Call Control

The SipXcallController is a RESTful service for third party call control. It is bundled as a JAR package file format and is loaded by the sipXrest container on initialization. It is invoked by the HTTP methods GET and POST to a specific URL.

The sipXrest service can run on any node in the cluster, but this is transparent for the end-user because end-user will always make the REST call to sipXconfig (admin node). SipXconfig is the only container that knows where sipXrest is running, and will route the call accordingly.

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.

When the operation is invoked from a node inside the cluster, no pin is required, and no certificate (HTTP protocol is used). For communication inside the cluster data encryption is optional and configurable from UI. Otherwise, a pin must be supplied for the operation to succeed and also a certificate must be accepted (HTTPS protocol is used). The security model for these operations is enforced by the sipXconfig container, that contains the most complex security layer including BASIC or DIGEST authentication schemes, LDAP authentication, HTTP secured (HTTPS). On the back scenes we proxy call controller calls through sipXconfig, and then sipXconfig will route the call on the host where sipXrest service is running.

Services

If you are issuing the cURL command from a node inside cluster please use only HTTP (no authentication is required) and the hostname is the node where sipXrest is running

From outside cluster you must authenticate the curl command with Basic, Digest authentication or perform LDAP authentication

You can use only HTTPS when issuing the curl command from outside the cluster

For calls issued from outside the cluster, only sipXconfig (admin) port and hostname are used (port 443, the standard HTTPS port)

The URI is as follows: */my/redirect<callcontroller URI>*

Query call progress

Resource URI: callcontroller/{callingUser}/{calledUser}
Default Resource Properties: N/A
URL Parameters:

Property	Description
sipMethod	The SIP request method used. Available methods are: • **Refer** - Asks recipient to issue SIP request (call transfer.) • **Invite** - Indicates a client is being invited to participate in a call session. Default is **Refer**.
action	The action to be performed. Options are • **Call** – Place a call • **Transfer** – Transfer a call Default is **Call**.
target	The target user for the call transfer. If an ongoing call exists, the call is transferred to the target user. This is meaningful only if the call was previously started using the **Invite** SIP method.
timeout	The time measured in seconds for which the calling party is alerted before pickup. If the calling party does not pickup in that time period, the call is aborted.

agent	The user name of the agent that is placing the third party call. This defaults to the calling user.
subject	The subject of the call. Determines what goes into the SIP subject header of the initial call setup invite.
isForwardingAllowed	Determines is forwarding is allowed. Values are • **True** • **False** Default is false whether or not forwarding is allowed for the initial invite.
resultCacheTime	The amount of time measured in seconds progress cache record are kept in memory.

Status Values: N/A
Specific Response Codes:
HTTP Method: POST
Initiates a call from the *callingUser* to the *calledUser*
Return Values: N/A
Unsupported HTTP Method: GET, PUT, DELETE

View ongoing calls

Resource URI: callcontroller/{callingUser}/{calledUser}
Default Resource Properties: N/A
URL Parameters: All URL parameters are ignored. Note that you an only get the current call state of you have previously initiated the call from *callingUser* to *calledUser.*
Status Values: N/A
Specific Response Codes:
HTTP Method: GET
Get the current call state for any ongoing call between *callingUser* and *calledUser.*
Return Values: N/A
Unsupported HTTP Method: POST, PUT, DELETE

Examples

Initiating a call from a remote host

From inside cluster:
curl -k -X POST http://<callcontroller_node_host>:6667/callcontroller/{callingUser}/{calledUser}?timeout=<seconds>
From outside cluster:
curl -k -X POST -u {callingUser}:{pin} https://<web_admin_host>/sipxconfig/rest/my/redirect/callcontroller/{callingUser}/{calledUser}?timeout=<seconds>

Initiating a call as third party

From inside cluster:
curl -k -X POST http://<callcontroller_node_host>:6667/callcontroller/<callingUser>/<calleedUser>?agent=<agentName>
From outside cluster:
curl -k -X POST -u {agent}:{agentPin} https://<web_admin_host>/sipxconfig/rest/my/redirect/callcontroller/<callingUser>/<calleedUser>?agent=<agentName>

Initiating a call between *user1* and *user2* when user1 is the calling party

From inside cluster:
curl -k -X POST http://sipxtest.sipxtest.net:6667/callcontroller/user1/user2
From outside cluster:
curl -k -X POST -u user1:123 https://sipxtest.sipxtest.net/sipxconfig/rest/my/redirect/callcontroller/user1/user2

Initiating a call between *user1* and *user2* when *user3* is a third party agent.

From inside cluster:
curl -k -X POST http://sipxtest.sipxtest.net:6667/callcontroller/user1/user2?agent=user3
From outside cluster:
curl -k -X POST -u user3:123 https://sipxtest.sipxtest.net/sipxconfig/rest/my/redirect/callcontroller/user1/user2?agent=user3
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.
From inside cluster:
curl -k http://sipxtest.sipxtest.net:6667/callcontroller/user1/user2
From outside cluster:
curl -k -u user1:123 https://sipxtest.sipxtest.net/sipxconfig/rest/my/redirect/callcontroller/user1/user2

Querying call progress setup

Here is an example of how to query call setup progress for call between user1 and user2 via agent user3 from the internal cluster, with no password required:
curl -k http://sipxtest.sipxtest.net:6667/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 the openUC external cluster:
curl -k -u user3:123 -X GET https://sipxtest.sipxtest.net/sipxconfig/rest/my/redirect/callcontroller/user1/user2?agent=user3
Return result from the query above
If the transfer is invoked using the **Refer** SIP method, the calling party records the **Notify** bodies and reports it (interpreted) when the user issues an HTTP GET to retrieve the call status. This is reported to the caller when the GET method is performed as shown below:

```
<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>
```