
    "subscriptions": [
      {
        "index": 3,
        "状態": "アクティブ"
      }
    ]
  }
}

```

これは、サブスクリプション（インデックス 3）がアクティブになったことがクライアントに通知され、クライアントはサブスクリプション固有の更新を受信できるようになります。前と同じように、ミーティングサーバからのこのメッセージはクライアントによって承認される必要があります。

6. クライアントは、messageId 2 のサブスクリプションの更新を確認します。クライアントは、確認で messageId 2 を提供します。これは、サブスクリプション更新で Meeting Server によって使用される値であるためです。

```

{
  "type": "messageAck",
  "messageAck": {
    "messageId": 2,
    "status": "success"
  }
}

```

7. クライアントが通話リストをサブスクライブした後のある時点で、Meeting Server は開始したばかりの電話会議に関する情報を送信します。

```

{
  "type": "メッセージ", "メッ
  セージ": {
    "messageId": 3,
    "type": "callListUpdate",
    "subscriptionIndex": 3,
    "updates": [
      {
        "call": "97c771ae-fc2e-4257-b129-30ee818e034b",
        "updateType": "add",

```

```

        "name": "Andy's coSpace",
        "participants": 0
      }
    ]
  }
}

```

前と同様に、このメッセージ messageId 3 はクライアントによって確認される必要があります。

---

**注：**重複を避けるため、この例では messageAcks が表示されなくなりました。

---

この更新には、更新が参照するサブスクリプションをクライアントに示す subscriptionIndex "3" のタグが付けられます。この更新には、データの解析を支援する「タイプ」が含まれます。「updates」配列には、Meeting Server が提供している新しい情報が含まれます。この例では、「updateType」は「add」です。これは、これがこの電話会議の最初の通知であることを意味し、問題の電話会議は「call」フィールドで GUID が提供されています。

「call」および「updateType」の値はすべての更新で表示されますが、残りのフィールドは、クライアントからのサブスクリプション要求で提供された「elements」の値によって決定されます。「elements」ノードが提供されない場合（またはそれが空である場合）、追加のフィールドは含まれません。ただし、「name」と「participants」は例に含まれていたため、更新にはこれらが含まれています。

- この会議の最初の「追加」メッセージの後、参加者が会議に参加すると、Meeting Server からさらに更新が受信されます。

```

{
  "type": "message",
  "message": {
    "messageId": 4,
    "type": "callListUpdate",
    "subscriptionIndex": 3,
    "updates": [
      {
        "call": "97c771ae-fc2e-4257-b129-30ee818e034b",
        "updateType": "update",
        "参加者": 1
      }
    ]
  }
}

```

参加者数が「1」になり、「updateType」が「update」になりました。これは、問題の通話（97c771ae-fc2e-4257-b129-30ee818e034b）に対する最初のメッセージではなく、前の通知の更新を示しています。この呼び出しの「名前」の値は変更されていないため、この更新には含まれていません。

9. クライアントアプリケーションが（イベントメカニズム、API クエリ、または「callStart」CDR のいずれかにより）関心のあるアクティブな通話の存在を知ると、クライアントアプリケーションはその通話に固有のリソースをサブスクライブする場合があります。例えば、次のような新しいメッセージで Meeting Server のサブスクリプションを再設定することがあります。

```
{
  "type": "message",
  "message": {
    "messageId": 9,
    "type": "subscribeRequest",
    "subscriptions": [
      {
        "index": 1,
        "type": "callRoster",
        "call": "97c771ae-fc2e-4257-b129-30ee818e034b",
        "elements": [
          "名前",
          "uri",
          "state",
          "importance"
        ]
      },
      {
        "インデックス": 2,
        "type": "callInfo",
        "call": "97c771ae-fc2e-4257-b129-30ee818e034b",
        "elements": [
          "name",
          "participants",
          "streaming"
        ]
      },
      {
        "index": 3,
        "type": "calls",
        "elements": [
```

```

        "name",
        "participants"
    ]
}
]
}
}
}

```

クライアントは、その「サブスクリプション」のセットで同じ「インデックス」3を持つサブスクリプションを維持することで、「コール」（アクティブな電話会議リスト）をサブスクライブしたままにします。しかし、現在は「callRoster」と「callInfo」のサブスクリプションをセットに追加しています。これらのサブスクリプションについては、特定の「通話」GUIDを提供する必要があります。この場合は、「97c771ae-fc2e-4257-b129-30ee818e034b」、クライアントはミーティングサーバから以前の "callListUpdate" メッセージで通知されていました。

## 6 WebSocket の仕様

このセクションでは、WebSocket を使用したイベントクライアントとミーティングサーバ間の接続の確立について詳しく説明します。以下で指定された制限を超えないでください。

- ミーティングサーバあたりの WebSocket 同時接続の最大数: 5
- WebSocket 接続あたりの同時サブスクリプションの最大数: 100
- 断片化された WebSocket フレームはサポートされていません。
- ping または pong コントロールフレーム、またはバイナリデータフレームはサポートしていません。

## Cisco の法的情報

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

Cisco が採用している TCP ヘッダー圧縮機能は、UNIX オペレーティング システムの UCB (University of California, Berkeley) のパブリック ドメイン バージョンとして、UCB が開発したプログラムを採用したものです。All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコシステムズおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコシステムズまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

★定型★このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。★定型★マニュアル内の例、コマンド出力、ネットワークポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

この文書の印刷されたハード コピーおよび複製されたソフト コピーは、すべて管理対象外と見なされます。最新版については、現在のオンライン バージョンを参照してください。

シスコは世界各国 200 箇所にオフィスを開設しています。各オフィスの住所と電話番号は、当社の Web サイト [www.cisco.com/go/offices](http://www.cisco.com/go/offices) をご覧ください。

© 2024 Cisco Systems, Inc. All rights reserved.

## Cisco の商標または登録商標

Cisco および Cisco ロゴは、シスコまたはその関連会社の米国およびその他の国における商標または登録商標です。シスコの商標の一覧については、[www.cisco.com/jp/go/trademarks](http://www.cisco.com/jp/go/trademarks) をご覧ください。記載されているサードパーティの商標は、それぞれの所有者に帰属します。

「パートナー」という用語の使用はシスコと他社との間のパートナーシップ関係を意味するものではありません。 (1721R)