

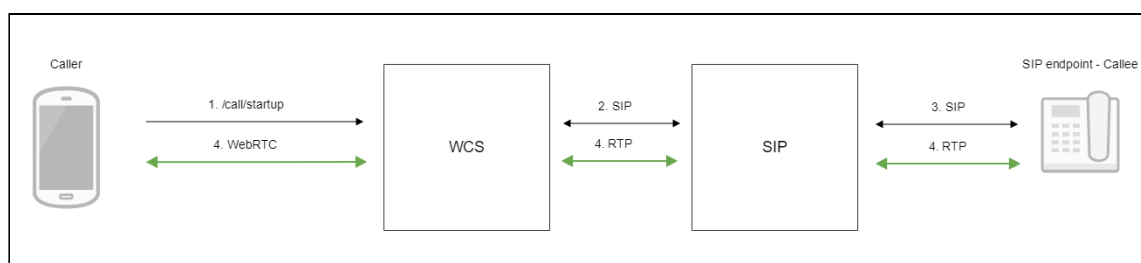
Звонки с использованием Android SDK

Описание

SIP звонки на устройствах Android возможны как [из браузера](#), так и с использованием [Android SDK](#).

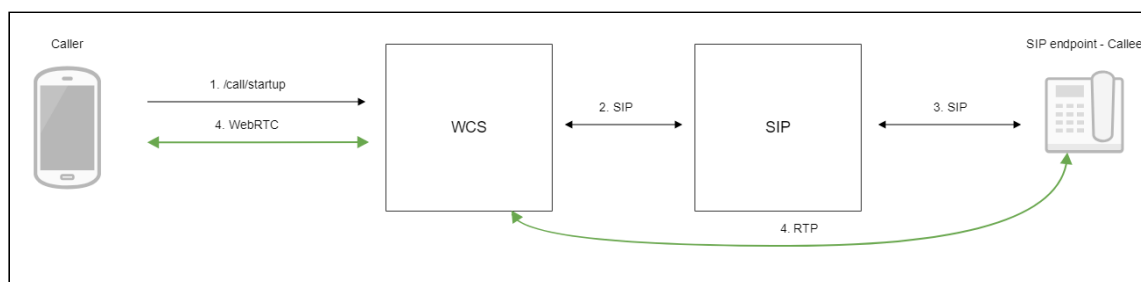
Схема работы

1. SIP-сервер как прокси-сервер для передачи вызовов и RTP медиа



1. Устройство Android начинает звонок
2. WCS соединяется с SIP-сервером
3. SIP-сервер соединяется с SIP-устройством, принимающим звонок
4. Устройство Android и SIP-устройство обмениваются аудио- и видеопотоками

2. SIP-сервер только как сервер для передачи вызовов



1. Устройство Android начинает звонок
2. WCS соединяется с SIP-сервером
3. SIP-сервер соединяется с SIP-устройством, принимающим звонок
4. Устройство Android и SIP-устройство обмениваются аудио- и видеопотоками

Тестирование

Исходящий звонок с Android на SIP-устройство

1. Для тестирования используем:
2. два SIP-аккаунта;
3. приложение [Phone](#) для совершения звонка;
4. программный телефон для ответа на звонок.
5. Установите на Android-устройство приложение [Phone](#). Запустите приложение, введите URL WCS-сервера для подключения к нему по Websocket и данные звонящего SIP-аккаунта:

Phone-min

WCS Url
wss://192.168.0.1:8443

Sip Login
1000

Sip Password
• • • •

Sip Domain
192.168.0.1

Sip Port
5060

☒ Register required

CONNECT

Callee

6. Запустите программный телефон, введите данные SIP-аккаунта, принимающего звонок:

Учетная запись | Голосовая почта | Топология | Присутствие | Транспорт | Дополнительно

Имя учетной записи: Account 2

Протокол: SIP

Разрешить использование этой учетной записи для

- ☒ Вызов
- ☒ Чат / присутствие

Сведения о пользователе

* Идентификатор пользователя: 10005

* Домен: yoursip.domain

Пароль: ●●●●●

Отображаемое имя: 10005

Имя авторизации: 10005

Прокси-сервер домена

☒ Зарегистрироваться в домене и принимать вызовы

Отправлять исходящие через:

☒ Домен

☐ Прокси-сервер Адрес:

7. Нажмите в приложении кнопку **Connect**, будет установлено соединение с сервером. Затем введите идентификатор SIP-аккаунта, принимающего звонок, и

Callee

10005

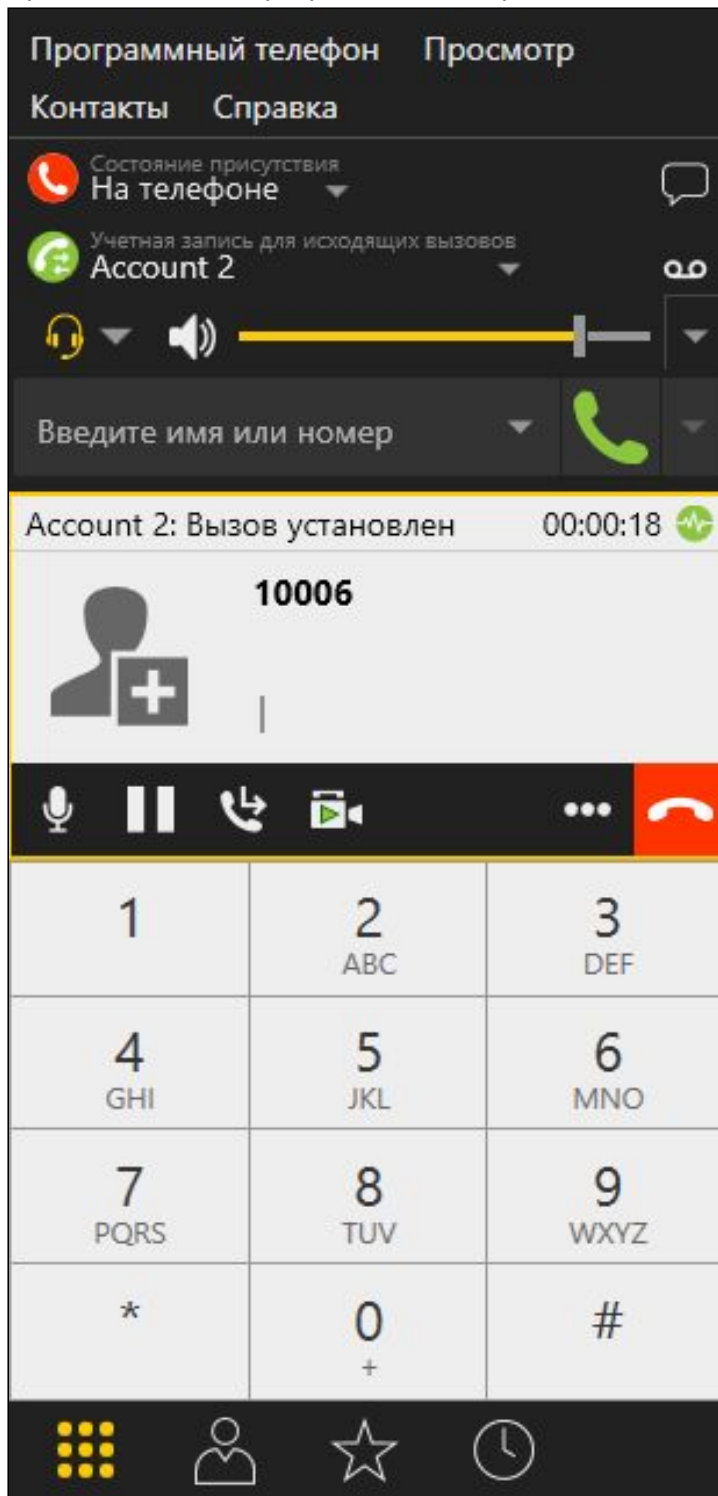
ESTABLISHED

HANGUP

HOLD

нажмите кнопку **Call**:

8. Примите звонок в программном телефоне, нажав кнопку ответа на звонок:



9. Для завершения звонка нажмите кнопку **Hangup** в приложении, либо кнопку завершения звонка в программном телефоне.

Прием входящего звонка с SIP-устройства на Android

1. Для тестирования используем:

2. два SIP-аккаунта;
3. программный телефон для совершения звонка;
4. приложение [Phone](#) для ответа на звонок.
5. Установите на Android-устройство приложение [Phone](#). Запустите приложение, введите URL WCS-сервера для подключения к нему по Websocket и данные SIP-аккаунта, принимающего звонок:

Phone-min

WCS Url
wss://192.168.0.1:8443

Sip Login
1000

Sip Password
• • • •

Sip Domain
192.168.0.1

Sip Port
5060

☒ Register required

CONNECT

Callee

Нажмите в приложении кнопку **Connect**, будет установлено соединение с сервером

6. Запустите программный телефон, введите данные звонящего SIP-аккаунта:

Учетная запись Голосовая почта Топология Присутствие Транспорт Дополнительно

Имя учетной записи: Account 2

Протокол: SIP

Разрешить использование этой учетной записи для

☒ Вызов

☒ Чат / присутствие

Сведения о пользователе

* Идентификатор пользователя: 10005

* Домен: yoursip.domain

Пароль: ●●●●

Отображаемое имя: 10005

Имя авторизации: 10005

Прокси-сервер домена

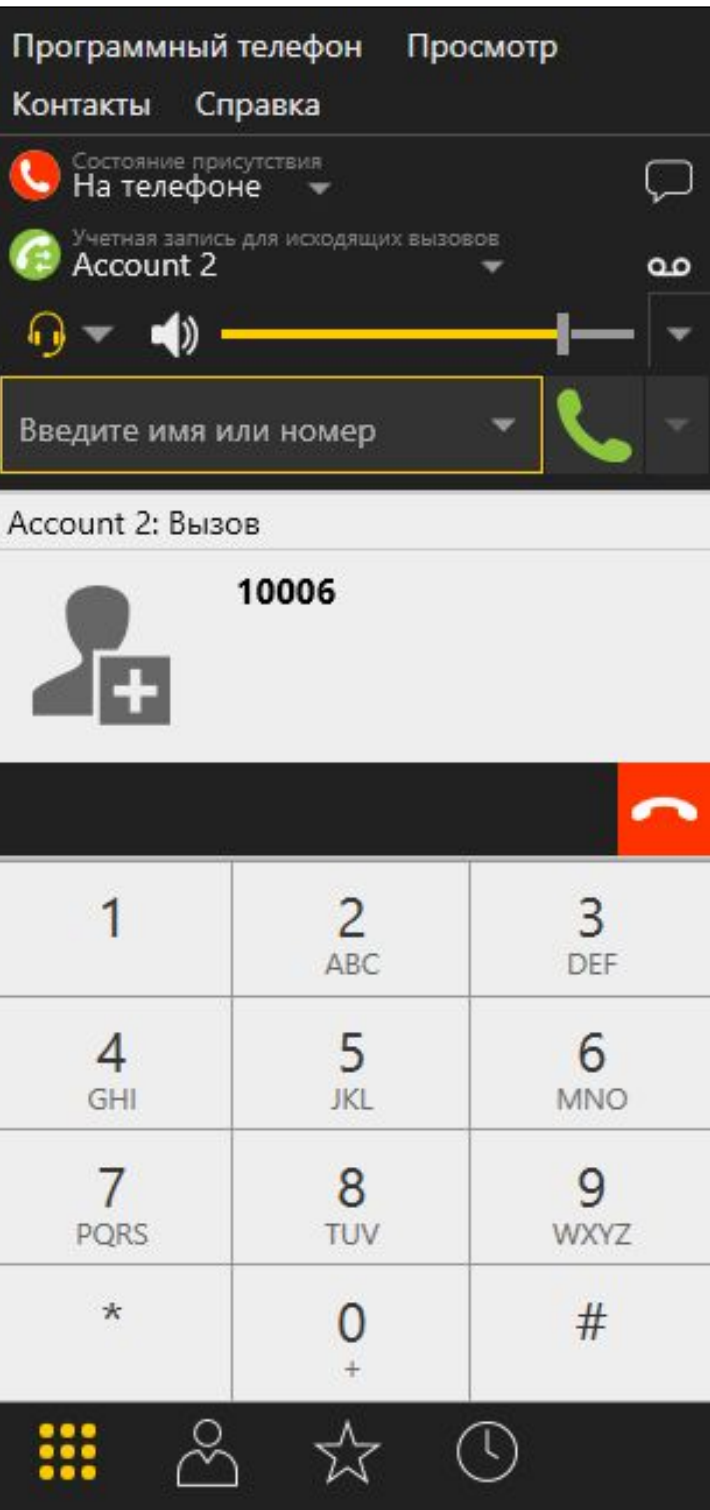
☒ Зарегистрироваться в домене и принимать вызовы

Отправлять исходящие через:

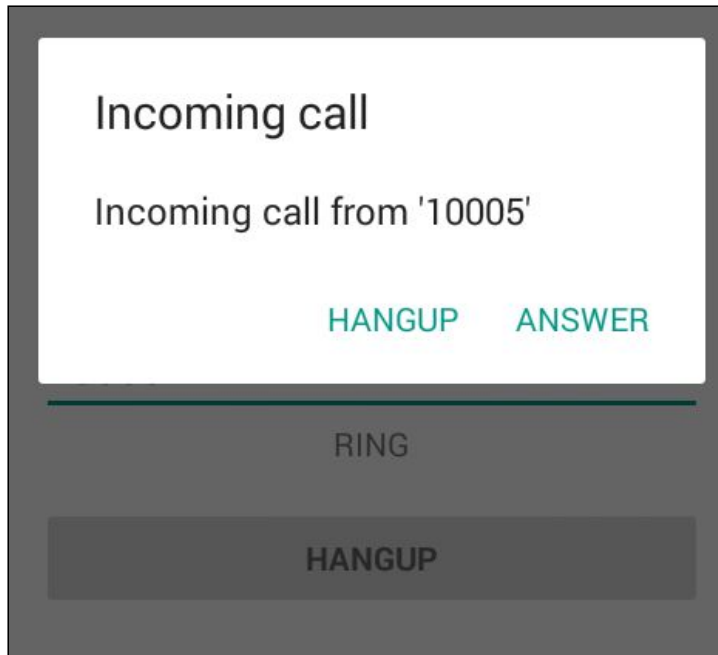
☒ Домен

☐ Прокси-сервер Адрес:

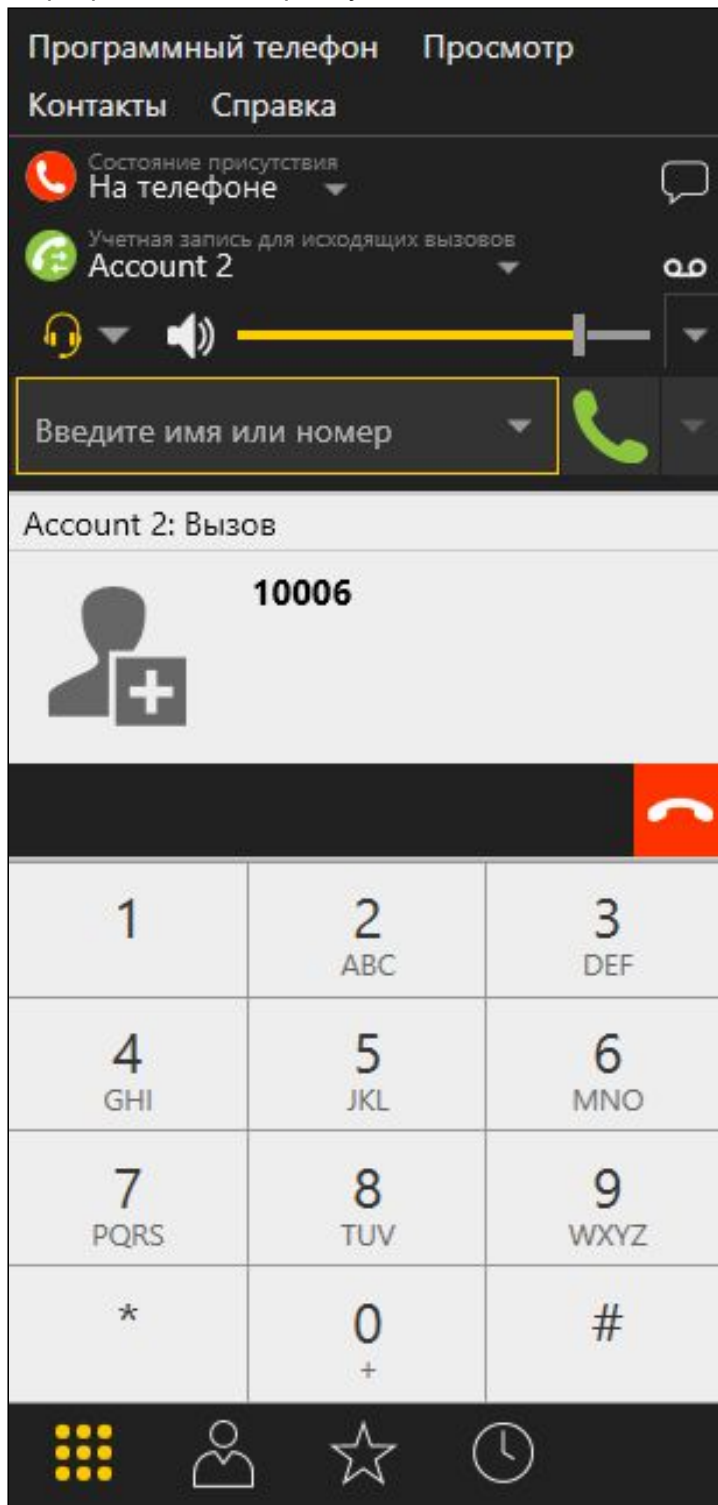
7. В программном телефоне введите идентификатор SIP-аккаунта, принимающего звонок, и нажмите кнопку вызова:



8. Примите звонок в приложении, нажав **Answer**:



9. В программном телефоне убедитесь, что звонок начался:

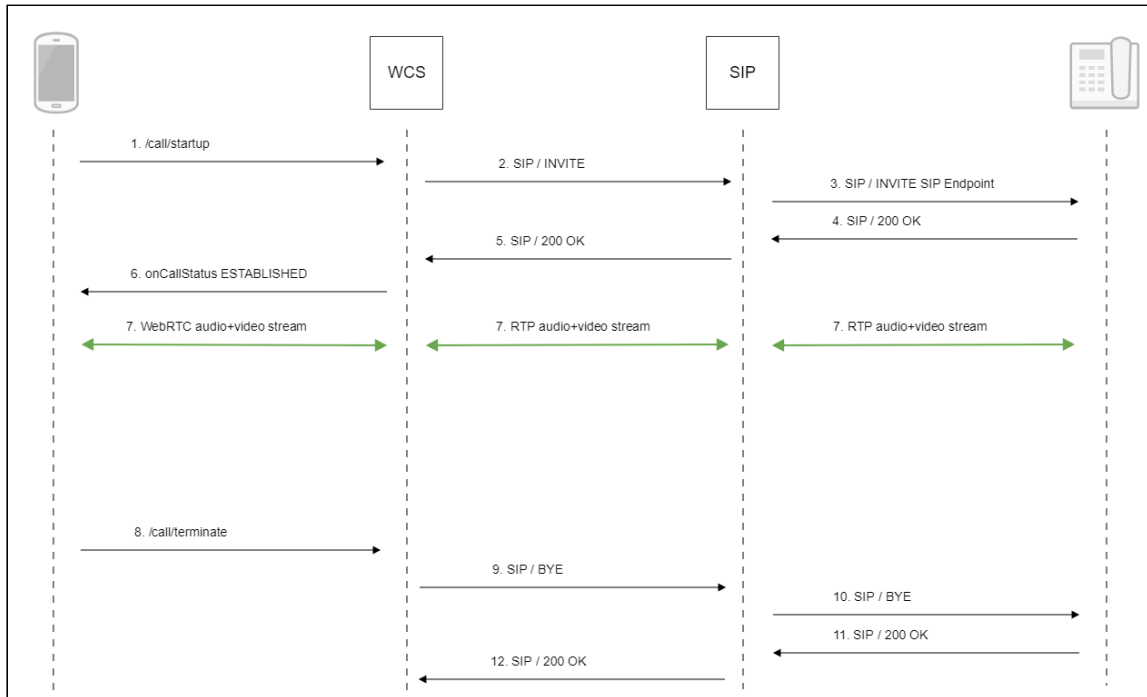


10. Для завершения звонка нажмите кнопку **Hangup** в приложении, либо кнопку завершения звонка в программном телефоне.

Последовательность выполнения операций

Ниже описана последовательность вызовов при использовании примера Phone-min для создания звонка

PhoneMinActivity.java



1. Создание звонка:

`Session.createCall()`, `Call.call()` code

```
CallOptions callOptions = new
CallOptions(mCalleeView.getText().toString());
AudioConstraints audioConstraints =
callOptions.getConstraints().getAudioConstraints();
MediaConstraints mediaConstraints =
audioConstraints.getMediaConstraints();
...
call = session.createCall(callOptions);
call.on(callStatusEvent);
/**
 * Make the outgoing call
 */
call.call();
Log.i(TAG, "Permission has been granted by user");
```

2. Отправка `SIP INVITE` на SIP сервер
3. Отправка `SIP INVITE` на SIP устройство
4. Получение подтверждения от SIP-устройства
5. Получение подтверждения от SIP-сервера
6. Получение от сервера события, подтверждающего успешное соединение.

7. Стороны звонка обмениваются аудио- и видеопотоками

8. Завершение звонка

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

```
if (mCallButton.getTag() == null ||
Integer.valueOf(R.string.action_call).equals(mCallButton.getTag())) {
    if ("".equals(mCalleeView.getText().toString())) {
        return;
    }
    ...
} else {
    mCallButton.setEnabled(false);
    call.hangup();
    call = null;
}
```

9. Отправка `SIP BYE` на SIP-сервер

10. Отправка `SIP BYE` на SIP-устройство

11. Получение подтверждения от SIP-устройства

12. Получение подтверждения от SIP-сервера

Известные проблемы

1. Невозможно совершить SIP-звонок, если поля `SIP Login`, `SIP Authentication name` содержат недопустимые символы



Симптомы

Звонок не совершается, зависает в статусе `PENDING`

✓ Решение

Согласно [RFC3261](#), `SIP Login` и `SIP Authentication name` не должны содержать незранированных 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'
```