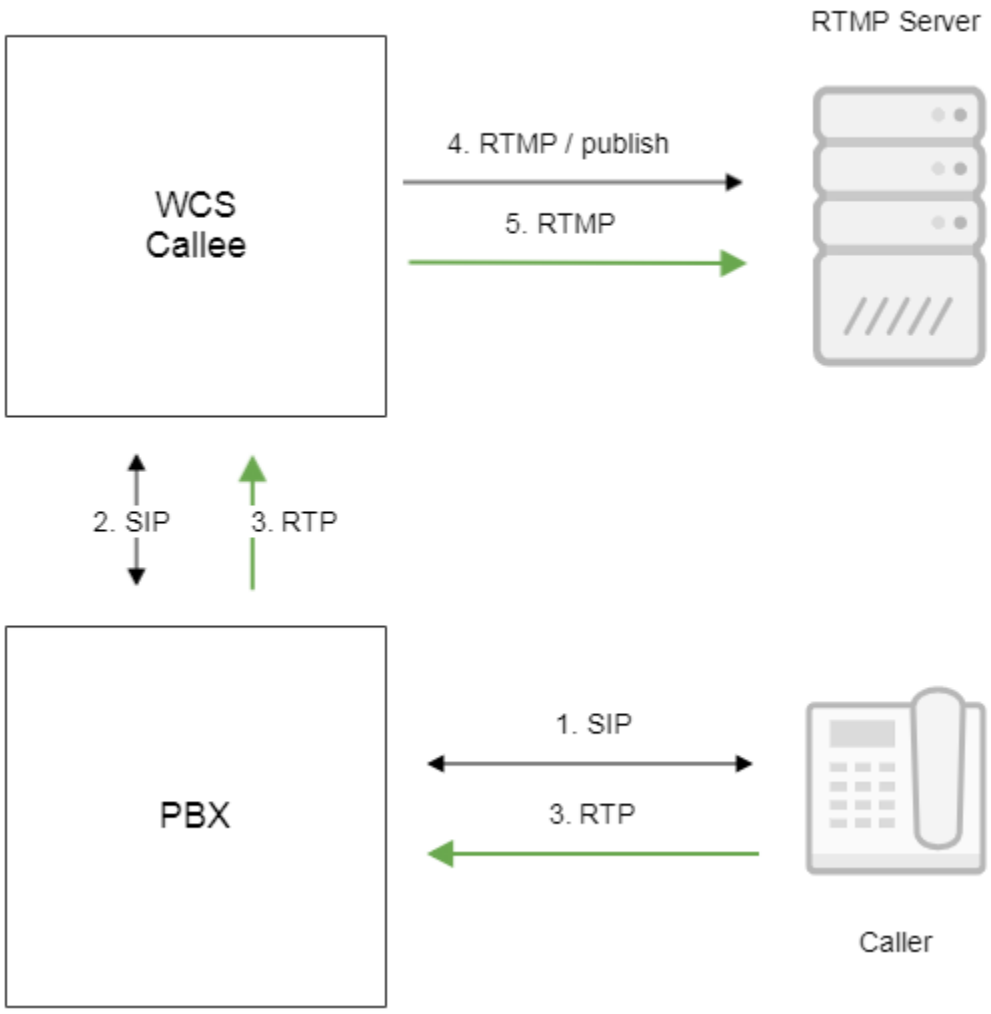
Ретрансляция входящего SIP звонка в поток

Описание

WCS 5.2 может принять входящий звонок от АТС и опубликовать поток этого звонка как RTMP на указанный сервер (например, на Wowza). Кроме того, другой поток, опубликованный на сервере или захваченный по VOD из файла в локальном или сетевом хранилище может быть пер енаправлен в звонок, в этом случае звонящий увидит и услышит этот поток.

Для этого WCS должен быть настроен на прием звонков от АТС при помощи SIP транков. Затем WCS ждет входящих звонков от АТС. После того, как звонок установлен, из этого звонка создается поток и ретранслируется на указанный RTMP сервер. Когда звонок завершится, поток также завершается.

Схема работы



1. Абонент звонит на номер, для которого настроен SIP транк
2. АТС перенаправляет звонок на WCS для приема
3. WCS получает медиапоток от звонящего абонента
4. WCS соединяется с RTMP сервером
5. WCS публикует медиапоток на RTMP сервер

Настройка

Настройка WCS

На стороне WCS, в файле flashphoner.properties должен быть установлен следующий параметр

```
sip_add_contact_id=false
```

Кроме того, в файле WCS_HOME/conf/sip_trunk.yml должен быть настроен SIP транк:

```
trunks:
 pbx_t0:
  localPort : 40000
  proxyIp : pbx_address
  remotePort : 5060
  url : rtmp://rtmp_server:1935/live
  visibleName : CUSTOM_NAME
  sdp : |
   v=0
   o=10009 2469 1555 IN IP4 0.0.0.0
   c=IN IP4 0.0.0.0
   t=0 0
   m=audio 7270 RTP/AVP 96
   a=rtpmap:96 opus/48000/2
   a=recvonly
   m=video 9202 RTP/AVP 96
   a=rtpmap:96 H264/90000
   a=fmtp:96 profile-level-id=42801F
   a=recvonly
  sdpParams :
   - b=AS:2000
   - b=RS:50
   - b=RR:100
```

Здесь

- pbx_t0 – наименование SIP транка
- localPort – порт для приема SIP звонков
- proxyIp – адрес АТС
- remotePort – порт АТС для регистрации на ней как принимающего звонки
- url – URL RTMP сервера для публикации потока
- visibleName - имя для отображения стороне, совершающей звонок, передается на АТС при регистрации
- sdp – SDP для отправки АТС в ответе 200 OK
- sdpParams - параметры, подставляемые в SDP для управления битрейтом и пропускной способностью канала передачи медиаданных звонка

WCS поддерживает TCP и UDP транспорт для SIP звонков и прослушивает порт, заданный в localPort, как по TCP, так и по UDP.

По умолчанию, имя RTMP потока будет сформировано как rtmp_0123456, где 0123456 - вызываемый номер. Для того, чтобы удалить префикс rtmp_, необходимо указать следующую настройку вфайле flashphoner.properties

```
rtmp_transponder_stream_name_prefix=
```

Настройка АТС

На стороне АТС, должен быть настроен SIP транк для перенаправления звонков WCS серверу. Звонки должны перенаправляться на порт, указанный в файле WCS_HOME/conf/sip_trunk.yml file (40000 в приведенном примере).

Например, АТС OpenSIPS может быть настроена следующим образом:

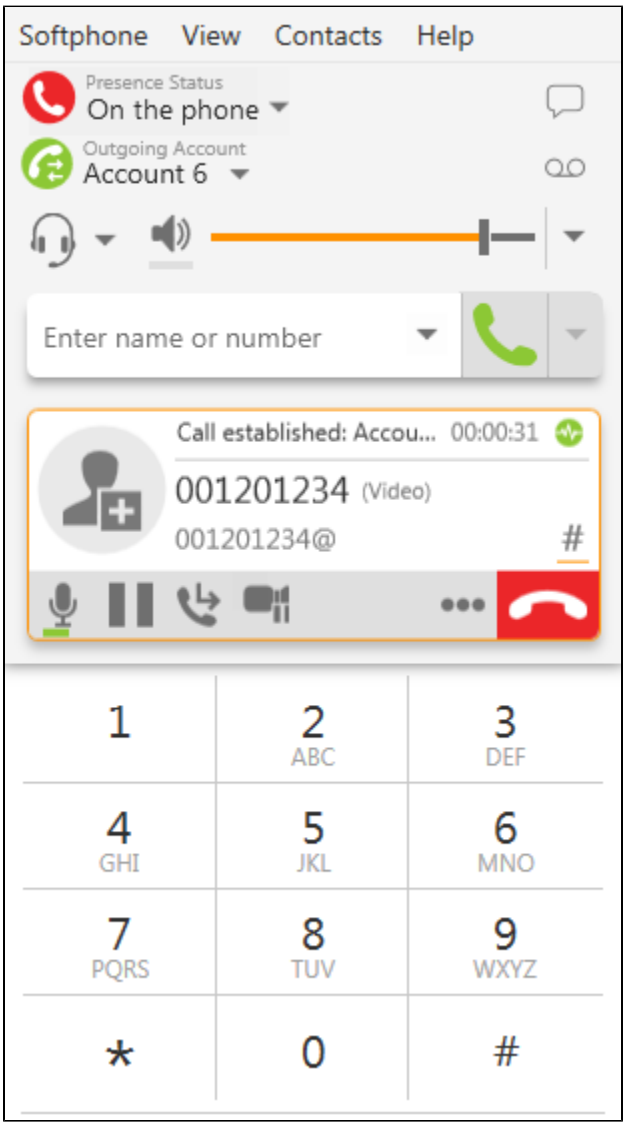
```
route{
        ...
        #WCS Sip trunk routing, 00 prefix + XX for server number (e.g. WCS1 => 01) + X for trunk number
        if ($rU =~ "^00050[0-9]+$") {
            # WCS5 address and port
            rewritehostport("192.168.1.5:40000");
            route(relay);
        }
}
```

Тестирование

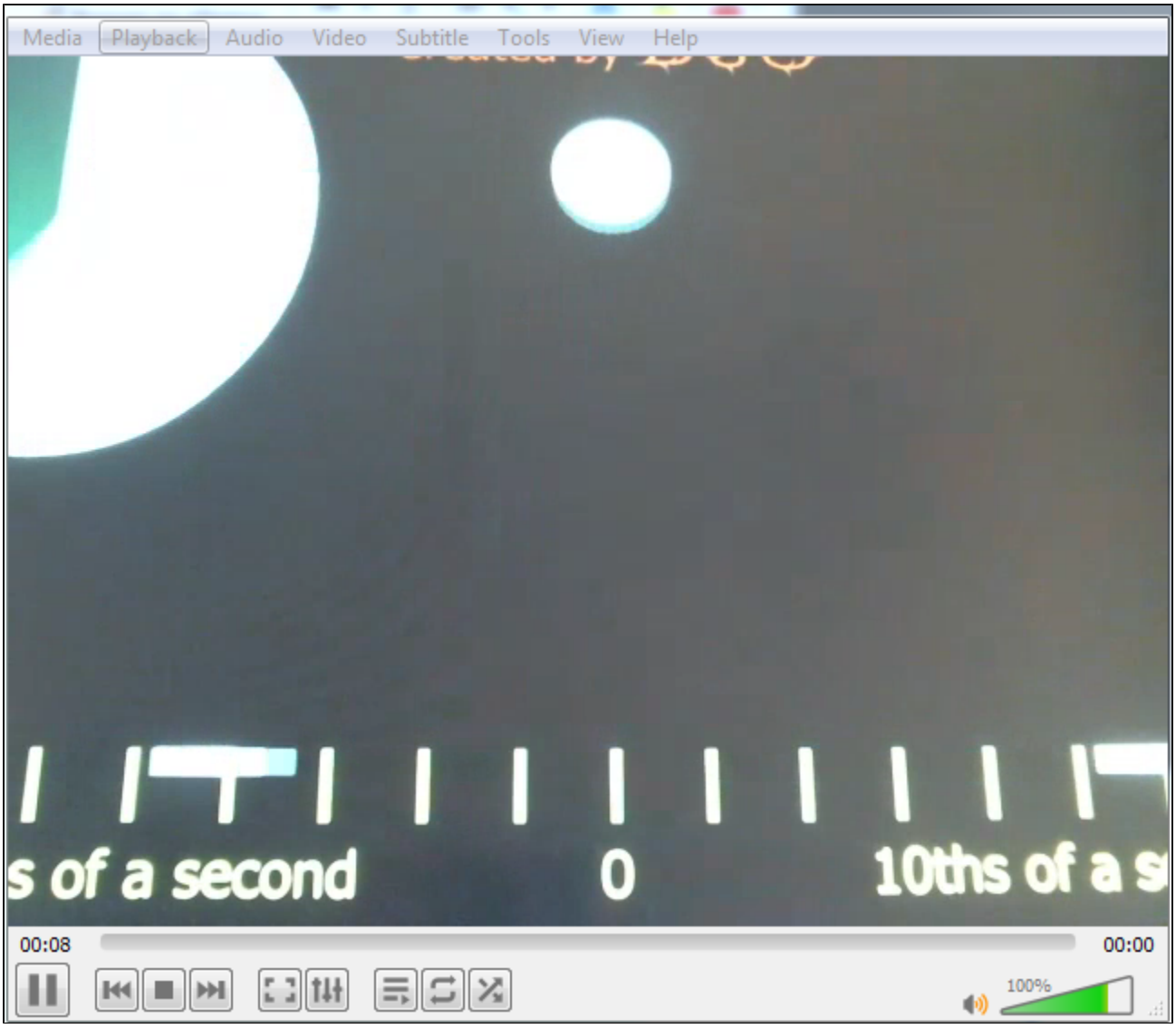
Для теста возьмем:

- WCS сервер
- АТС
- Программный телефон для установки звонка
- RTMP сервер
- VLC для воспроизведения RTMP потока

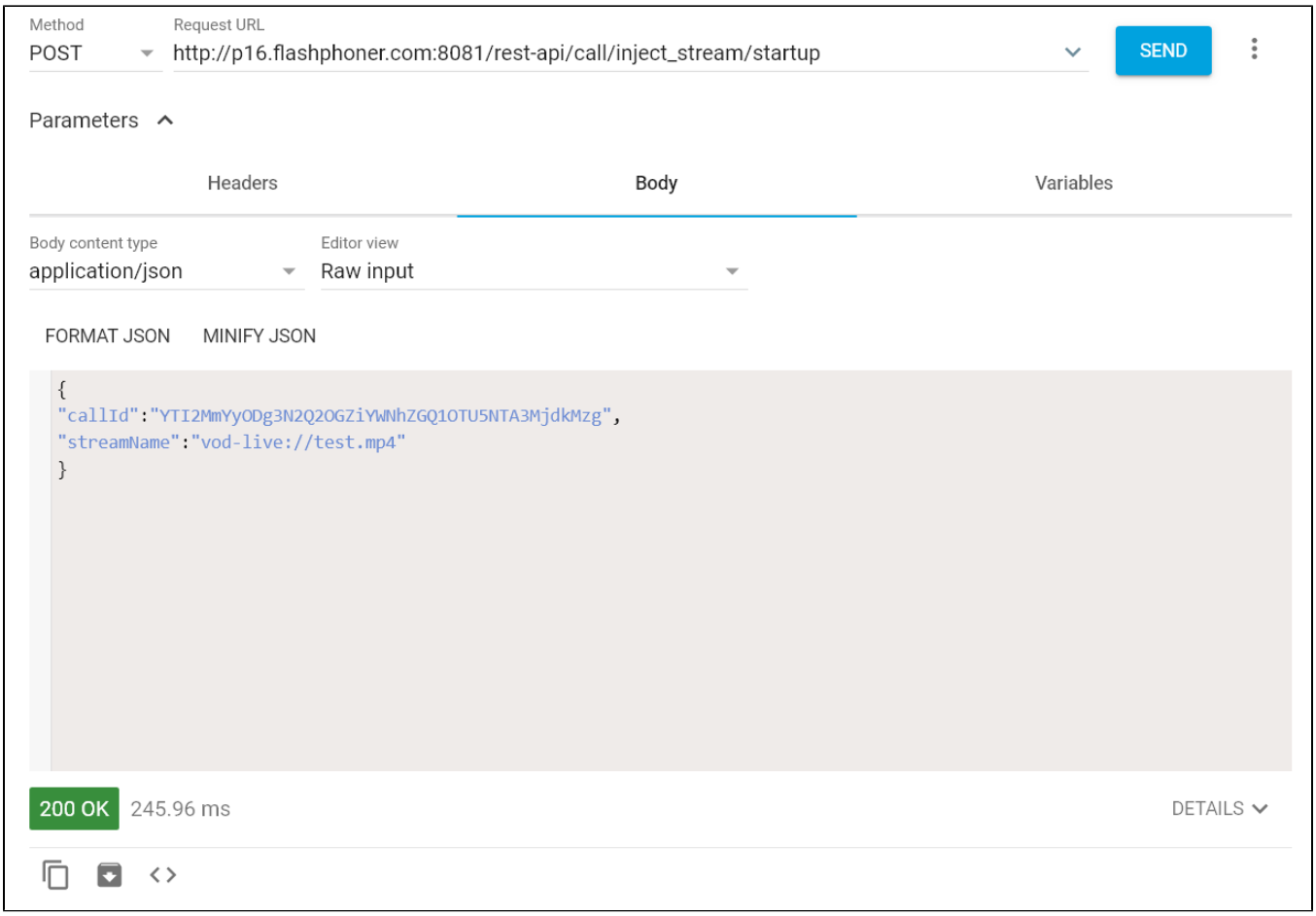
1. Откройте программный телефон, зарегистрируйтесь на АТС, сделайте видеозвонок на номер, определенный при настройке SIP транка на АТС, например 001201234



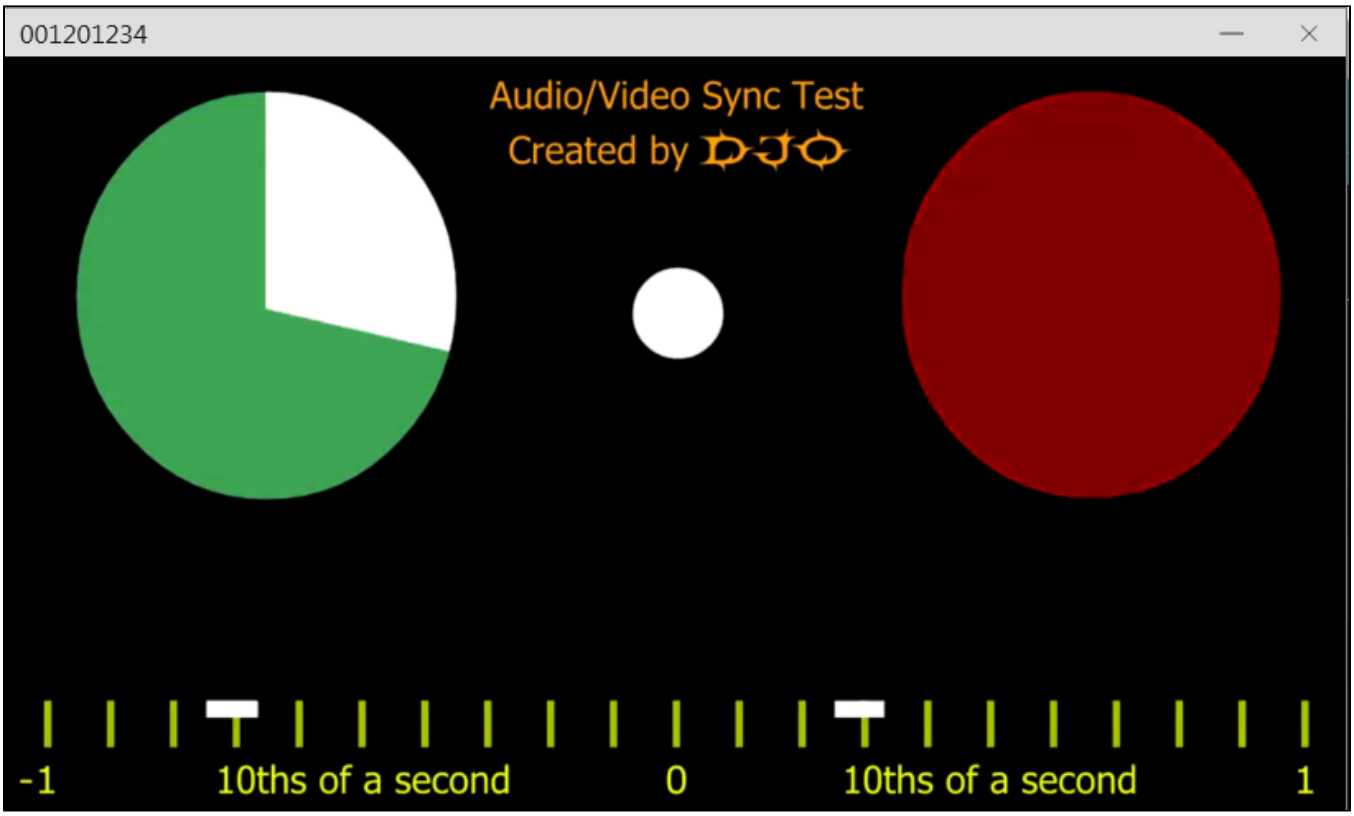
2. Откройте в VLC потокrtmp://rtmp_server:1935/live/rtmp_001201234



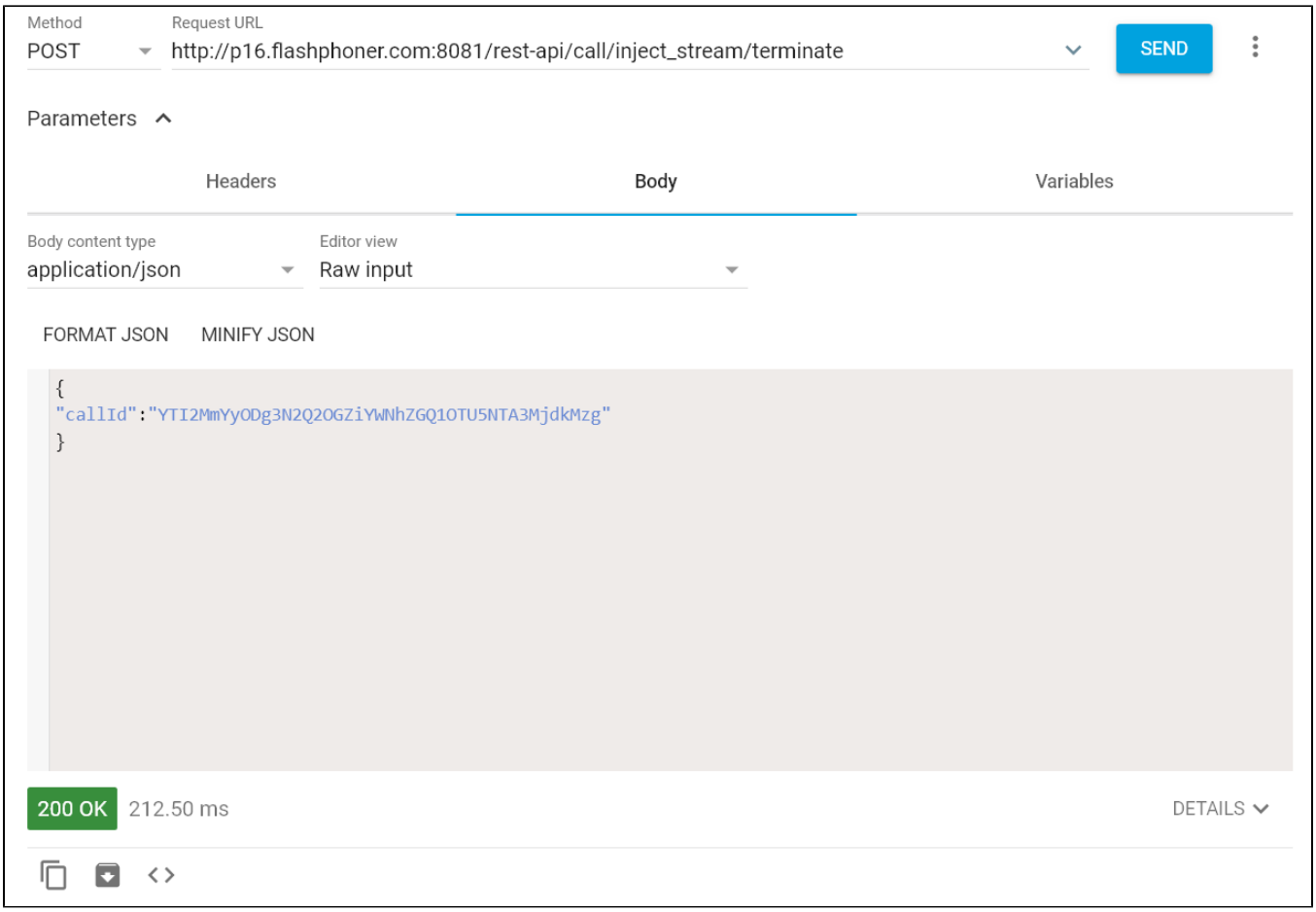
3. Отправьте запрос /call/inject_stream/startup для перенаправления в звонок потока из файла на диске WCS



4. Содержимое файла играет на стороне программного телефона



5. Отправьте запрос /call/inject_stream/terminate, чтобы остановить проигрывание файла



Реализация собственного обработчика входящих SIP сообщений

В некоторых случаях, необходима дополнительная обработка входящих SIP сообщений. Для этого должен быть разработан собственный Java класс, реализующий интерфейс `ISipMessageListener`, который будет перехватывать входящие SIP сообщения и обрабатывать их.

Рассмотрим пример, который добавляет порт в Request URI входящего INVITE запроса, если порт не указан. Исходный текст класса:

customSipMessageListener.java

```
package com.customListener;

import com.flashphoner.sdk.sip.ISipMessageListener;
import com.flashphoner.server.client.IClient;
import gov.nist.javax.sip.address.Authority;
import gov.nist.javax.sip.address.SipUri;
import gov.nist.javax.sip.message.SIPMessage;
import gov.nist.javax.sip.message.SIPRequest;
import gov.nist.javax.sip.stack.MessageChannel;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import javax.sip.message.Request;

public class customSipMessageListener implements ISipMessageListener {

    private static Logger log = LoggerFactory.getLogger("customSipMessageListener");

    @Override
    public void processMessage(SIPMessage sipMessage, IClient client, MessageChannel channel) {
        if (sipMessage instanceof SIPRequest) {
            SIPRequest request = (SIPRequest) sipMessage;
            String method = request.getRequestLine().getMethod();
            if (Request.INVITE.equals(method)) {
                Authority authority = ((SipUri)request.getRequestURI()).getAuthority();
                int port = authority.getPort();
                if (port <= 0) {
                            if (log.isDebugEnabled()) {
                                    log.debug("Inject port " + channel.getPort());
                            }
                    authority.setPort(channel.getPort());
                }
            }
        }
    }
}
```

Создадим на сервере структуру каталогов

```
mkdir -p com/customListener
```

Копируем исходный текст класса в созданный каталог и компилируем его

```
javac -cp "/usr/local/FlashphonerWebCallServer/lib/*" ./com/customListener/customSipMessageListener.java
```

Упаковываем скомпилированный класс в jar файл

```
jar cf customSipMessageListener.jar com/customListener/customSipMessageListener.class
```

Копируем jar файл в каталог сервера

```
cp customSipMessageListener.jar /usr/local/FlashphonerWebCallServer/lib
```

В файлеflashphoner.propertiesнеобходимо указать созданный класс в настройке

```
sip_msg_listener=com.customListener.customSipMessageListener
```

и перезапустить сервер.

Запись потока входящего SIP звонка

Потоки, полученные из входящих SIP-звонков, могут быть записаны на сервере. Для того, чтобы записывать все входящие звонки, необходимо указать следующие настройки в файлеflashphoner.properties:

```
sip_record_stream=true
```

Чтобы записать поток отдельного звонка, необходимо использовать соответствующий REST запрос.

Обратите внимание, что входящие звонки не будут записываться, если активна настройка

```
sip_single_route_only=true
```

Известные проблемы

1. В RTMP потоке, ретранслированном из входящего звонка, может наблюдаться рассинхронизация

Симптомы: рассинхронизация в потоке звонка при воспроизведении с RTMP сервера

Решение:

a) выставить настройку

```
sip_force_rtcp_feedback=true
```

b) свести к минимум либо исключить потери на канале между АТС и сервером