

# Call Control Web Service

## Overview

The *SipXcallController* is a RESTful service for third party call control. It is bundled as a jar file and is loaded by the *sipXrest* container on initialization. It is invoked by HTTP Post to a specific URL.

The *SipXcallController* implements a third party call controller. The third party call controller works by either sending an INVITE to the calling party and subsequently sending that calling party a REFER to transfer the call to the called party (fire and forget) or by staying in the call path using the call flow 4 of RFC 3725.

The latter method allows subsequent control of the call such as being able to transfer the call after call setup.

The call controller tracks the state of the call in progress for a specified period of time after the call is placed and provides this status for query by applications.

The security model for these operations is enforced by the sipXrest container. That is, when the operation is invoked from the trusted domain using HTTPS, no pin is required. Otherwise, a pin must be supplied for the operation to succeed.

The service is initiated using HTTPS using HTTPS post and queried using HTTPS get. HTTP is also supported with Digest authentication but it is not the preferred method of service invocation.

## Services

If you are issuing the curl command from a trusted host ( i.e. one where sipx proxy is running) please use only HTTPS (no authentication is required). From a non-trusted host you must authenticate the curl command with Digest authentication.

There are some examples with notes in the source directory. Download the source code from svn and take a look at the scripts in sipXcallController/test

6666 is the HTTPS port.

6667 is the HTTP port.

You can use either HTTPS or HTTP when issuing the curl command from a non-trusted host.

Since version 4.6 you should use the redirector in sipxconfig as using port 6666 seems deprecated. The REST interface can be reached (using digest authentication) on `http(s)://<host>/sipxconfig/rest/my/redirect/callcontroller` etc.

The URI is as documented below:

URI	URL Parameters (Optional)	Methods	Formats
<code>callcontroller/ {callingUser}/ {calledUser}</code>	<p>sipMethod=[REFER INVITE] the default is REFER</p> <p>action=[call transfer] the default action is call</p> <p>target=&lt;targetUser&gt; the target user for the call transfer. If an ongoing call exists, the call is transferred to the targetUser. This is meaningful only if the call was previously started using INVITE.</p> <p>timeout=&lt;seconds&gt; - the time for which the calling party is alerted before pickup. If the calling party does not pickup in that time period, the call is aborted.</p> <p>agent=&lt;agentUser&gt; the user name of the agent that is placing the third party call. This defaults to the callingUser</p> <p>subject=&lt;subject&gt; Subject of the call. Determines what goes into the SIP subject header of the initial call setup INVITE.</p> <p>isForwardingAllowed=[true false] default is false whether or not forwarding is allowed for the initial INVITE</p> <p>resultCacheTime=&lt;seconds&gt; The amount of time for which to keep the call progress cache record in memory.</p>	<b>POST</b>	Initiates a call from the callingUser to the calledUser
<code>callcontroller/ {callingUser}/ {calledUser}</code>	All URL parameters are ignored. Note that you can only get the current call state if you have previously initiated the call from callingUser to calledUser	<b>GET</b>	Get the current call state for any ongoing call between callingUser and calledUser

## Examples

Using curl, you would initiate a call from callingUser to calledUser from a remote host as follows:

```
curl -k -X POST -u {callingUser}:{pin} https://<host>:6666/callcontroller/{callingUser}/{calledUser}?
timeout=<seconds>
```

Or as an agent (third party)

```
curl -k -X POST -u {agent}:{agentPin} https://<host>:6666/callcontroller/<callingUser>/<calleedUser>?
agent=<agentName>
```

The following example is to initiate a call between user1 and user2 as user1 who is the calling party.

```
curl -k -X POST -u user1:123 https://sipxtest.sipxtest.net:6666/callcontroller/user1/user2
```

An example to initiate a call between user1 and user2 as user3 who is a third party agent.

```
curl -k -X POST -u user3:123 https://sipxtest.sipxtest.net:6666/callcontroller/user1/user2?agent=user3
```

On success, HTTP 200 OK returned. no message body.

On Failure, corresponding HTTP status code returned and more plain/text details contained in the message body, for example. "not valid user pin".

Similar to "place a call", the query can be made either from calling party. The URL is exactly the same as that you would use when placing the call but the method is GET.

Here is an example of how to query call setup progress for call between user1 and user2 as user1.

```
curl -k -u user1:123 https://sipxtest.sipxtest.net:6666/callcontroller/user1/user2
```

Here is an example of how to query call setup progress for call between user1 and user2 via agent user3 from sipX trusted domain, no password required.

```
curl -k https://sipxtest.sipxtest.net:6666/callcontroller/user1/user2?agent=user3
```

Here is an example of how to query call setup progress for call between user1 and user2 via agent user3 from sipX's non-trusted domain.

```
curl -k -u user3:123 -X GET https://sipxtest.sipxtest.net:6666/callcontroller/user1/user2?agent=user3
```

## Return result from the query above

If the transfer is invoked using REFER, the calling party records the NOTIFY bodies and reports it (uninterpreted) when the user issues an HTTP GET to fetch the call status. This is reported to the caller when a GET is performed as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<status-lines xmlns="http://www.sipfoundry.org/sipX/schema/xml/call-status-00-
00
">
<status>
  <timestamp>1259114103662</timestamp>
  <call-id>7959e447b88502416ee2477f3d36a480@12.23.34.45</call-id>
  <method>INVITE</method>
  <status-line>SIP/2.0 100 Trying</status-line>
</status>
<status>
  <timestamp>1259114103667</timestamp>
  <call-id>7959e447b88502416ee2477f3d36a480@12.23.34.45</call-id>
  <method>INVITE</method>
  <status-line>SIP/2.0 407 Proxy Authentication Required</status-line>
</status>
<status>
  <timestamp>1259114103678</timestamp>
  <call-id>7959e447b88502416ee2477f3d36a480@12.23.34.45</call-id>
  <method>INVITE</method>
  <status-line>SIP/2.0 100 Trying</status-line>
</status>
<status>
  <timestamp>1259114103841</timestamp>
  <call-id>7959e447b88502416ee2477f3d36a480@12.23.34.45</call-id>
  <method>INVITE</method>
  <status-line>SIP/2.0 180 Ringing</status-line>
</status>
</status-lines>
```