

Live call modification API

Live call modification allows you to interrupt an in-progress call and terminate it or have it begin processing RCML from a new URL. This is useful for any application where you want to asynchronously change the behavior of a running call. For example: hold music, call queues, transferring calls, forcing hangup, etc.

HTTP POST to a Call

To redirect or terminate a live call, you make an HTTP POST request to an in-progress Call instance resource URI:

/2012-04-24/Accounts/{AccountSid}/Calls/{CallSid}

or

/2012-04-24/Accounts/{AccountSid}/Calls.json/{CallSid}

POST Parameters

The following parameters are available for you to POST when modifying a phone call:

Parameter	Description
Url	A valid URL that returns RCML. Restcomm will immediately redirect the call to the new RCML.
Method	The HTTP method Restcomm should use when requesting the above URL. Defaults to POST.
Status	Either canceled or completed . Specifying canceled will attempt to hangup calls that are queued or ringing but not affect calls already in progress. Specifying completed will attempt to hang up a call even if it's already in progress.

Note that any call which is currently ringing within a **<Dial>** verb is in-progress from the point of view of Restcomm, and thus you must use 'Status=completed' to cancel it.

Optional Parameters

You may POST the following parameters:

Parameter	Description
FallbackUrl	A URL that Restcomm will request if an error occurs requesting or executing the RCML at Url.
FallbackMethod	The HTTP method that Restcomm should use to request the FallbackUrl. Must be either GET or POST. Defaults to POST.

StatusCallback	A URL that Restcomm will request when the call ends to notify your app.
StatusCallbackMethod	The HTTP method Restcomm should use when requesting the above URL. Defaults to POST.
MoveConnectedCallLeg	If True, Restcomm will move both call legs to the new url

Description

Prerequisite for the Live Call Modification API is to know the **CallSid** of the call. The **CallSid** is generated by Restcomm for every incoming or outgoing call.

When we create an outgoing call using the Calls REST API, Restcomm will generate the **CallSid** and the response will contain the **CallSid**.

For an Incoming call, Restcomm will generate the **CallSid**, and will include it in the parameters of the GET or POST request to download the RCML from the application server.

So the application server, will get the **CallSid** for this incoming call and can store it for later use.

So the steps for an incoming call are the following:

1. New incoming call to Restcomm
2. Restcomm generates CallSid and other parameters for this call
3. Restcomm prepares GET/POST request and attaches the previously prepared parameters
4. Restcomm sends the GET/POST request to the URL assigned to the DID called
5. Application server receives the GET/POST request
6. Application server should store the CallSid along with the rest of the parameters for later use
7. Application server prepares the response with the appropriate RCML for the given DID and send it back to Restcomm to process it

Given that we have the **CallSid** available, the next step to modify a live call is to prepare the new POST request to Restcomm with the URL of the new destination application.

Lets assume that initially the Application server prepared an RCML that will connect the incoming call to Bob and that later Bob wants this call to be transferred to Alice.

Application Server will have to prepare a new RCML that will dial to Alice, and using the Live Call Modification API, will ask Restcomm to process this new RCML for the incoming call and thus will connect it to Alice.

Here are the steps:

1. Incoming call - CallSid 1234567890
2. Application server:
 - a. Store CallSid 1234567890 and other parameters
 - b. Prepare and send RCML that connects the call to Bob (using Dial verb)
3. Restcomm process the given RCML and connects incoming call to Bob
4. Bob wants to transfer the call to Alice and sends the request to Controller (application server)
5. Application server:
 - a. Prepares new RCML with a given URL: <http://app.server/CallToAlice> that using Dial will dial to Alice
 - b. Uses Live Call Modification prepares a new POST request:
 - i. http://RESTCOMM_IP:8080/restcomm/2012-04-24/Accounts/{AccountSid}/Calls/1234567890
 - ii. in the POST requests adds the following parameters:
 1. Url=<http://app.server/CallToAlice>
6. Restcomm process this Live Call Modification request and redirects the incoming call to the new RCML application (to the new URL)
7. Eventually the incoming call will be connected to Alice.

Similar, application server can change an incoming call to Bob to a conference call that we can add more participants later:

Given that application server already knows the CallSid for the incoming call to Bob

1. Will prepare a new RCML for the Dial Conference
2. Will prepare new POST Live Call Modification request for the incoming call to be connected to the new url (that will dial to the conference)
3. Will then prepare new outgoing call request, using the Calls Rest API that will connect Bob to the conference application url:
 - a. http://RESTCOMM_IP:8080/restcomm/2012-04-24/Accounts/{AccountSid}/Calls
 - b. and will pass the following parameters:
 - i. From=INCOMING_CALL_ID
 - ii. To=client:bob
 - iii. Url=CONFERENCE_APPLICATION_URL
4. Later application server can create more outgoing call requests, similar to the one above, to invite other clients or sip uris or pstn numbers to this conference call

Examples

Redirect Call to a new RCML

Live Call Modification POST request URL :

<http://127.0.0.1:8080/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json/CAd62f17e7c7f149ac8095f7574495d988>

Parameters:

Url=<http://127.0.0.1:8080/restcomm/dial-alice.xml>

curl -X POST

```
http://ACae6e420f425248d6a26948c17a9e2acf:dcd25094354d425c8ab85b6621083d20@192.168.1.151:8080/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json/CAd62f17e7c7f149ac8095f7574495d988 -d
"Url=http://192.168.1.151:8080/restcomm/demos/dial-alice.xml"
```

Redirect Call to a new RCML and connect both call legs

Live Call Modification POST request URL :

<http://127.0.0.1:8080/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json/CAd62f17e7c7f149ac8095f7574495d988>

Parameters:

Url=<http://127.0.0.1:8080/restcomm/conference.xml>

curl -X POST

```
http://ACae6e420f425248d6a26948c17a9e2acf:dcd25094354d425c8ab85b6621083d20@192.168.1.151:8080/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json/CAd62f17e7c7f149ac8095f7574495d988 -d
"Url=http://192.168.1.151:8080/restcomm/demos/conference.xml" -d
"MoveConnectedCallLeg=true"
```

Terminate In Progress call

Live Call Modification POST request URL :

<http://127.0.0.1:8080/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json/CAe29bf33b54ba43dc952f0d3eb331567c>

Parameters:

Status=completed

Terminate Ringing call

Live Call Modification POST request URL :

<http://127.0.0.1:8080/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json/CA312824ebd262419dbd4c00b057448d57>

Parameters:

Status=canceled