

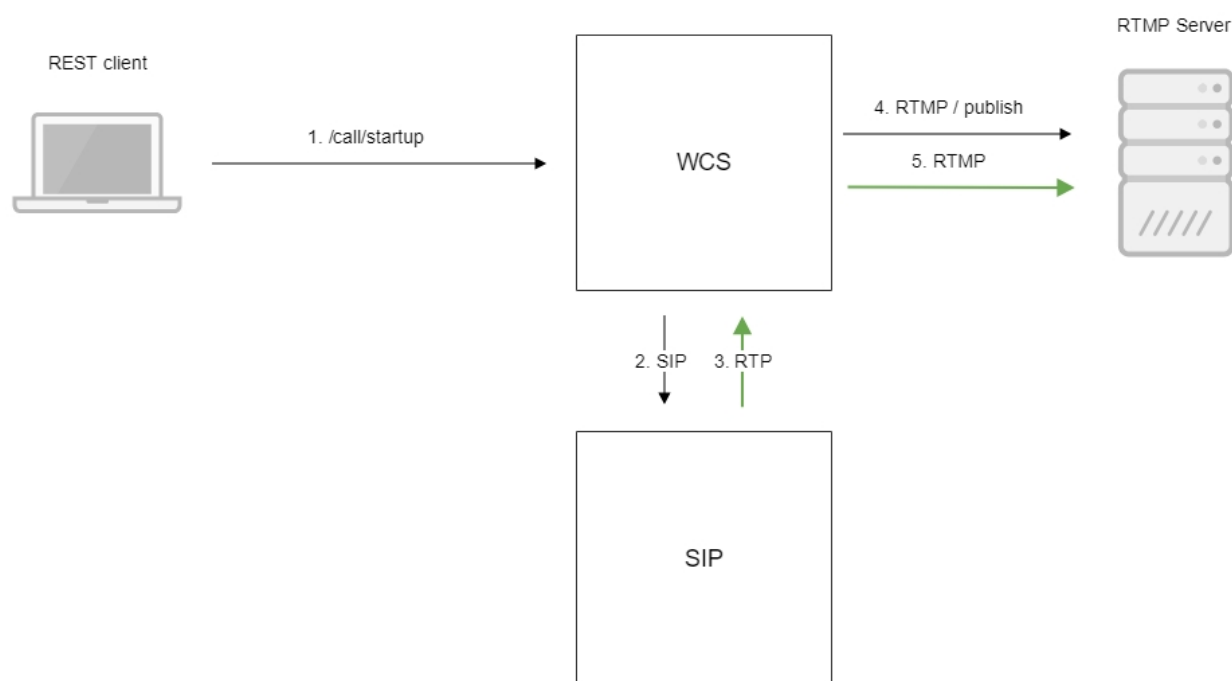
# Ретрансляция SIP-звонка в RTMP поток на заданный сервер (функция SIP as RTMP)

- [Описание](#)
  - [Схема работы](#)
- [Краткое руководство по тестированию](#)
- [Последовательность выполнения операций \(Call Flow\)](#)
- [Известные проблемы](#)

## Описание

SIP-звонок, произведенный через WCS-сервер, может быть захвачен в RTMP-поток и ретранслирован на указанный RTMP-сервер при создании звонка. Одним из примеров использования может быть трансляция звонка на Facebook или Youtube.

## Схема работы



1. Браузер начинает звонок с помощью REST-вызова /call/startup
2. WCS соединяется с SIP-сервером
3. SIP-сервер передает RTP-поток звонка на WCS
4. WCS соединяется с RTMP-сервером
5. RTMP-сервер получает RTMP-поток

## Краткое руководство по тестированию

1. Для тестирования используем:

- два SIP-аккаунта;
- программный телефон для ответа на звонок;
- [REST-клиент](#) в браузере Chrome;
- RTMP-сервер для приема трансляции;
- веб-приложение [Player](#) для воспроизведения потока с RTMP-сервера.

2. Откройте REST-клиент. Отправьте запрос /call/startup на WCS-сервер, указав в параметрах запроса:

- параметры Вашего SIP-аккаунта, с которого будет совершен звонок;

- URL RTMP-сервера, куда будет ретранслирован звонок, в данном случае укажите URL WCS-сервера;
- имя потока для ретрансляции звонка (параметр rtmpStream), например, rtmp\_stream1;
- имя Вашего второго SIP-аккаунта, на который будет совершаться звонок.

Method

Request URL

POST

http://p11.flashphoner.com:9091/rest-api/call/startup

SEND

Parameters ^

Headers

Body

Variables

Body content type

Editor view

application/json

Raw input

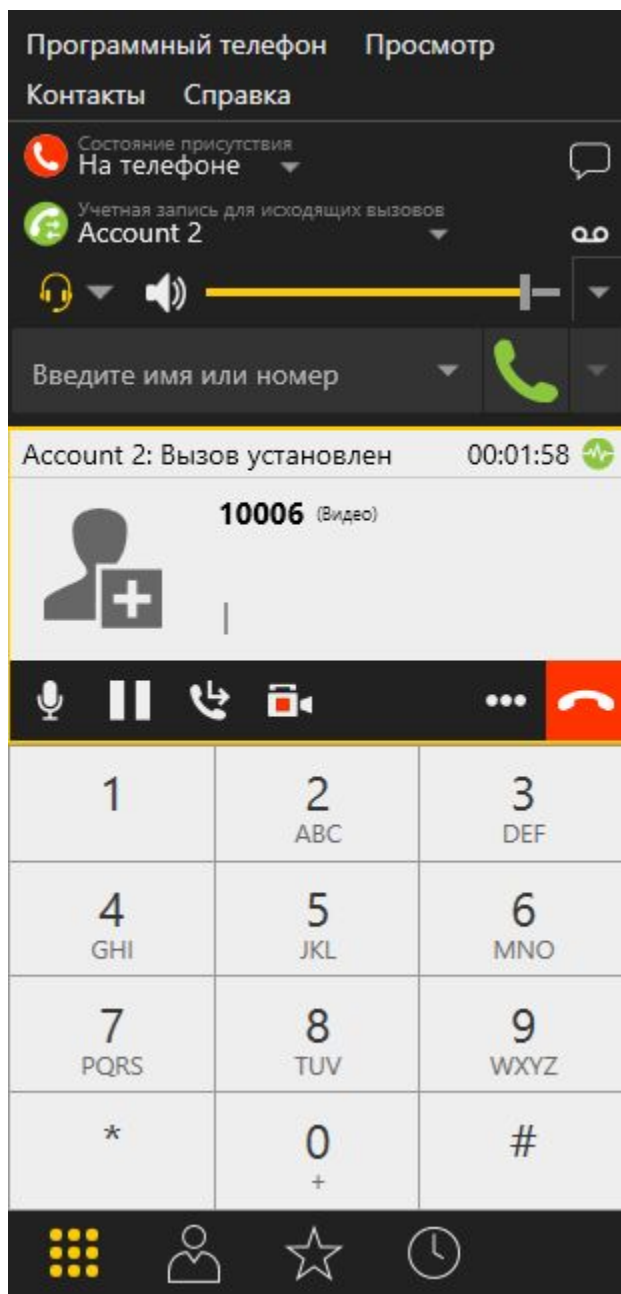
FORMAT JSON MINIFY JSON

```
{  "callId": "123456789",  "callee": "10005",  "rtmpUrl": "rtmp://p11.flashphoner.com:1935/live/",  "rtmpStream": "rtmp_stream1",  "hasAudio": "true",  "hasVideo": "true",  "sipLogin": "10006",  "sipAuthenticationName": "10006",  "sipPassword": "*****",  "sipDomain": "domain.net",  "sipOutboundProxy": "domain.net",  "sipPort": "5060",  "appKey": "defaultApp",  "sipRegisterRequired": "true"}
```

200 OK 199.60 ms

DETAILS ^

3. Примите входящий звонок на программном телефоне:



4. Откройте веб-приложение Player. Укажите URL RTMP-сервера и имя RTMP-потока, в который перенаправлен звонок, и нажмите кнопку "Play". Начнется воспроизведение звонка:

## Player



**WCS URL**

wss://p11.flashphoner.com

**Stream**

rtmp://p11.flashphoner.com

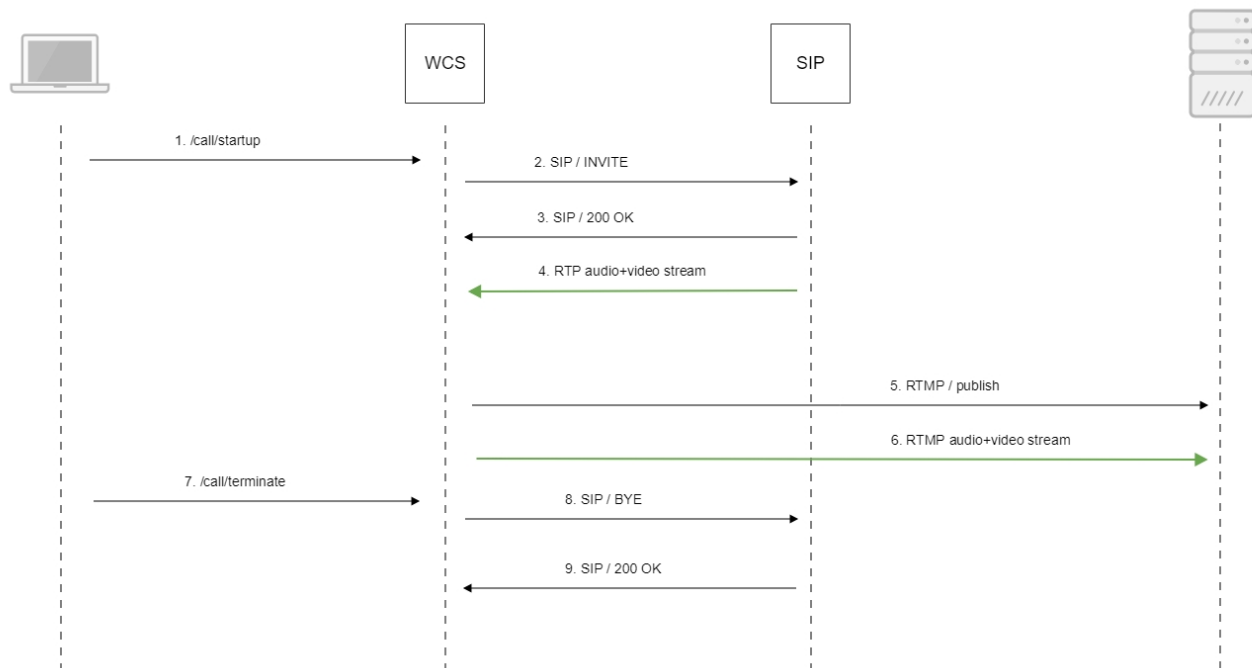
5. Завершите звонок со стороны программного телефона.

## Последовательность выполнения операций (Call Flow)

Ниже описана последовательность вызовов при использовании примера SIP as RTMP для создания звонка

[sip-as-rtmp-4.html](#)

[sip-as-rtmp-4.js](#)



1. Отправка REST-запроса /call/startup:

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. Установка соединения с SIP-сервером
3. Получение подтверждения от SIP-сервера
4. RTP-поток передается на WCS-сервер
5. Публикация RTMP-потока на RTMP-сервере
6. RTMP-поток передается на RTMP-сервер
7. Отправка REST-запроса /call/terminate:

sendREST()[code](#)

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

8. Отправка команды на SIP-сервер
9. Получение подтверждения от SIP-сервера

## Известные проблемы

1. Поток, захваченный из звонка, не проигрывается, если не инициализирована RTP-сессия для этого потока

Симптомы: поток звонка создается на сервере, но не воспроизводится.

Решение: включить принудительную инициализацию RTP-сессии при помощи настройки

```
rtp_session_init_always=true
```

2. DTMF-посылки не распознаются SIP-стороной, если не включена генерация аудиоданных

Симптомы: SIP-сторона не распознает PIN-код, набранный при помощи DTMF

Решение: включить генерацию аудио и видеоданных для звонка при помощи настройки

```
generate_av_for_ua=all
```

3. При ретрансляции потока звонка как RTMP, могут быть фризы и рассинхронизация звука и видео при проигрывании потока

Симптомы: при проигрывании ретранслированного потока, наблюдаются фризы и рассинхронизация звука и видео

Решение:

- a) в сборках до [5.2.1541](#) добавить задержку на включение аудио-видео генератора

```
generate_av_start_delay=1000
```

- b) обновить WCS до сборки [5.2.1541](#), в которой данная проблема исправлена

4. При ретрансляции видеозвонка в RTMP поток в некоторых случаях необходимо включить буферизацию RTP трафика

Симптомы: при видеозвонках на некоторые софтофоны заметна рассинхронизация между видео и аудио при проигрывании RTMP потока

Решение: обновить WCS до сборки [5.2.1910](#) и включить буферизацию RTP трафика

```
rtp_in_buffer=true
```