

Захват потока с SIP-звонка

- [Описание](#)
 - [Типичный сценарий использования](#)
 - [Поддерживаемые протоколы](#)
 - [Поддерживаемые кодеки на SIP](#)
 - [Поддерживаемые кодеки на RTMP](#)
- [REST-вызовы](#)
 - [Общие правила](#)
 - [REST-методы и статусы ответа](#)
 - [Параметры](#)
 - [SDP параметры recvonly и sendrecv](#)
 - [Дополнительная информация по статусам](#)
- [Настройка](#)
 - [Конфигурация запуска](#)
 - [Приложение CallApp](#)
 - [Настройка HTTPS](#)
 - [Включение HTTPS](#)
 - [Аутентификация](#)
- [Известные проблемы](#)

Описание

WCS может выступать в качестве WebRTC-SIP шлюза. При этом аудио- и видеопоток SIP-звонка, произведенного через WCS, может быть захвачен и воспроизведен в браузере, либо ретранслирован на другой сервер.

Типичный сценарий использования

1. Между WCS и SIP-устройством (SIP MCU, сервер конференций или SIP-софтфон) установлен видеозвонок
2. WCS получает аудио и видео данные с этого SIP-устройства
3. Полученный аудио и видео трафик WCS-сервер перенаправляет на RTMP-сервер или другое устройство, способное принять и обработать RTMP-поток

Поддерживаемые протоколы

- RTMP
- SIP

Поддерживаемые кодеки на SIP

- Видеокодеки: H.264, VP8
- Аудиокодеки: G.711, Speex, Opus

Поддерживаемые кодеки на RTMP

- Видеокодеки: H.264
- Аудиокодеки: AAC, G.711, Speex

Захват и ретрансляция SIP-звонков управляется при помощи REST API вызовов.

REST-вызовы

Общие правила

1. Каждый SIP-звонок при создании может быть ассоциирован только с одним RTMP-поток. В случае, если инициируется новый SIP-звонок с тем же RTMP URL и именем потока (rtmpUrl+rtmpStream), как у существующего звонка, этот второй звонок будет отклонен сервером с HTTP статусом 409 Conflict. Однако, при ретрансляции звонка в RTMP-поток при помощи REST-запроса /push/startup, количество RTMP-потоков, создаваемых из одного звонка, не ограничивается.
2. SIP Call ID звонка должен быть уникальным. Попытка инициировать новый SIP-звонок с уже существующим Call ID будет отклонена WCS-сервером с HTTP-статусом 409 Conflict.

REST-запрос должен быть HTTP/HTTPS POST запросом в таком виде:

- HTTP: <http://sip-as-rtmp.flashphoner.com:8081/rest-api/call/startup>
- HTTPS: <https://sip-as-rtmp.flashphoner.com:8444/rest-api/call/startup>

Здесь:

- sip-as-rtmp.flashphoner.com - адрес WCS-сервера

- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- rest-api - обязательная часть URL
- /call/startup - используемый REST-метод

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа
/call /startup	<pre>{ "callId": "123456711", "callee": "10000", "toStream": "stream1", "rtmpUrl": "rtmp://localhost:1935/live/", "rtmpStream": "rtmp_stream1", "hasAudio": "true", "hasVideo": "true", "sipLogin": "10009", "sipAuthenticationName": "10009", "sipPassword": "1234", "sipDomain": "226.226.225.226", "sipOutboundProxy": "226.226.225.226", "sipPort": "5060", "appKey": "defaultApp", "sipRegisterRequired": "false" }</pre>		200 - Звонок принят на обработку 409 - Конфликт с существующим RTMP URL
/call/find	<pre>{"status" : "ESTABLISHED"}</pre> <p>or</p> <pre>{ "callId": "R08NQya-5NMe5v7q-JNkboaS-CGMlFi" }</pre>	<pre>[{ "custom": {}, "nodeId": null, "appKey": null, "sessionId": null, "callId": "R08NQya-5NMe5v7q-JNkboaS-CGMlFi", "parentCallId": null, "incoming": false, "status": "ESTABLISHED", "sipStatus": 200, "rtmpUrl": null, "rtmpStream": null, "streamName": null, "rtmpStreamStatus": null, "caller": "001", "callee": "002", "hasAudio": true, "hasVideo": false, "sdp": null, "visibleName": "001", "inviteParameters": null, "mediaProvider": "Flash", "sipMessageRaw": null, "isMsrp": false, "target": null, "holdForTransfer": false }]</pre>	200 - звонок найден 404 - звонок с указанными параметрами не найден

/call /find_all		<pre>[{ "custom": {}, "nodeId": null, "appKey": null, "sessionId": null, "callId": "R08NQya-5NMe5v7q- JNkboaS-CGMlFi", "parentCallId": null, "incoming": false, "status": "ESTABLISHED", "sipStatus": 200, "rtmpUrl": null, "rtmpStream": null, "streamName": null, "rtmpStreamStatus": null, "caller": "001", "callee": "002", "hasAudio": true, "hasVideo": false, "sdp": null, "visibleName": "001", "inviteParameters": null, "mediaProvider": "Flash", "sipMessageRaw": null, "isMsrp": false, "target": null, "holdForTransfer": false }]</pre>	200 - звонки найдены 404 - звонков не найдено
/call /terminate	<pre>{ "callId" : "becee2c0-13b4-11e7-b817- c1649197cae8" }</pre>		200 - звонок прерван 404 - звонок не найден
/call /send_dtmf	<pre>{ "callId" : "52173e00-13b6-11e7-b817- c1649197cae8", "dtmf" : "9", "type" : "RFC2833" }</pre>		200 - DTMF отправлен 404 - звонок не найден

Параметры

Имя параметра	Описание	Пример
callId	SIP Call ID - уникальный идентификатор в виде строки	Xq2tLcX89tTjaji
callee	SIP callee - вызываемый абонент	10001
toStream	Имя потока на WCS-сервере, в который транслируется звонок	call_stream1
rtmpUrl	RTMP URL - адрес RTMP-сервера	rtmp://rtmp-server.flashphoner.com:1935/live Здесьlive- имя RTMP-приложения. В RTMP URL также могут быть указаны instance name и строка запроса, например rtmp://rtmp-server.flashphoner.com:1935/live/_definst_?param1=value1¶m2=value2
rtmpStream	Название RTMP-потока на RTMP-сервере	streamName2
hasAudio	Если true, SDP будет иметь 'sendrecv' параметр в аудио части. Если false, то 'recvonly'.	true

hasVideo	Если true, SDP будет иметь 'sendrecv' параметр в видео части. Если false, то 'recvonly'.	true
status	Статус звонка на WCS-сервере	ESTABLISHED Полный список статусов доступен здесь (см CallStatusEvent).
sipStatus	Ассоциированный SIP-статус	200
rtmpStreamStatus	Статус RTMP-потока	RTMP_STREAM_ACTIVE RTMP_STREAM_WAIT - RTMP-поток инициализируется RTMP_STREAM_ACTIVE - RTMP-поток проинициализирован и соединение установлено RTMP_CONNECTION_LOST - RTMP-соединение было потеряно RTMP_CONNECTION_FAILED - RTMP-соединение не было установлено
caller	SIP caller -вызывающий абонент	
visibleName	Отображаемое имя вызывающего абонента	
mediaProvider	НЕ ИСПОЛЬЗУЕТСЯ	НЕ ИСПОЛЬЗУЕТСЯ

SDP параметры recvonly и sendrecv

SDP параметры recvonly и sendrecv

Существует два основных режима для SIP-REST звонков:

1. sendrecv

```
hasAudio: true
hasVideo: true

v=0
o=Flashphoner 0 1437391553771 IN IP4 sip.flashphoner.com
s=Flashphoner/1.0
c=IN IP4 sip.flashphoner.com
t=0 0
m=audio 31022 RTP/AVP 8 0
c=IN IP4 46.101.139.106
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=ptime:20
a=rtcp:31023 IN IP4 sip.flashphoner.com
a=sendrecv
a=ssrc:1478013757 cname:rtp/audio/Xq2t1LcX89tTjaji
m=video 31024 RTP/AVP 112 113
c=IN IP4 sip.flashphoner.com
a=rtpmap:112 H264/90000
a=fmtp:112 packetization-mode=1; profile-level-id=420020
a=rtpmap:113 H264/90000
a=fmtp:113 packetization-mode=0; profile-level-id=420020
a=rtcp-fb:* ccm fir
a=rtcp-fb:* nack
a=rtcp-fb:* nack pli
a=rtcp:31025 IN IP4 sip.flashphoner.com
a=sendrecv
a=ssrc:979076678 cname:rtp/video/Xq2t1LcX89tTjaji
```

2. recvonly

```
hasAudio: false
hasVideo: false

v=0
o=Flashphoner 0 1437391553771 IN IP4 sip.flashphoner.com
s=Flashphoner/1.0
c=IN IP4 sip.flashphoner.com
t=0 0
m=audio 31022 RTP/AVP 8 0
c=IN IP4 46.101.139.106
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
aptime:20
a=rtcp:31023 IN IP4 sip.flashphoner.com
a=recvonly
a=ssrc:1478013757 cname:rtp/audio/Xq2tlLcX89tTjaji
m=video 31024 RTP/AVP 112 113
c=IN IP4 sip.flashphoner.com
a=rtpmap:112 H264/90000
a=fmtp:112 packetization-mode=1; profile-level-id=420020
a=rtpmap:113 H264/90000
a=fmtp:113 packetization-mode=0; profile-level-id=420020
a=rtcp-fb:* ccm fir
a=rtcp-fb:* nack
a=rtcp-fb:* nack pli
a=rtcp:31025 IN IP4 sip.flashphoner.com
a=recvonly
a=ssrc:979076678 cname:rtp/video/Xq2tlLcX89tTjaji
```

В обоих случаях WCS не отправляет RTP аудио и видео трафик, т.к. инициатором звонка выступает REST-клиент, который не является источником аудио и видео потоков.

При этом WCS может явно указать в SDP, что с его стороны не будет аудио и видео трафика (режим 'recvonly').

Если ваше SIP-устройство - это софтфон или другой SIP-телефон, он скорее всего будет сбрасывать звонок (в режиме 'sendrecv') примерно в течение минуты после установки соединения. Это происходит из-за отсутствия RTP-трафика со стороны WCS.

Некоторые софтфоны корректно поддерживают режим 'recvonly', например MicroSIP.

В других софтфонах, таких как Bria, таймер проверки RTP-активности может быть увеличен, для того чтобы получить большую длительность звонка в режиме 'sendrecv'.

Если ваше SIP-устройство - это MCU или сервер SIP-конференций, скорее всего оно корректно отработает режим 'recvonly', и звонок сможет быть установлен на длительное время.

Дополнительная информация по статусам

WCS использует внутреннее приложение 'callApp' для передачи промежуточных статусов.

Примеры:

- TRYING, RTMP_STREAM_WAIT

```
{
  "nodeId" : "w9NiNKZCtjK6C4vz1zVnzGWBjGkA2Cke@192.168.88.101",
  "appKey" : "callApp",
  "sessionId" : "127.0.0.1:1403649870519623722",
  "callId" : "Xq2t1LcX89tTjaji_3",
  "incoming" : false,
  "status" : "TRYING",
  "sipStatus" : 100,
  "rtmpUrl" : "rtmp://rtmp.flashphoner.com:1935/live",
  "rtmpStream" : "streamName2",
  "rtmpStreamStatus" : "RTMP_STREAM_WAIT",
  "caller" : "3000",
  "callee" : "3002",
  "hasAudio" : true,
  "hasVideo" : true,
  "visibleName" : "3000",
  "mediaProvider" : "Flash",
  "isMsrp" : false
}
```

- ESTABLISHED, RTMP_STREAM_ACTIVE

```
{
  "nodeId" : "w9NiNKZCtjK6C4vz1zVnzGWBjGkA2Cke@192.168.88.101",
  "appKey" : "callApp",
  "sessionId" : "127.0.0.1:1403649870519623722",
  "callId" : "Xq2t1LcX89tTjaji_3",
  "incoming" : false,
  "status" : "ESTABLISHED",
  "sipStatus" : 200,
  "rtmpUrl" : "rtmp.flashphoner.com:1935/live",
  "rtmpStream" : "streamName2",
  "rtmpStreamStatus" : "RTMP_STREAM_ACTIVE",
  "caller" : "3000",
  "callee" : "3002",
  "hasAudio" : true,
  "hasVideo" : true,
  "visibleName" : "3000",
  "mediaProvider" : "Flash",
  "isMsrp" : false
}
```

- ESTABLISHED, RTMP_CONNECTION_LOST

```
{
  "nodeId" : "w9NiNKZCt jK6C4vz1zVnzGWBjGkA2Cke@192.168.88.101",
  "appKey" : "callApp",
  "sessionId" : "127.0.0.1:1403649870519623722",
  "callId" : "Xq2t1LcX89tTjaji_3",
  "incoming" : false,
  "status" : "ESTABLISHED",
  "sipStatus" : 200,
  "rtmpUrl" : "rtmp.flashphoner.com:1935/live",
  "rtmpStream" : "streamName2",
  "rtmpStreamStatus" : "RTMP_CONNECTION_LOST",
  "caller" : "3000",
  "callee" : "3002",
  "hasAudio" : true,
  "hasVideo" : true,
  "visibleName" : "3000",
  "mediaProvider" : "Flash",
  "isMsrp" : false
}
```

Эти нотификации, которые проходят локально на стороне сервера через внутренний REST-интерфейс. См. раздел ["Управление приложениями"](#), чтобы получить больше информации о внутренних REST-приложениях. Кроме этого, может быть создано стороннее web-приложение, которое будет получать нотификации с WCS-сервера.

Настройка

Конфигурация запуска

По-умолчанию, WCS-сервер запускается в dev-режиме.

Чтобы запустить сервер с профилем 'production', раскомментируйте следующую строку в файле `/usr/local/FlashphonerWebCallServer/conf/wcs-manager.properties`:

```
-Dspring.profiles.active=production
```

В production-режиме включается поддержка HTTPS для REST.

Приложение CallApp

`http://localhost:9091/CallApp` - это внутренний адрес, доступный по умолчанию, который будет получать все промежуточные статусы звонка, сделанного через REST: RING, TRYING и др.

Кроме этого, передаются промежуточные статусы RTMP-потока, ассоциированного с этим звонком:

```
RTMP_STREAM_WAIT
RTMP_STREAM_ACTIVE
RTMP_CONNECTION_LOST
RTMP_CONNECTION_FAILED
```

Этот адрес может быть изменен через WCS CLI. См. [описание интерфейса командной строки](#), чтобы получить больше информации об [управлении и приложениями WCS](#).

Настройка HTTPS

Включение HTTPS

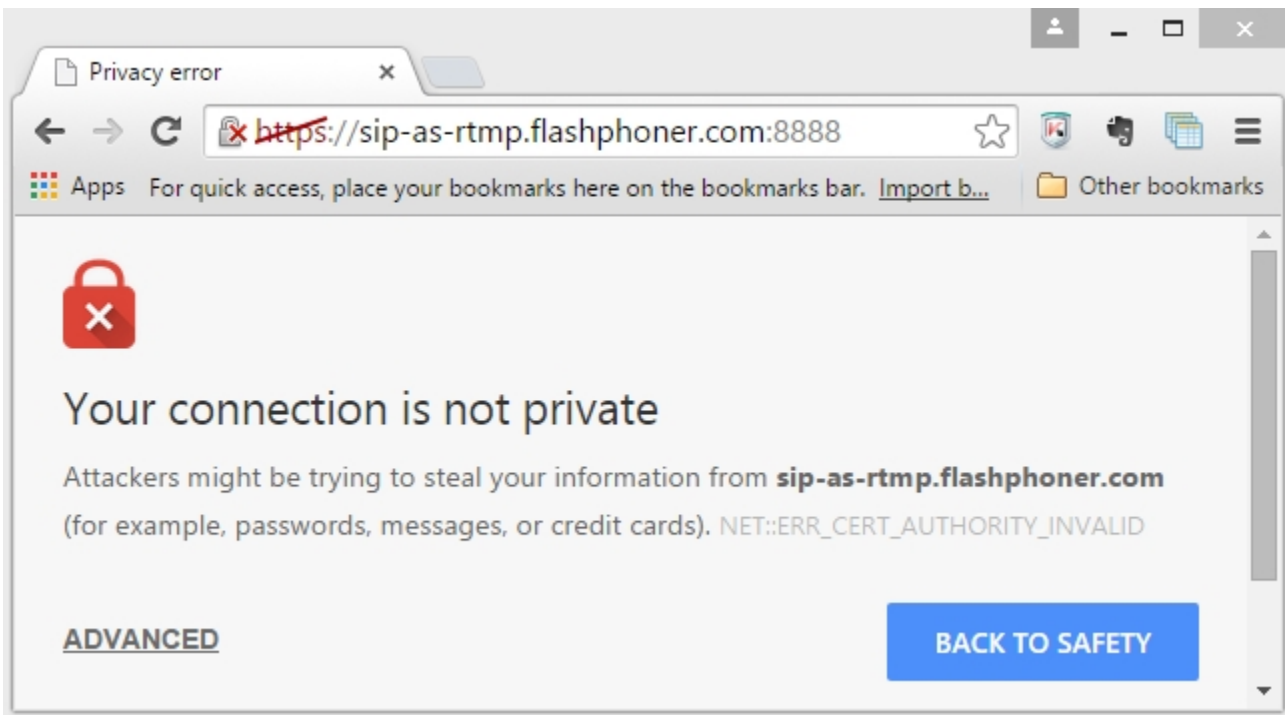
REST работает через HTTP на порту 8081 (по умолчанию) и по HTTPS на порту 8444.

Чтобы включить HTTPS, нужно запустить сервер с профилем 'production'.

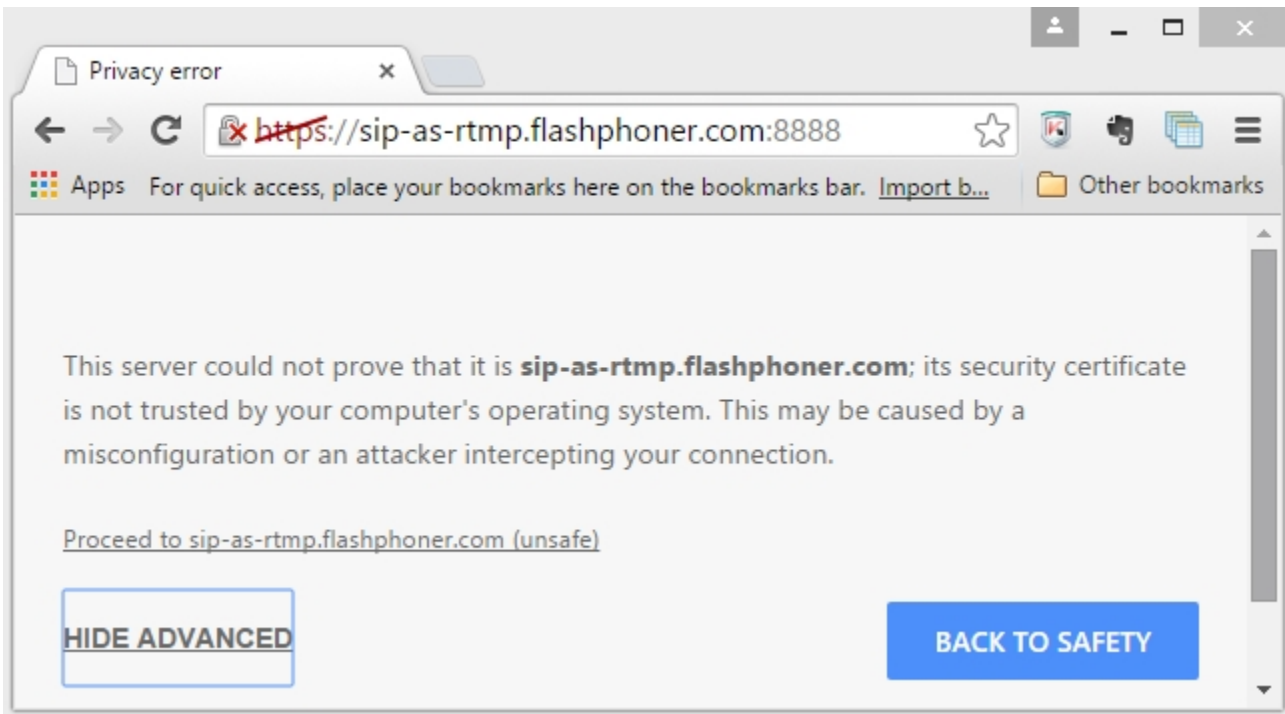
По умолчанию, WCS использует self-signed SSL сертификат.

Чтобы подтвердить исключение безопасности для этого сертификата:

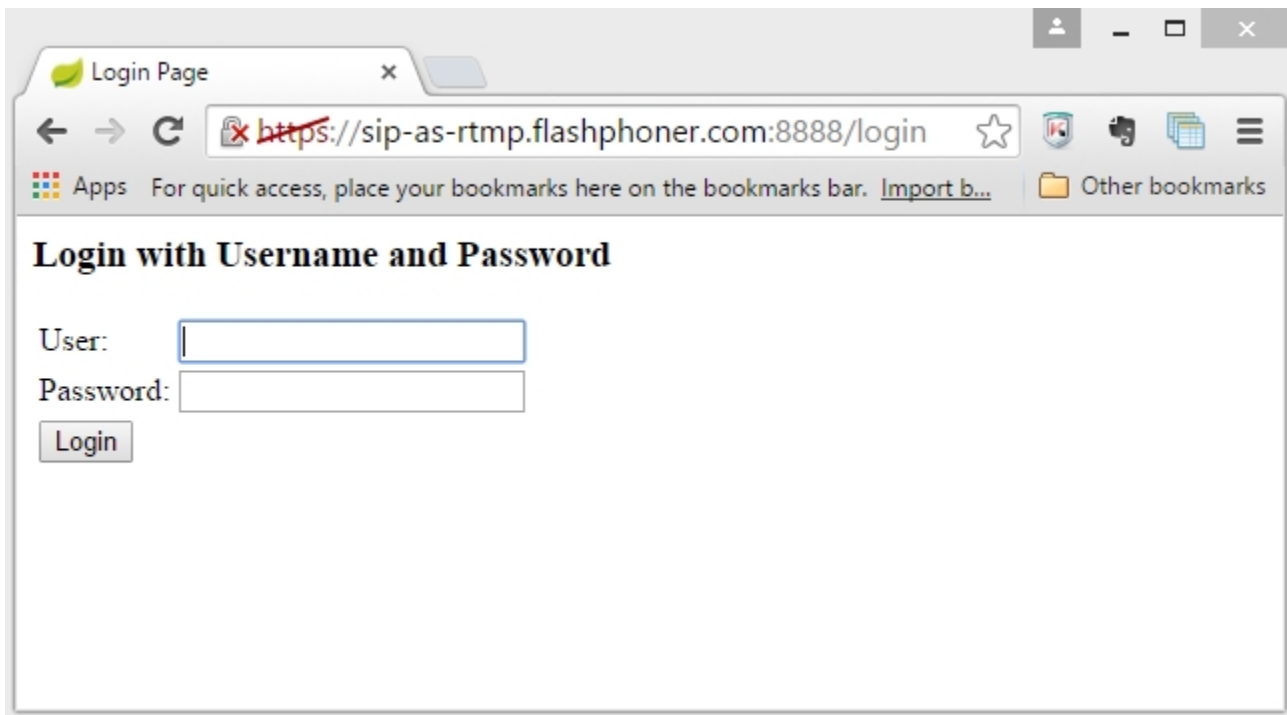
а) Откройте такой URL
`https://sip-as-rtmp.flashphoner.com:8444/`
(здесь `sip-as-rtmp.flashphoner.com` - адрес WCS-сервера)



б) Кликните 'ADVANCED'



с) Кликните 'Proceed'



После этого WCS self-signed сертификат будет импортирован в ваш браузер и HTTPS URL может быть использован для REST-звонков, например <https://sip-as-rtmp.flashphoner.com:8888/rest-api/call>

Аутентификация

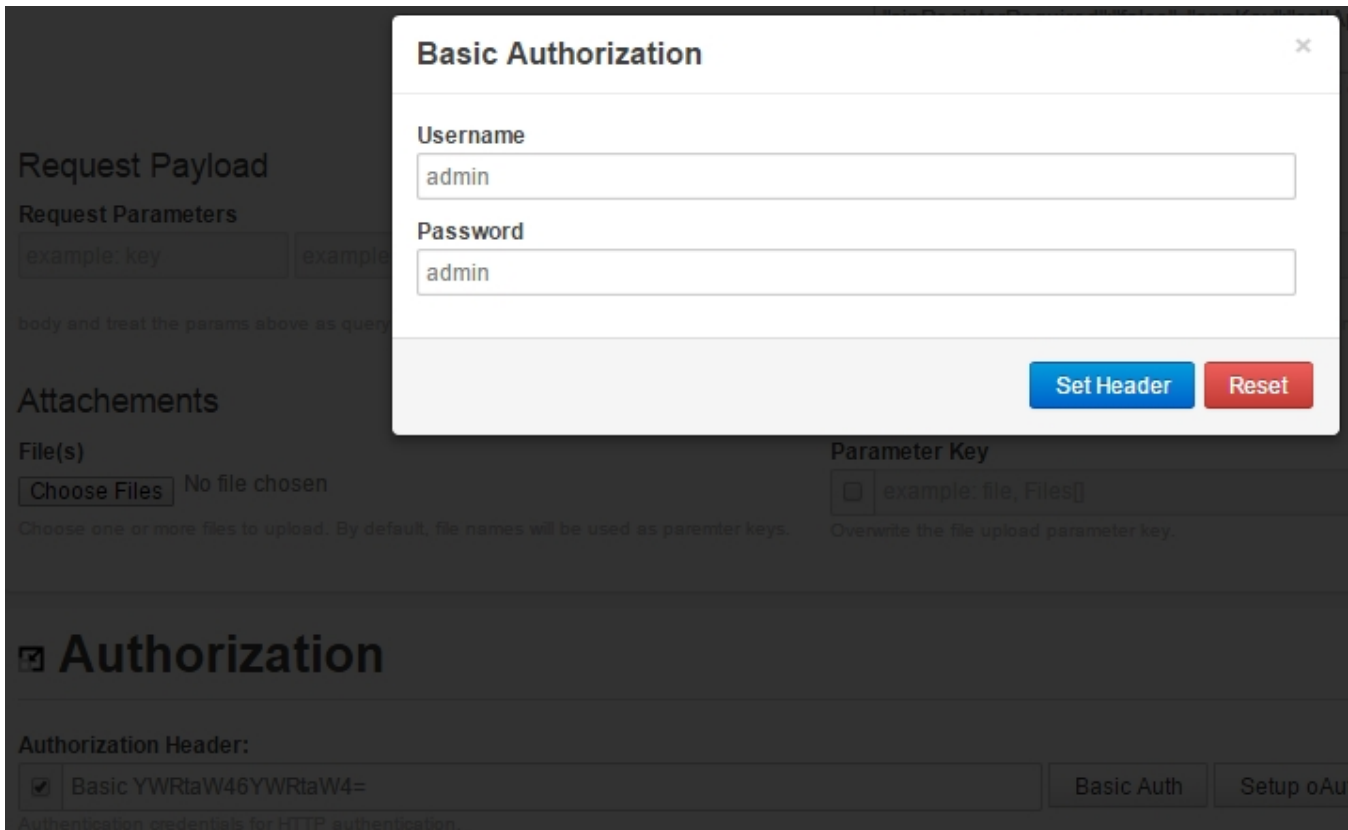
Когда включен режим 'production', каждый REST/HTTPS или REST/HTTP запрос требует [HTTP Basic Authentication](#).

Стандартные имя пользователя и пароль - это admin:admin.

Пароль можно поменять в WCS CLI. (Получить больше информации о Command Line Interface можно в [Руководстве Администратора](#).)

В REST Console авторизацию можно добавить следующим способом

- кликните 'Basic Auth' в Authorization / Authorization Header,
- добавьте admin:admin в качестве имени пользователя и пароля,
- кликните 'Set Header'



В результате будет установлен Authorization-заголовок



Известные проблемы

1. При ретрансляции [SIP as RTMP](#) на серверы Wowza и получении потока с Wowza по HLS зритель наблюдает фризы, кратковременную рассинхронизацию.

Решение: включить транскодинг на сервере, указав в файле `flashphoner.properties`

```
disable_streaming_proxy=true
```

2. При ретрансляции [SIP as Stream](#) поток аудиозвонка не воспроизводится [no WebRTC в браузере](#).

Решение: поток аудиозвонка необходимо воспроизводить в браузере как аудиопоток, указав ограничение явным образом при создании потока в скрипте плеера, например

```
session.createStream({constraints:{audio:true,video:false}}).play();
```

3. Невозможно совершить SIP-звонок при некорректно заданных параметрах звонка SIP Login, SIP Authentication name

Симптомы: звонок не совершается, при создании звонка при помощи REST API возвращается

```
HTTP/1.1 500 Internal Server Error
Access-Control-Allow-Origin: *
Content-Type: application/json
```

```
{
  "error": "Internal Server Error",
  "exception": "com.flashphoner.rest.server.exception.InternalErrorException",
  "message": "SIP login or authentication name contains reserved symbols",
  "path": "/rest-api/call/startup",
  "status": 500,
  "timestamp": 1559029484840
}
```

Решение: согласно [стандарту](#), SIP Login и SIP Authentication name должны содержать неэкранированных пробелов, спецсимволов и не должны заключаться в угловые скобки '<>'.

Например, такое заполнение полей не соответствует стандарту

```
sipLogin='Ralf C12441@host.com'
sipAuthenticationName='Ralf C'
sipPassword='demo'
sipVisibleName='null'
```

а такое соответствует

```
sipLogin='Ralf_C12441'
sipAuthenticationName='Ralf_C'
sipPassword='demo'
sipVisibleName='Ralf C'
```