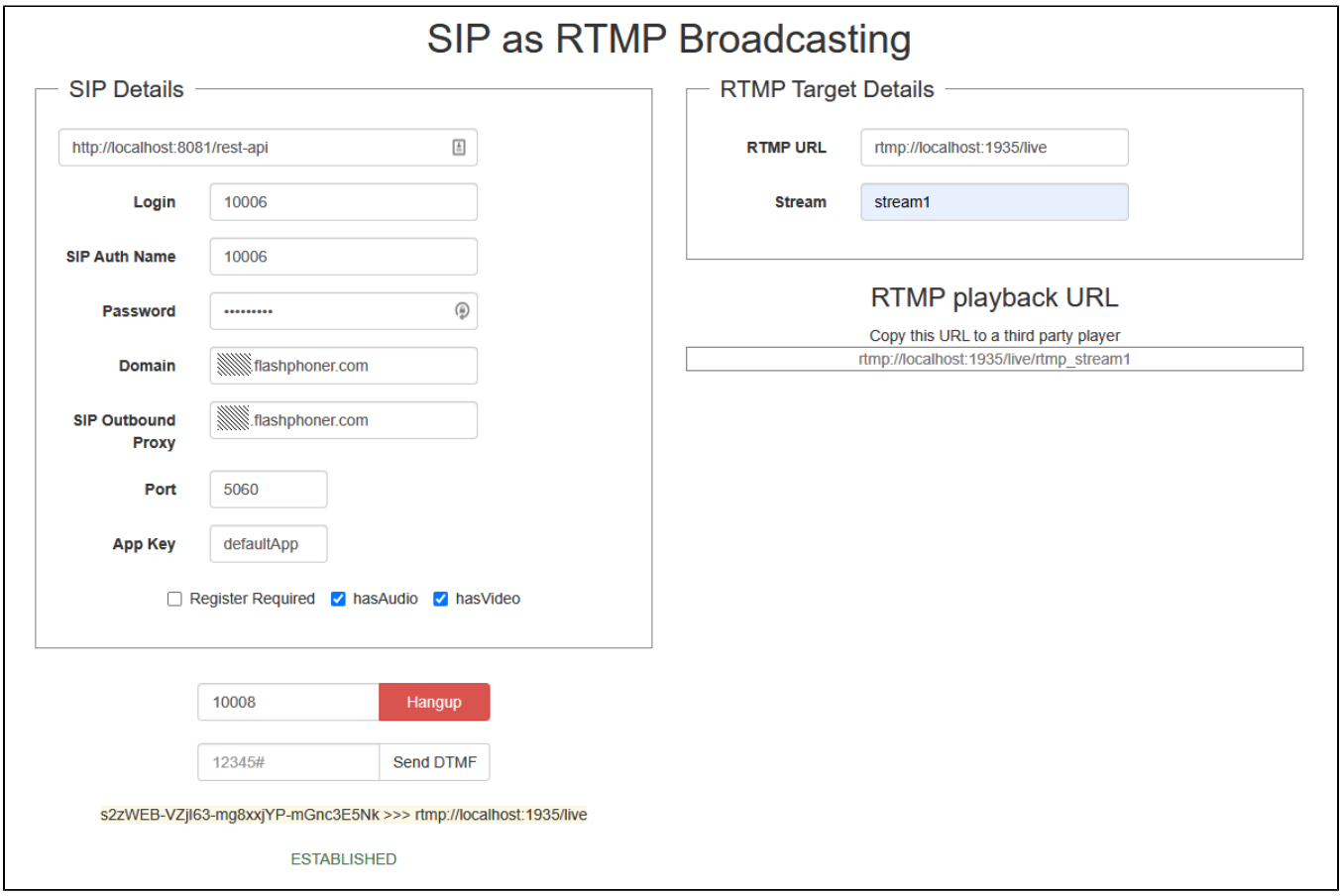
SIP as RTMP

- Пример доставки видеопотока с SIP на RTMP-сервер
- Код примера
- Работа с кодом примера

Пример доставки видеопотока с SIP на RTMP-сервер

Данный пример показывает, как можно сделать вызов на SIP, получить от SIP стороны аудио и видео трафик и затем перенаправить полученный видеопоток на сторонний RTMP-сервер, встроенный в WCS RTMP-сервер, или стороннюю CDN, которая принимает RTMP-потоки для дальнейшей раздачи.



На скриншоте происходит следующее:

1. Заполняем в левой части данные SIP-аккаунта, который будет использоваться для звонка. Если SIP сервер не требует авторизации, можно указать произвольный логин и пароль, например 'abcd'.
2. В правой части вводим RTMP-адрес сервера и название видеопотока. На этот адрес будет перенаправлен SIP трафик в случае успешного соединения.
3. Вызываем SIP-абонента под номером '10050'. Ниже есть возможность отправить DTMF-сигнал, если на SIP-стороне задействовано голосовое меню. После того как соединение с SIP будет установлено успешно, отобразится статус ESTABLISHED.
4. RTMP URL видеопотока можно скопировать в сторонний RTMP плеер (ffplay или VLC)

Код примера

Пример представляет собой простого REST-клиента, написанного на JavaScript и находится по следующему пути:

/usr/local/FlashphonerWebCallServer/client2/examples/demo/sip/sip-as-rtmp.html

sip-as-rtmp.js - скрипт, обеспечивающий REST вызовы на WCS-сервер
sip-as-rtmp.html - страница примера

Работа с кодом примера

Для разбора кода возьмем версию файла sip-as-rtmp.js с хешем ecbadc3, которая находится здесь и доступна для скачивания в соответствующей сборке 2.0.212.

1. Отправка REST / HTTP - запросов.

code

Отправка происходит методом POST с ContentType application/json AJAX запросом с использованием фреймворка jquery.

```
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. Создание исходящего звонка при помощи REST-запроса /call/startup

code

Из текстовых форм собираются данные для установки соединения (connection) и данные для звонка (RESTCall)

```
    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. Получение статуса звонка запросом /call/getStatus.

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. Отправка DTMF сигнала запросом /call/sendDTMF.

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. Отображение RTMP URL на странице для копирования в сторонний плеер

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);
}
```