

# ClickToCall Sample Snap-in

## Introduction

The ClickToCall sample snap-in demonstrates the two-party make call feature. The Avaya Breeze® platform calls both parties during the two-party make call operation.

The basic make-call operation is as follows:

- 1.The Avaya Breeze calls the first party.
- 2.When the first party answers, the Avaya Breeze calls the second party.
3. The first party will hear ring back tone until the second party answers.

The sample snap-in provides a simple web page that allows the user to enter the first party, the second party number, the calling identity, the calling display, and then make a call. The calling Identity in the 2-party make call helps the first party understand the context of the call, since there is no caller on the other end at this point. The calling Identity is who the call appears to be from when ringing the first party and when ringing the second party. Not all phones will display the calling identity, so an optional field calling display is also available and allows the user to override their own calling display. This sample snap-in provides a REST interface for the user interface (UI) to interact with the backend server code.

## Overview

Once the snap-in has been installed on the Avaya Breeze, the user can navigate to the page using the following URL to interact the application:

<https://<Avaya Breeze Asset IP>/services/ClickToCall/index.jsp>

The user will see four text fields for the first party, second party , the calling identity and the calling display. When the user clicks the ClickToCall button, the snap-in initiates the two-party make call operation. If the request is accepted, Avaya Breeze will make a call to the second party.

2

When the first party answers, the Avaya Breeze makes a call to the second party. The first party will hear a ring back tone until the second party answers. The ring back tone is generated by the Avaya Media Server.

## Concepts Demonstrated

- Two-party make call feature using the Collaboration API's `IdentityFactory` and `CallFactory` methods to create a call.
- Initiating a call using Collaboration API's `Call` method `initiate`.

## Detailed description

The web page allows the user to enter the first party, second party, the calling identity and the calling display. Upon clicking the ClickToCall button, RESTful `POST` request is handled by the `doPost()` method of the `TwoPartyCallManagerResource` class. The body of the `POST` request contains the first, second and calling identity numbers:  
`first=3403@avaya.com&second=3404@avaya.com&calling=3405@avaya.com`

These numbers are extracted by the `RequestProcessImpl` class that through methods checks if the domain has been supplied or not. If no domain is given, then the Default SIP Domain cluster attribute from the cluster is used.

Next, the first and second party numbers, the calling identity as well as the calling display are passed to the `ClickToCallImpl` class. This class calls the `create()` method belonging to the `IdentityFactory` Class that creates an identity that represents the calling identity and calling display. The first and second party numbers and the newly created identity are then passed to `create()` method of the `CallFactory` class that creates a new instance of a `Call`. The outgoing call is then initiated using Collaboration API `call.initiate()`.

3

The first, second and calling identity numbers can be a simple telephone number, either E.164 or any digit string that matches your system's dial plan. The provided number can optionally include a domain (number@domain). If no domain is supplied, the Default SIP Domain provisioned for the Cluster will be used.

As an example, consider the case of an enterprise user initiating a call from a mobile or home phone. In

this case, 3 numbers would come into play:

- The user's Avaya Aura® number would be used as the Identity
- The user's home/mobile number would be the first party number
- The destination number would be the second party number

This would allow the first party and other applications to see the call as being from the user's Avaya Aura® number rather than from the user's home/mobile number.

## Installation and Configuration

The ClickToCall snap-in is located in the SDK zip at: `/samples/ClickToCall`.

Change the directory to where the snap-in resides and compile it: `mvn clean install`.

For information on installing the snap-in, see the [\[DEVC: Installing, Configuring and Testing an Avaya Breeze Service guide\]](#).

## Example usage of the Snap-in

Assuming three users: first party: 3403@avaya.com, second party:3404@avaya.com, calling identity:3405@avaya.com, calling display "user1 display"

1. Enter the four numbers in the text fields provided on the index.jsp web page.
2. Click the ClickToCall button.
3. 3403 will be called first.
4. Calling identity 3405 and user1 display appears on the display of 3403.
5. When 3403 answers the call, 3404 will be called.
6. Calling identity 3405 and “user1 display” appears on the display of 3404 as well.
7. 3403 will hear a ring back tone until 3404 answers.
8. Text area will provide a status message.

## **POSTMAN (HTTP REQUEST GENERATOR)**

Alternatively, user can post the REST http from a tool like postman to initiate a call (instead of using the UI).

This would involve sending a POST with a body in following format:

[first=3403@avaya.com&second=3404@avaya.com&calling=3405@avaya.com](https://<Avaya Breeze Asset IP>/services/ClickToCall/twoPartyCallManager?first=3403@avaya.com&second=3404@avaya.com&calling=3405@avaya.com) to the URL:

[https://< Avaya Breeze Asset IP >/services/ClickToCall/twoPartyCallManager](https://<Avaya Breeze Asset IP>/services/ClickToCall/twoPartyCallManager)

If the request is accepted, Avaya Breeze will make a call to the second party. When the first party answers, the Avaya Breeze makes a call to the second party. The first party will hear a ring back tone until the second party answers. The ring back tone is generated by the Avaya Media Server.

The sample is also bundled with POSTMAN environment/collection setting files that can be readily imported in POSTMAN. The purpose of the Postman Environment is to create default HTTP request URL encoded parameters (default domains, host addresses, etc.) that can be substituted in the Postman Collection. In the Postman Environment, set the host field to the IP address of the Security Module on Avaya Breeze you will be sending your HTTP request to. Do not enter the FQDN or IP address of the Network Management interface of Avaya Breeze! The servicename and servicepath should be left to their defaults. Change the firstNumber and secondNumber fields to their desired values.

In order to use the Postman Environment defaults (i.e. host, servicename, etc.) in the Postman Collection use double braces surrounding the field name. (i.e. `{{host}}/services/{{servicename}}/{{servicepath}}`, etc.). When editing or viewing the Postman Collection, please use the “x-www-form-urlencoded” button to view the data sent in the HTTP request. Leave the line containing the URL to `{{host}}/services/{{servicename}}/{{servicepath}}` unmodified. These are the values used from the Postman Environment.