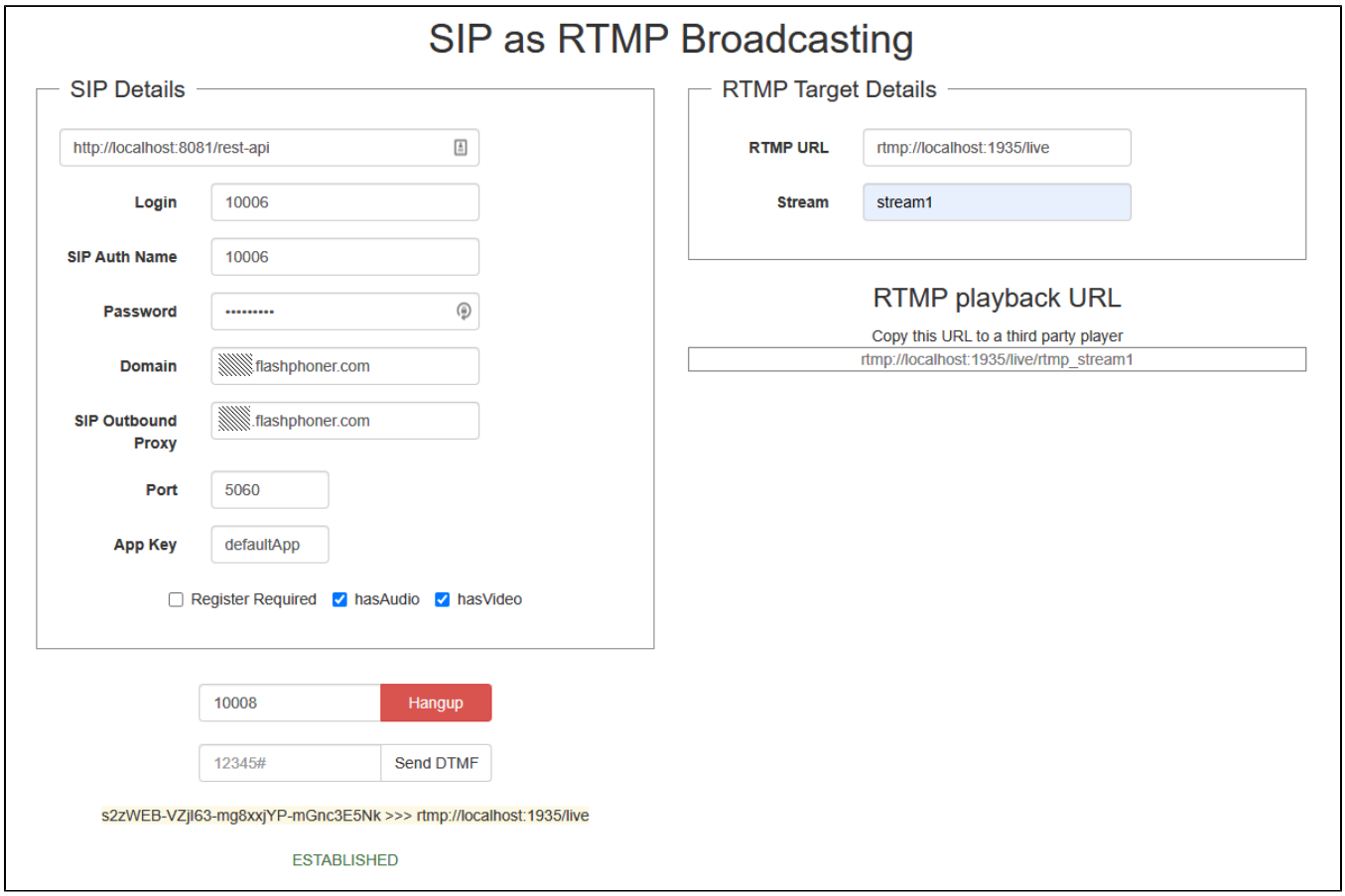
# SIP as RTMP



## Example of delivery of a video stream from SIP to RTMP server

This example demonstrates how to make a call to SIP, receive audio and video traffic from SIP in response and then redirect the received video stream to a third-party RTMP server or to a built into the WCS RTMP server, or to a third-party CDN accepting RTMP streams for further broadcasting.




The screenshot illustrates the following:

1. In the left part, we fill in the data of the SIP account we will be using for the call. If the SIP server requires no authorization, you can specify arbitrary login and password, like 'abcd'.
2. In the right part, we enter the RTMP address of the server and the name of the video stream. This address will accept the redirected SIP traffic as soon as connection is successfully established.
3. Call the SIP subscriber '10050'. Below, we have an option to send a DTMF signal if the SIP side implements a voice menu. After the connection to SIP is successfully established, the ESTABLISHED status is displayed.
4. RTMP playback URL may be copied to a third party player (ffplay, VLC etc)

## Code of the example

This example is a simple REST client written on JavaScript, available at:

*/usr/local/FlashphonerWebCallServer/client/examples/demo/sip/sip-as-rtmp.html*

sip-as-rtmp.js - a script dealing with REST calls to the WCS server
sip-as-rtmp.html - example page

## Analyzing the code

To examine the code, let's take this version of the sip-as-rtmp.js with the hash ecbadc3 which is available here and can be downloaded with the corresponding build 2.0.212.

1. REST / HTTP queries sending.

code

Sending is done using POST method with ContentType application/json by AJAX query using Jquery framework.

```
function sendREST(url, data) {
    console.info("url: " + url);
    console.info("data: " + data);
    $.ajax({
        url: url,
        beforeSend: function ( xhr ) {
            xhr.overrideMimeType( "text/plain;" );
        },
        type: 'POST',
        contentType: 'application/json',
        data: data,
        success: handleAjaxSuccess,
        error: handleAjaxError
    });
}
```

2. Making outgoing call with REST-request /call/startup

code

Connection data (connection) to establish connection and call data (RESTCall) are collected from the boxes on page.

```
    var url = field("restUrl") + "/call";
    callId = generateCallID();

    var connection = {};
    connection.sipLogin = field("sipLogin");
    connection.sipPassword = field("sipPassword");
    connection.sipPort = field("sipPort");
    connection.sipDomain = field("sipDomain");
    connection.appKey = field("appKey");
    connection.sipRegisterRequired = field("sipRegisterRequired");

    for (var key in connection) {
        setCookie(key, connection[key]);
    }

    var RESTCall = {};
    RESTCall.rtmpStream = field("rtmpStream");
    RESTCall.hasAudio = field("hasAudio");
    RESTCall.hasVideo = field("hasVideo");
    RESTCall.callId = callId;
    RESTCall.rtmpUrl = field("rtmpUrl");

    for (var key in RESTCall) {
        setCookie(key, RESTCall[key]);
    }

    RESTCall.connection = connection;
    RESTCall.callee = field("callee");

    var data = JSON.stringify(RESTCall);

    sendREST(url, data);
    startCheckStatus();
    sendDataToPlayer();
```

3. The status of the SIP call is monitored with the /getStatus query

code

```
function getStatus() {
    var url = field("restUrl") + "/getStatus";
    var currentCallId = { callId: callId };
    $("#callTrace").text(callId + " >>> " + field("rtmpUrl"));
    var data = JSON.stringify(currentCallId);
    sendREST(url, data);
}
```

4. If the PBX has a voice menu, the DTMF signal can be sent using the /sendDTMF query.

code

```
function sendDTMF(value) {
    var url = field("restUrl") + "/sendDTMF";
    var data = {};
    data.callId = callId;
    data.dtmf = value;
    data.type = "RFC2833";
    data = JSON.stringify(data);
    sendREST(url, data);
}
```

5. RTMP URL displaying on the page to copy to a third party player

code

```
function sendDataToPlayer() {
    var host = field("rtmpUrl")
        .replace("localhost", window.location.hostname)
        .replace("127.0.0.1", window.location.hostname);

    var rtmpStreamPrefix = "rtmp_";
    var url = host + "/" + rtmpStreamPrefix + field("rtmpStream");
    $("#player").text(url);
}
```