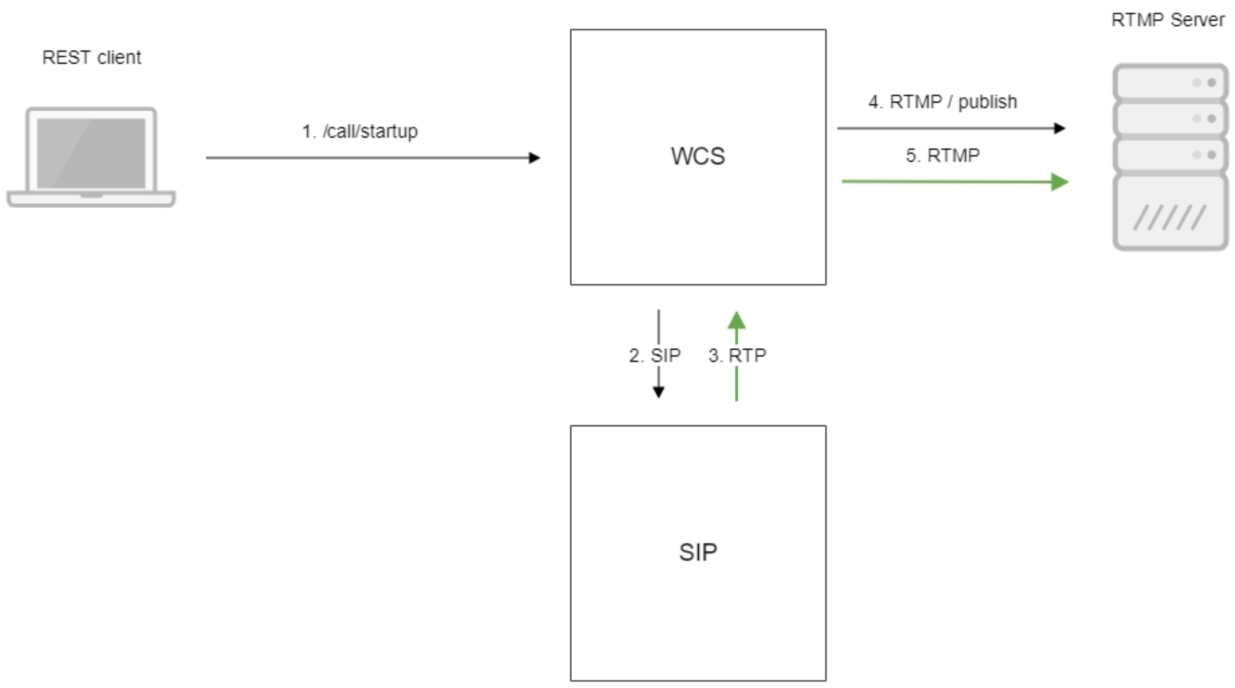
Republishing a SIP call to an RTMP stream to the given server (SIP as RTMP function)

Overview

A SIP call made through the WCS server can be captured to an RTMP stream and rebroadcast to the specified RTMP server when the call is created. One usage example is publishing a call to Facebook or Youtube.

Operation flowchart



1. The browser starts a call by sending the /call/startup REST query.
2. WCS connects to the SIP server
3. The SIP server sends the RTP stream of the call to WCS
4. WCS connects to the RTMP server
5. The RTMP server receives the RTMP stream

Quick manual on testing

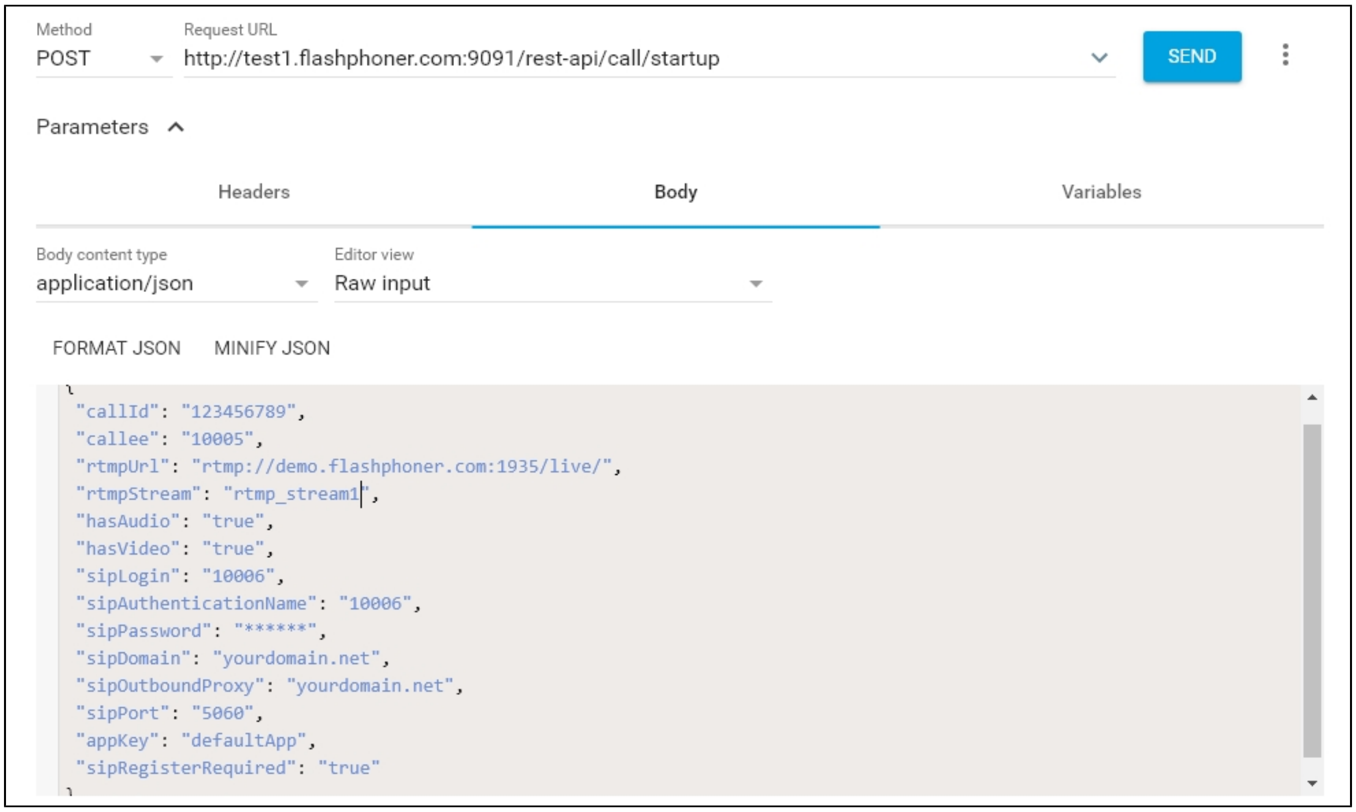
1. For the test we use:

- two SIP accounts;
- a softphone to answer the call;
- theREST clientin the Chrome browser;
- the RTMP server to receive the broadcast;
- thePlayerweb application to play the stream from the RTMP server.

2. Open the REST client. Send /call/startup to the WCS server and specify in the parameters of the query the following data:

- parameters of your SIP account the call is made from;

- URL of the RTMP server the call is republished to, in this case - specify the URL of the WCS server;
- the name of the stream to rebroadcast the call to (the rtmpStream parameter), for instance, rtmp_stream1;
- the name of your second SIP account where the call is made to.

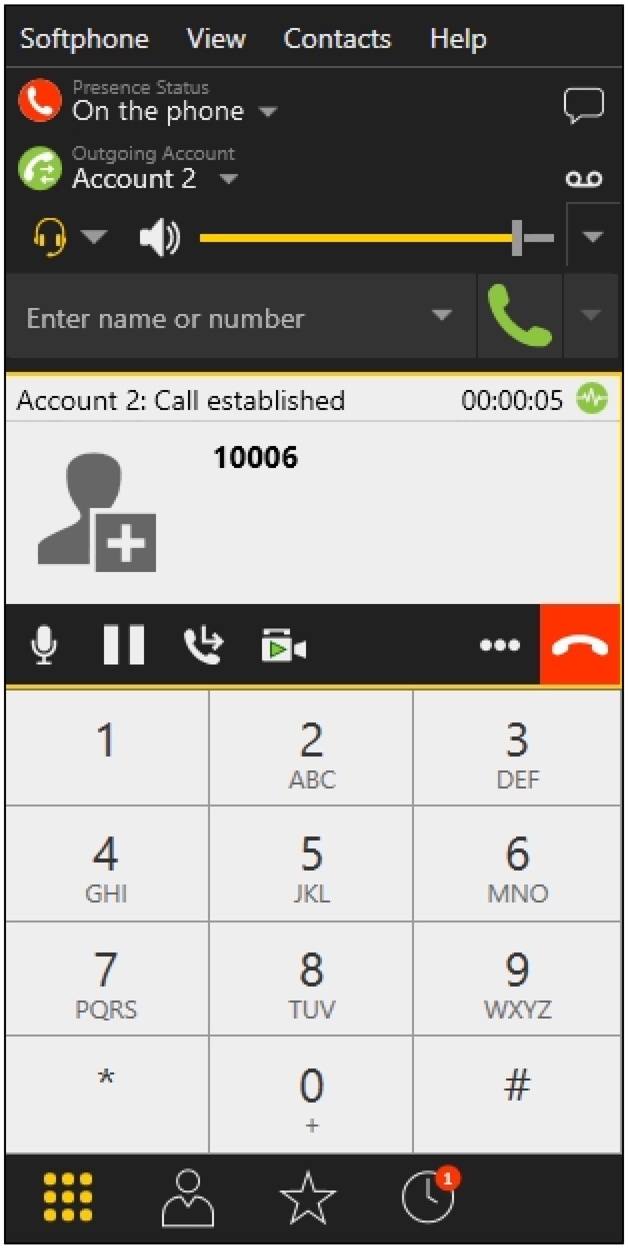


Method: POST
Request URL: http://test1.flashphoner.com:9091/rest-api/call/startup

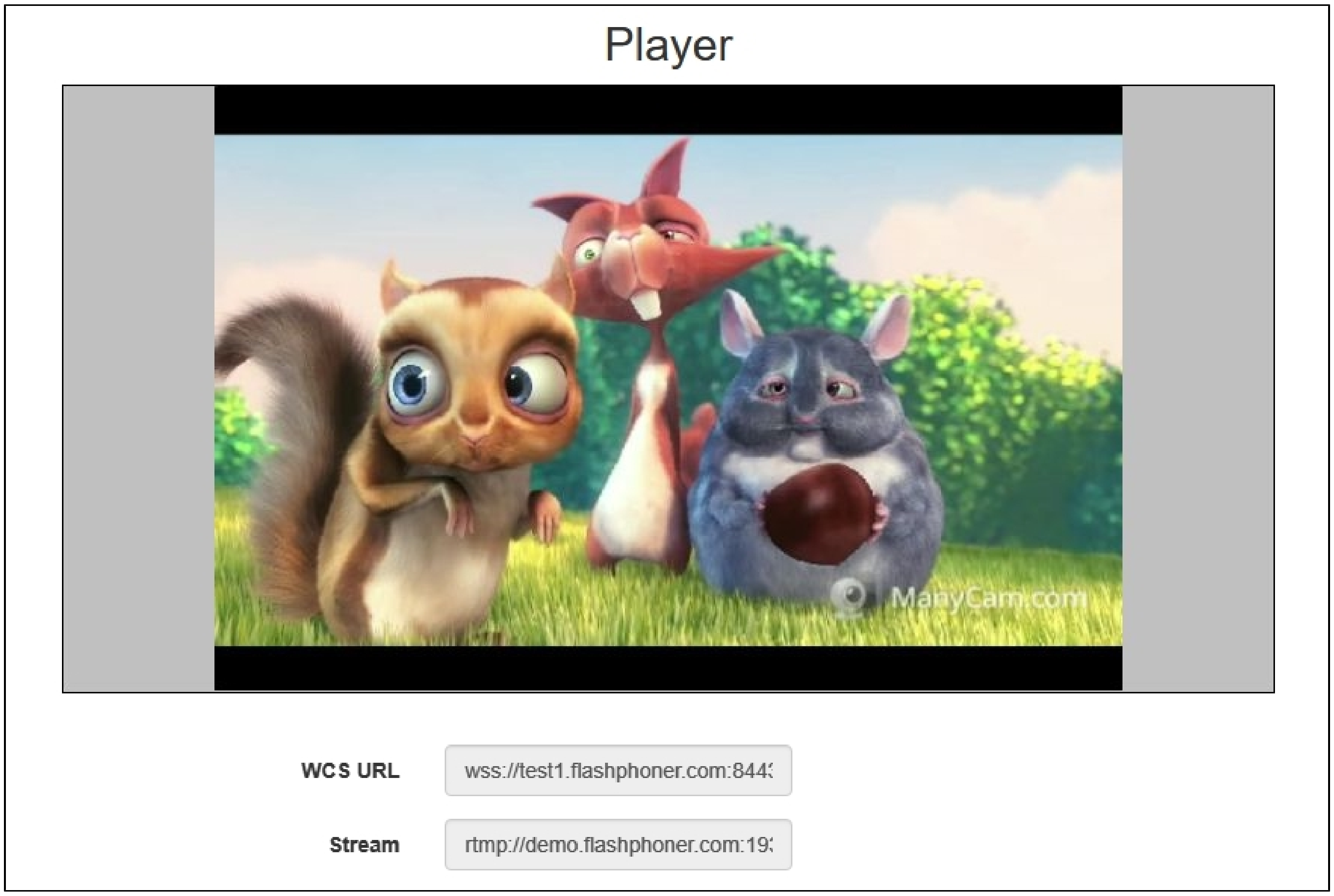
Body content type: application/json
Editor view: Raw input

```
{
 "callId": "123456789",
 "callee": "10005",
 "rtmpUrl": "rtmp://demo.flashphoner.com:1935/live/",
 "rtmpStream": "rtmp_stream1",
 "hasAudio": "true",
 "hasVideo": "true",
 "sipLogin": "10006",
 "sipAuthenticationName": "10006",
 "sipPassword": "******",
 "sipDomain": "yourdomain.net",
 "sipOutboundProxy": "yourdomain.net",
 "sipPort": "5060",
 "appKey": "defaultApp",
 "sipRegisterRequired": "true"
}
```

3. Receive and answer the call on the softphone:



4. Open the Player web application. Specify the URL of the RTMP server and the name of the RTMP stream the call is redirected to, then click the "Play" button. The call starts playing:



WCS URL wss://test1.flashphoner.com:844

Stream rtmp://demo.flashphoner.com:19

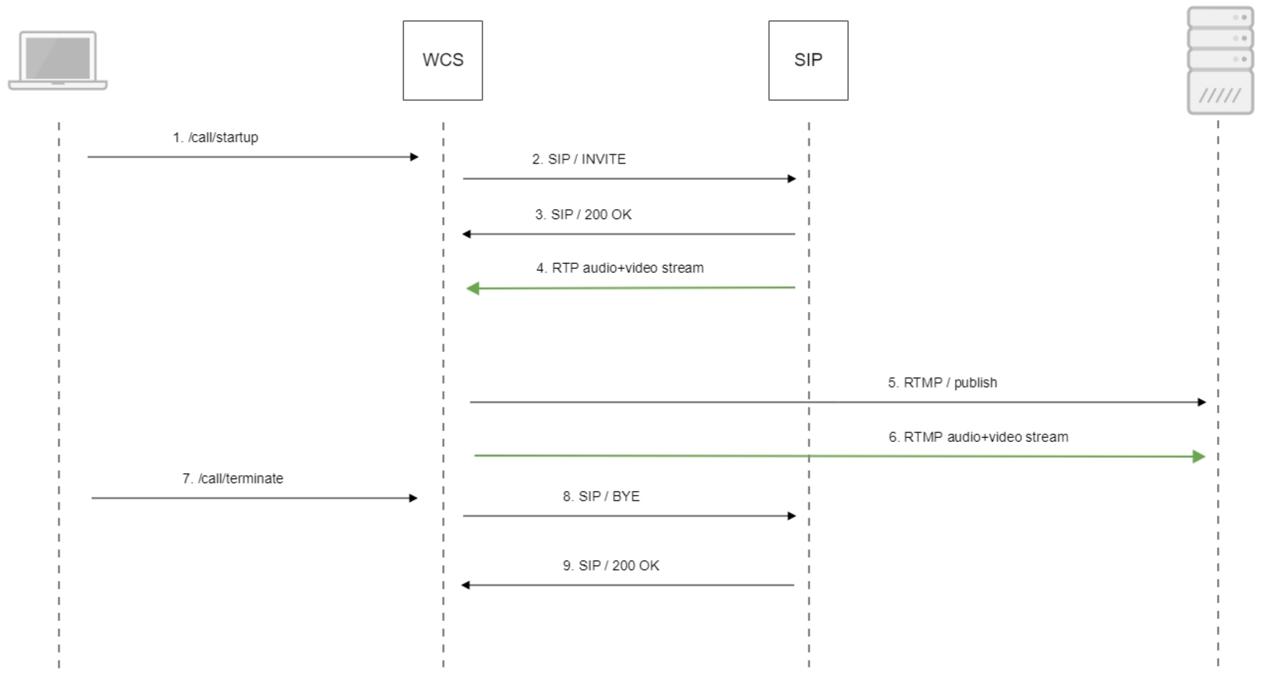
5. Terminate the call in the softphone.

Call flow

Below is the call flow when using the SIP as RTMP example to create a call

sip-as-rtmp-4.html

sip-as-rtmp-4.js



1. Sending the /call/startup REST query:

sendREST()code

```
function startCall() {
    ...
    var url = field("restUrl") + "/call/startup";
    callId = generateCallID();
    $("#sipCallId").val(callId);
    ...
    var RESTCall = {};
    RESTCall.toStream = field("rtmpStream");
    RESTCall.hasAudio = field("hasAudio");
    RESTCall.hasVideo = field("hasVideo");
    RESTCall.callId = callId;
    RESTCall.sipLogin = field("sipLogin");
    RESTCall.sipAuthenticationName = field("sipAuthenticationName");
    RESTCall.sipPassword = field("sipPassword");
    RESTCall.sipPort = field("sipPort");
    RESTCall.sipDomain = field("sipDomain");
    RESTCall.sipOutboundProxy = field("sipOutboundProxy");
    RESTCall.appKey = field("appKey");
    RESTCall.sipRegisterRequired = field("sipRegisterRequired");

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

    RESTCall.callee = field("callee");

    var data = JSON.stringify(RESTCall);

    sendREST(url, data);
    startCheckCallStatus();

}
```

2. Establishing a connection to the SIP server

3. Receiving confirmation from the SIP server

4. The RTP stream is sent to the WCS server

5. Publishing of the RTMP stream on the RTMP server

6. The RTMP stream is passed to the RTMP server

7. Sending the /call/terminate REST query:

sendREST()code

```
function hangup() {
    var url = field("restUrl") + "/call/terminate";
    var currentCallId = { callId: callId };
    var data = JSON.stringify(currentCallId);
    sendREST(url, data);
}
```

8. Sending the command to the SIP server

9. Receiving confirmation from the SIP server

Known issues

1. Stream captured from SIP call, can not be played, if RTP session is not initialized for this stream.

Symptoms: SIP stream is published on server, but can not be played.

Solution: enable RTP session initializing with the following parameter

```
rtp_session_init_always=true
```

2. SIP callee does not recognize DTMF signals if audio data generation is no enabled

Symptoms: SIP callee does not recognize PIN code sent as DTMF

Solution: enable audio and video data generation for SIP call with the following parameter

```
generate_av_for_ua=all
```

3. Freezes may occur, audio may be out of sync with video when republishing a SIP call stream as RTMP

Symptoms: freezes and audio/video out of sync are observed while playing an RTMP stream republished from a SIP call

Solution:

a) in WCS builds before 5.2.1541 add the delay to audio/video generator start

```
generate_av_start_delay=1000
```

b) update WCS to build 5.2.1541 where the issue was fixed

4. RTP traffic buffering should be enabled in some cases when republishing SIP as Stream or SIP as RTMP

Symptoms: audio and video may be out of sync when playing a SIP call RTMP stream

Solution: update WCS to build 5.2.1910 and enable RTP traffic buffering

```
rtp_in_buffer=true
```