

# Звонки с использованием iOS SDK

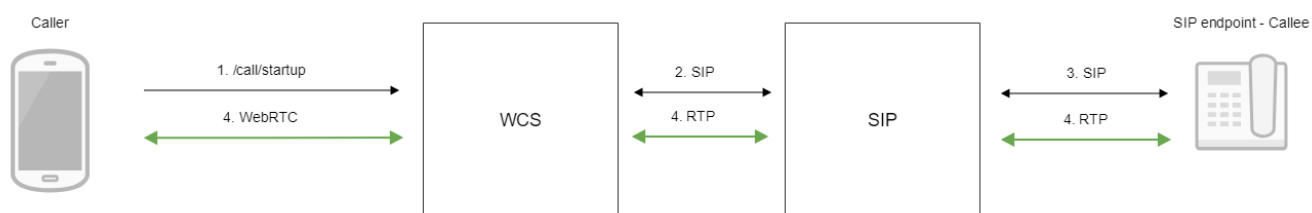
- [Описание](#)
  - [Схема работы](#)
- [Исходящий звонок с iOS на SIP-устройство](#)
- [Прием входящего звонка с SIP-устройства на iOS](#)
- [Последовательность выполнения операций \(Call Flow\)](#)
- [Известные проблемы](#)

## Описание

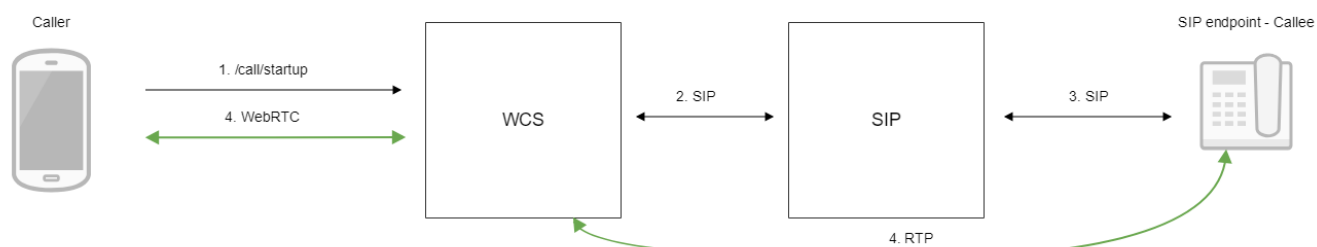
SIP звонки на устройствах iOS возможны как [из браузера](#), так и с использованием [Описания](#).

## Схема работы

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



### 2: SIP-сервер только как сервер для передачи вызовов



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

## Исходящий звонок с iOS на SIP-устройство

### 1. Для тестирования используем:

- два SIP-аккаунта;
- приложение [Phone](#) для совершения звонка;
- программный телефон для ответа на звонок.

2. Установите на iOS-устройство приложение [Phone](#). Запустите приложение, введите URL WCS-сервера для подключения к нему по Secure Websocket и данные звонящего SIP-аккаунта:

No SIM 16:50 40 %

wss://wcs5-eu.flashphoner.com:8443

Sip Login

1000

Sip Auth Name

1000

Sip Password

1234

Sip Domain

192.168.0.1

Sip Outbound Proxy

192.168.0.1

Sip Port

5060

Sip Register Required

☒

NO STATUS

3. 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:

4. Нажмите в приложении кнопку Connect, будет установлено соединение с сервером. Затем введите идентификатор SIP-аккаунта, принимающего звонок, и нажмите кнопку Call:

No SIM

87.226.225.56

07:37

86 %

Sip Port

5060

Sip Register Required

REGISTERED

DISCONNECT

Invite Parameters

Callee

10005

RING

HANGUP

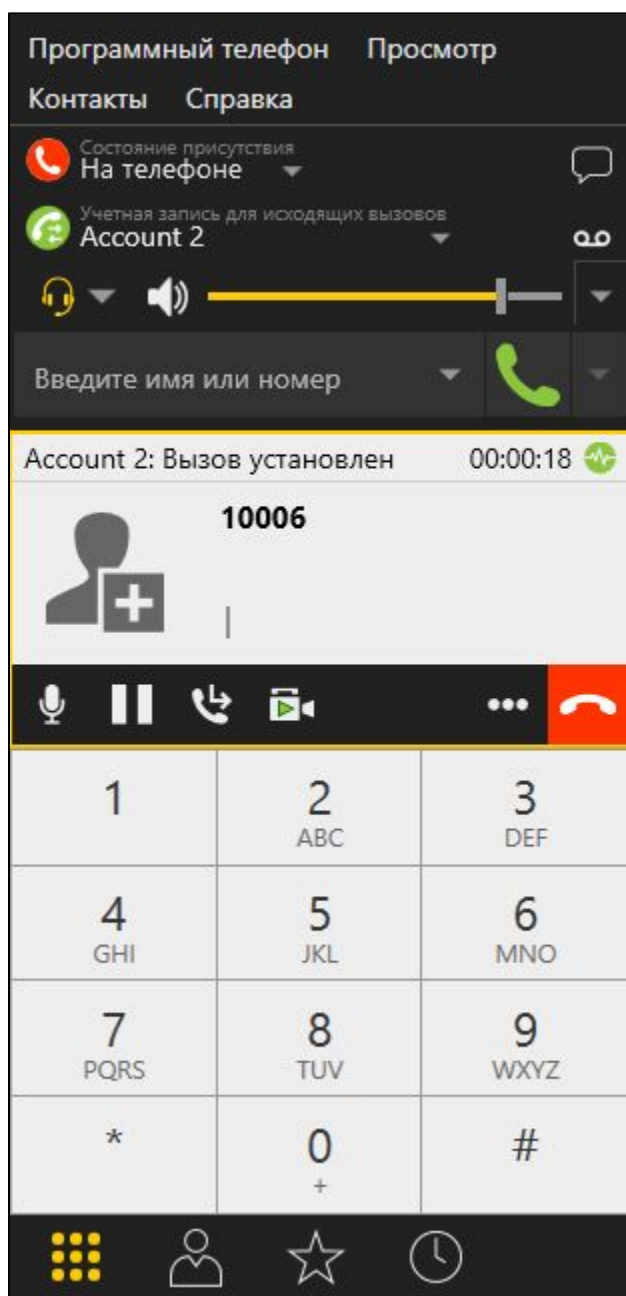
HOLD

DTMF

1

DTMF

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



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

## Прием входящего звонка с SIP-устройства на iOS

1. Для тестирования используем:

- два SIP-аккаунта;
- программный телефон для совершения звонка;
- приложение [Phone](#) для ответа на звонок.

2. Установите на iOS-устройство приложение [Phone](#). Запустите приложение, введите URL WCS-сервера для подключения к нему по Secure Websocket и данные SIP-аккаунта, принимающего звонок:

No SIM



16:50

40 %



wss://wcs5-eu.flashphoner.com:8443

Sip Login

1000

Sip Auth Name

1000

Sip Password

1234

Sip Domain

192.168.0.1

Sip Outbound Proxy

192.168.0.1

Sip Port

5060

Sip Register Required



NO STATUS

3. 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

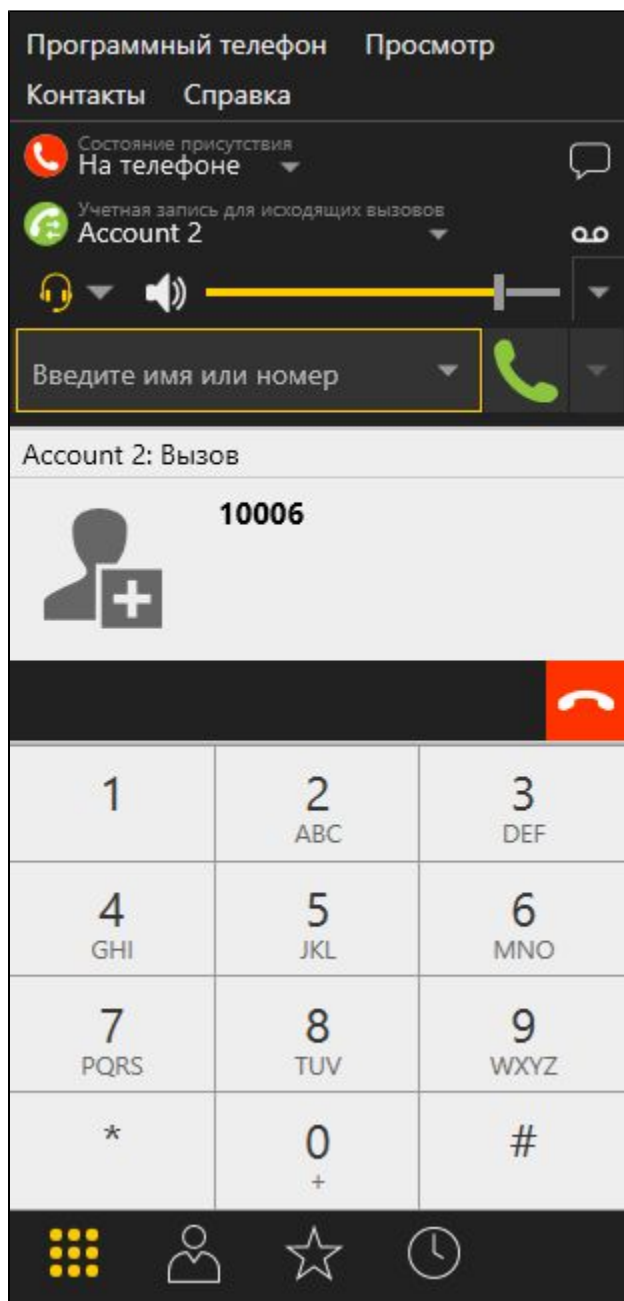
☒ Зарегистрироваться в домене и принимать вызовы

Отправлять исходящие через:

☒ Домен

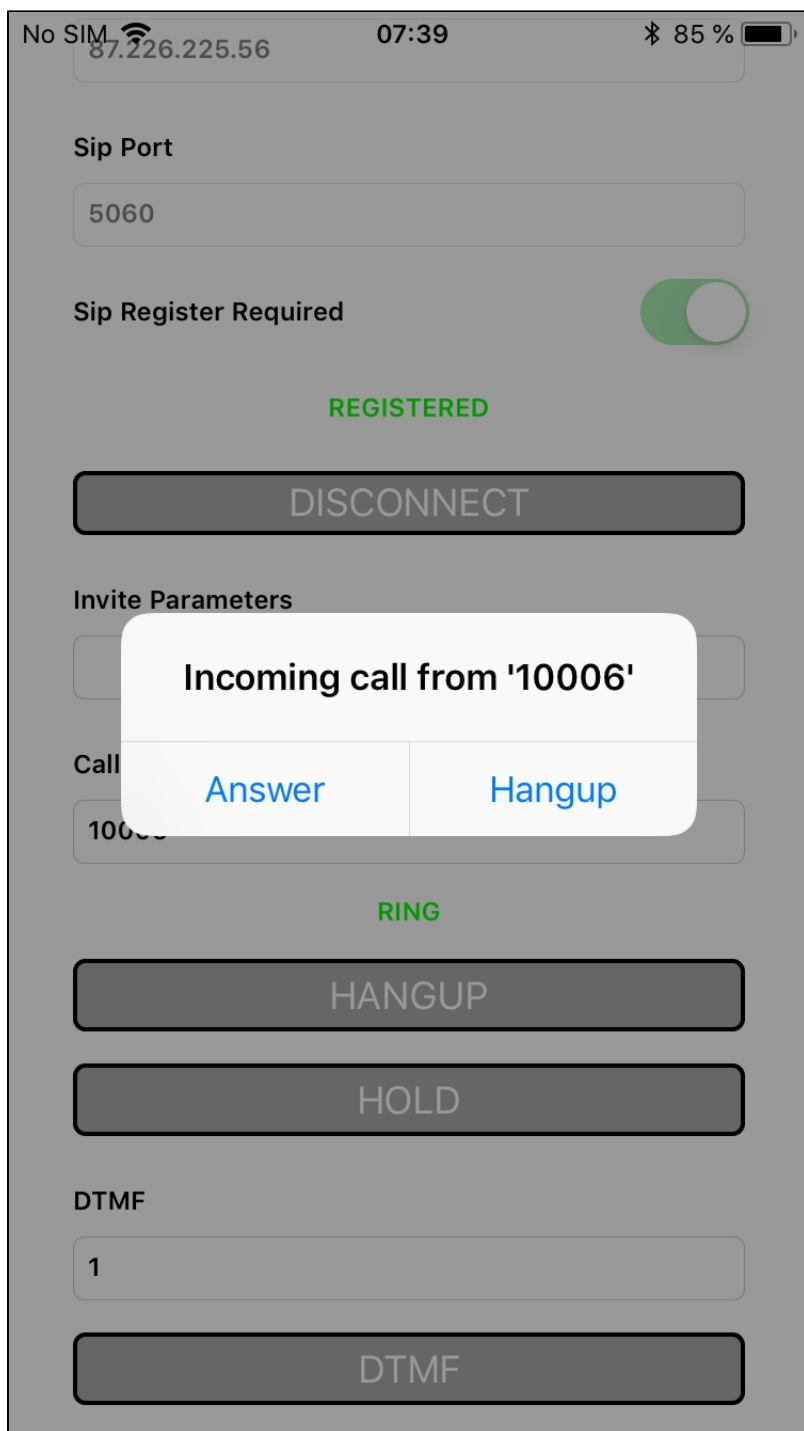
☐ Прокси-сервер Адрес:

4. Нажмите в приложении кнопку "Connect", будет установлено соединение с сервером. В программном телефоне введите идентификатор SIP-аккаунта, принимающего звонок, и нажмите кнопку вызова:

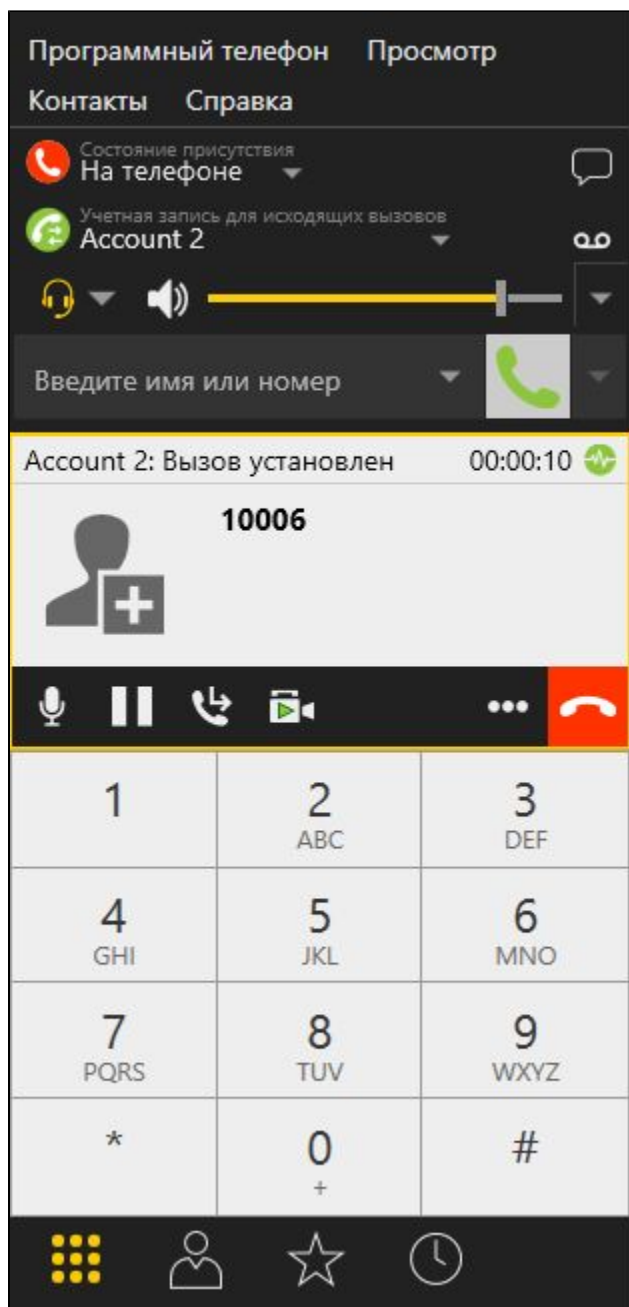


5. Примите звонок в приложении, нажав Answer:





6. В программном телефоне убедитесь, что звонок начался:

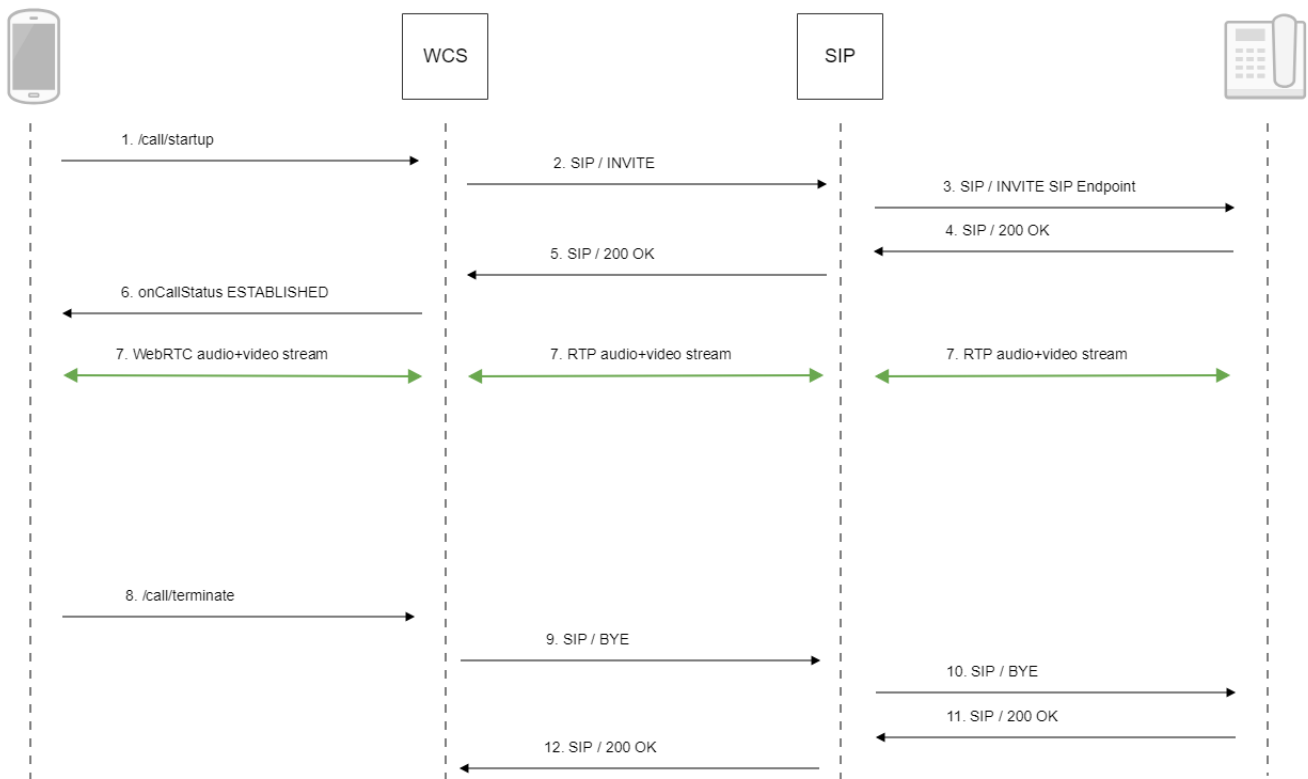


7. Для завершения звонка нажмите кнопку Hangup в приложении, либо кнопку завершения звонка в программном телефоне.

## Последовательность выполнения операций (Call Flow)

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

Класс для основного вида приложения: ViewController (заголовочный файл [ViewController.h](#); файл имплементации [ViewController.m](#))



## 1. Создание звонка:

FPWCSApi2Session createCall, FPWCSApi2Call callcode

При создании звонка в метод createCall передаются параметры:

- имя вызываемого SIP-аккаунта
- дополнительные параметры SIP INVITE запроса, введенные пользователем

```

- (FPWCSApi2Call *)call {
    FPWCSApi2Session *session = [FPWCSApi2 getSessions][0];
    FPWCSApi2CallOptions *options = [[FPWCSApi2CallOptions alloc] init];
    NSString *parameters = _inviteParameters.input.text;
    if (parameters && [parameters length] > 0) {
        NSError* err = nil;
        parameters = [parameters stringByReplacingOccurrencesOfString:@"\"" withString:@"\""];
        NSMutableDictionary *dictionary = [NSJSONSerialization JSONObjectWithData:[parameters dataUsingEncoding:NSUTF8StringEncoding] options:0 error:&err];
        if (err) {
            NSLog(@"Error converting JSON Invite parameters to dictionary %@, JSON %@", err, parameters);
        } else {
            options.inviteParameters = dictionary;
        }
    }
    options.callee = _callee.input.text;
    //used for only recv audio
    // options.localConstraints = [[FPWCSApi2MediaConstraints alloc] initWithAudio:NO video:NO];
    // options.remoteConstraints = [[FPWCSApi2MediaConstraints alloc] initWithAudio:YES video:NO];
    NSError *error;
    call = [session createCall:options error:&error];
    ...
    [call call];
    return call;
}
  
```

## 2. Установка соединения с SIP-сервером

3. Установка соединения с адресатом
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

FPWCSSApi2Call hangupcode

```
- (void)callButton:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"HANGUP"]) {
        if ([FPWCSSApi2 getSessions].count) {
            [call hangup];
        } else {
            [self toCallState];
        }
        ...
    }
}
```

9. Отправка команды на SIP-сервер
10. Отправка команды на SIP-устройство
11. Получение подтверждения от SIP-устройства
12. Получение подтверждения от SIP-сервера

## Известные проблемы

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

Симптомы: звонок не совершается, зависает в статусе PENDING

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'
```