

# Звонок на мобильный телефон через SIP сервер

## Описание

SIP звонок на мобильный телефон является частным случаем [звонков между браузером и SIP-устройством](#), при этом SIP-сервер либо сам предоставляет услуги GSM/PSTN шлюза, либо соединяется с таковым в процессе звонка.

## Поддерживаемые платформы и браузеры

	Chrome	Firefox	Safari	Edge
Windows	✓	✓	✗	✓
Mac OS	✓	✓	✓	✓
Android	✓	✓	✗	✓
iOS	✓	✓	✓	✓

## Поддерживаемые протоколы

- WebRTC
- RTP
- SIP

## Поддерживаемые кодеки

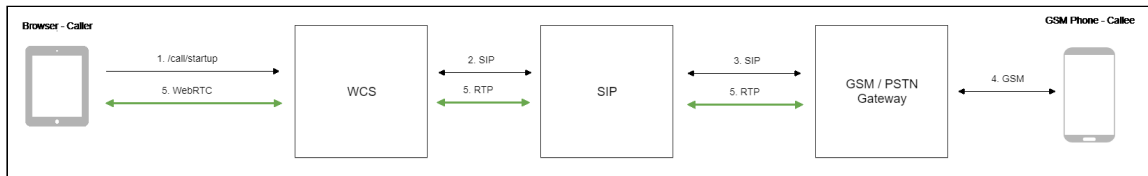
- H.264
- VP8
- G.711
- Speex
- G.729
- Opus

## Поддерживаемые SIP функции

- DTMF
- Удержание звонка
- Перевод звонка

SIP функции управляются при помощи WebSDK.

## Схема работы



1. Браузер начинает звонок
2. WCS соединяется с SIP-сервером
3. SIP-сервер соединяется с GSM/PSTN шлюзом
4. GSM/PSTN шлюз соединяется с телефоном
5. Браузер и телефон обмениваются аудиопотоками

## Краткое руководство по тестированию

1. Для тестирования используем:
2. SIP-аккаунт;
3. веб-приложение [Phone UI](#) для совершения звонка;
4. мобильный телефон для ответа на звонок.
5. Откройте веб-приложение Phone UI. Нажмите **Log in** и введите данные SIP-аккаунта:

# Phone

Log in

Enter your number here

Video call

Voice call

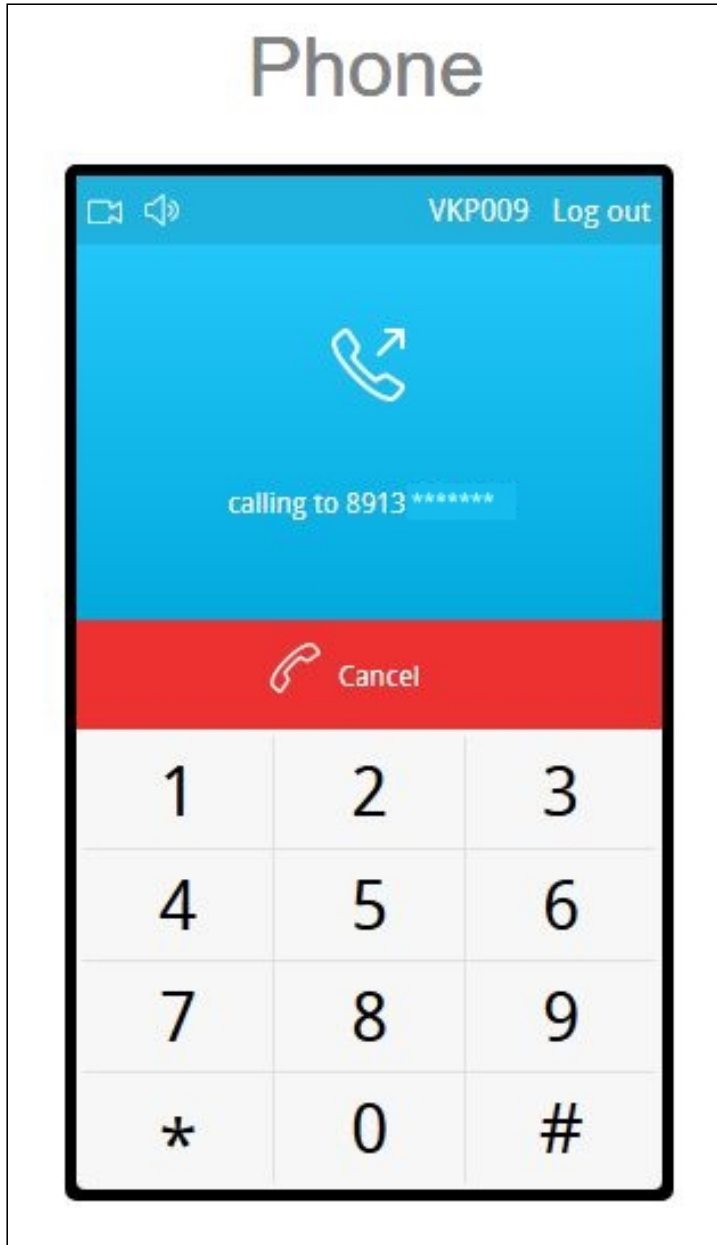
1	2	3
4	5	6
7	8	9
*	0	#

login: VKP009  
auth. name: VKP009  
password: .....  
domain: yoursip.net  
outbound proxy: yoursip.net  
port: 5060

log in

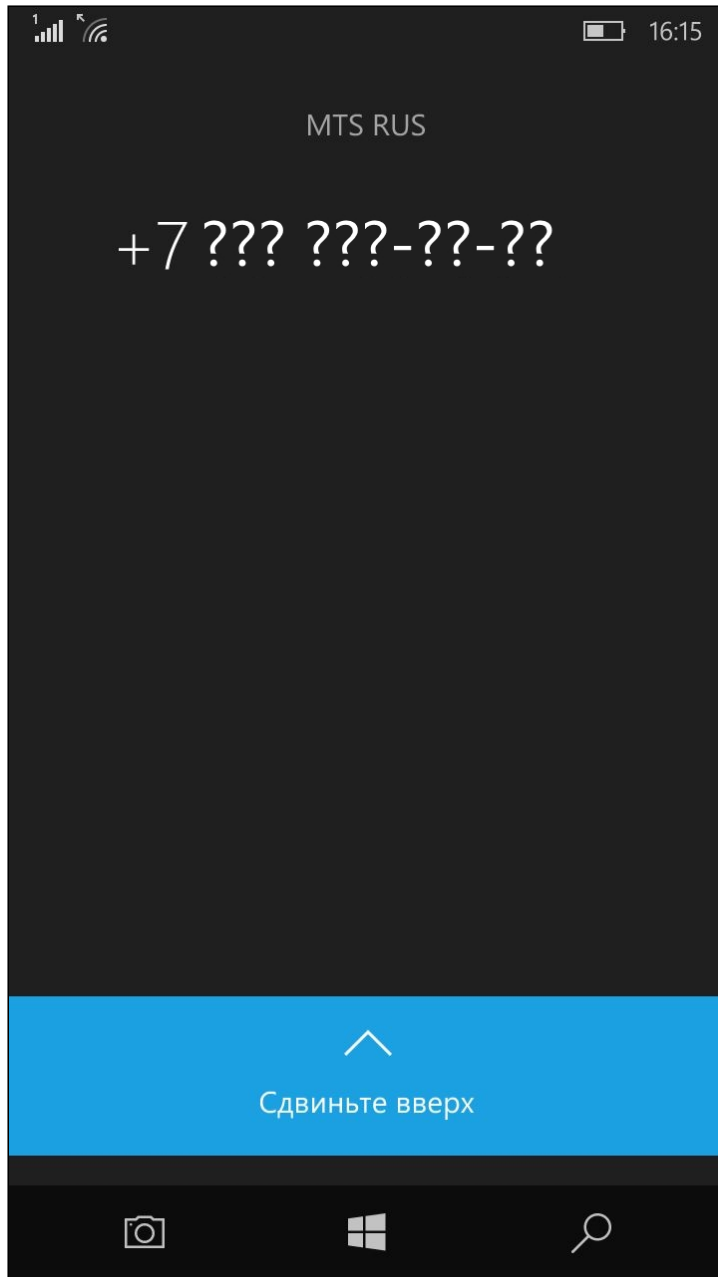
cancel

6. Введите номер мобильного телефона и нажмите **Voice call**. Начнется дозвон:

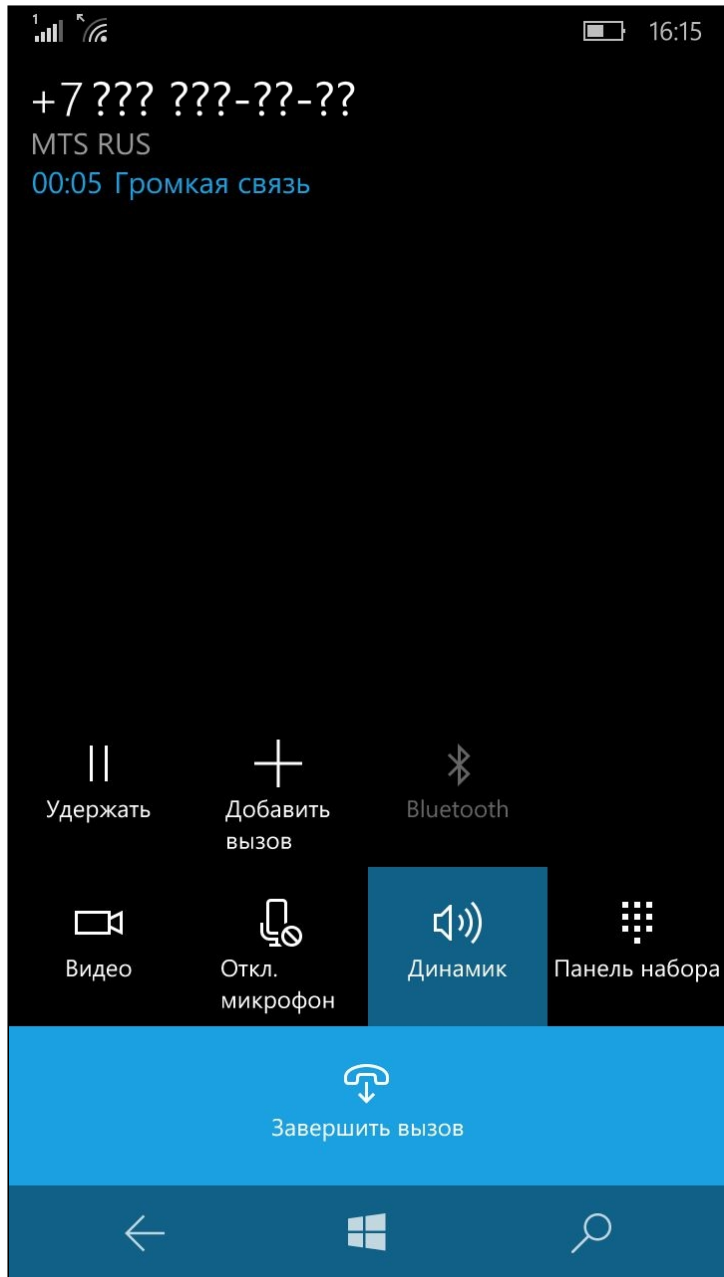




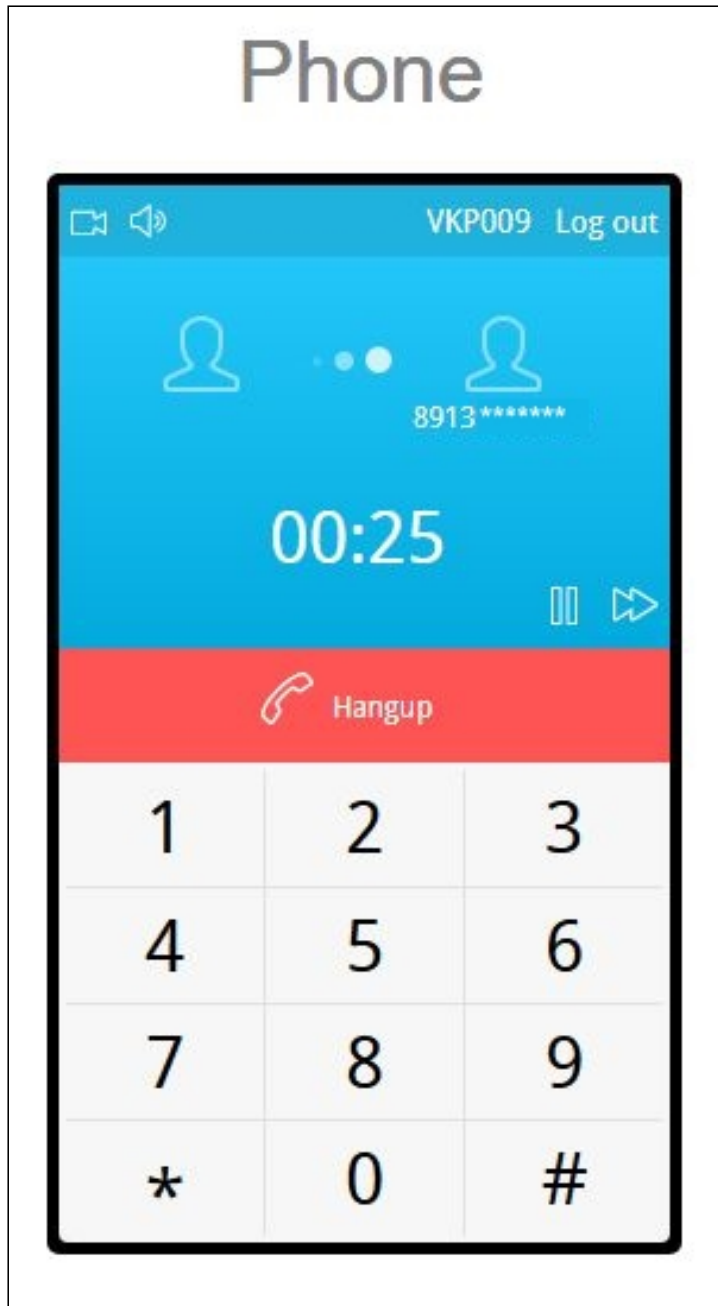
7. На экране мобильного телефона отображается входящий вызов:



8. Примите звонок на мобильном телефоне:



9. В браузере также отображается установленное соединение:



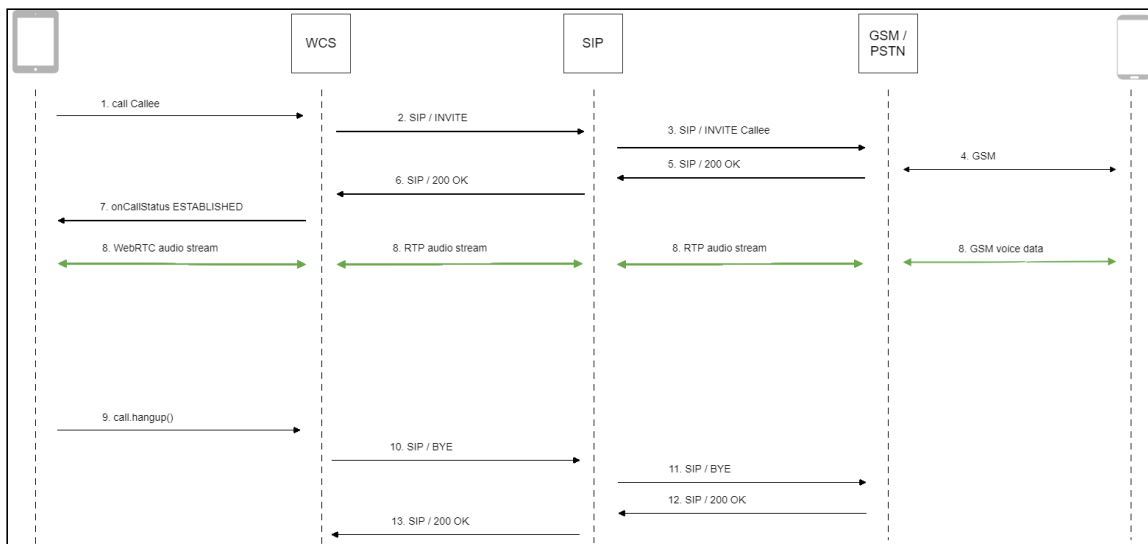
10. Для завершения звонка нажмите кнопку `Hangup`

## Последовательность выполнения операций

Ниже описана последовательность вызовов при использовании примера Phone для создания звонка

[Phone.html](#)

[Phone.js](#)



## 1. Создание звонка

`Session.createCall()`, `Call.call()` [code](#)

```

var outCall = this.session.createCall({
  callee: callee,
  visibleName: this.sipOptions.login,
  localVideoDisplay: this.localVideo,
  remoteVideoDisplay: this.remoteVideo,
  constraints: constraints
  ...
});

outCall.call();
  
```

2. Отправка `SIP INVITE` на SIP-сервер

3. Отправка `SIP INVITE` на GSM/PSTN шлюз

4. Установка соединения с мобильным терминалом

5. Получение подтверждения от GSM/PSTN шлюза

6. Получение подтверждения от SIP-сервера

7. Получение от сервера события, подтверждающего успешное соединение

`CALL_STATUS.ESTABLISHED` [code](#)

```

var outCall = this.session.createCall({
  ...
}).on(CALL_STATUS.ESTABLISHED, function(call){
  me.callStatusListener(call);
  ...
});
  
```

8. Стороны звонка обмениваются аудиопотоком

## 9. Завершение звонка

`Call.hangup()` [code](#)

```
Phone.prototype.hangup = function () {
    trace("Phone - hangup " + this.currentCall.id() + " status " +
this.currentCall.status());
    this.hideFlashAccess();
    if (this.currentCall.status() == CALL_STATUS.PENDING) {
        this.callStatusListener(this.currentCall);
    } else {
        this.currentCall.hangup();
    }
    this.flashphonerListener.onHangup();
};
```

10. Отправка `SIP BYE` на SIP-сервер

11. Отправка `SIP BYE` на GSM/PSTN шлюз

12. Получение подтверждения от GSM/PSTN шлюза

13. Получение подтверждения от SIP-сервера