

# iOS Phone Video

## Пример iOS-приложения для видеозвонков




На скриншоте ниже представлен пример интерфейса до совершения звонка.

Интерфейс приложения такой же, как в примере [Phone](#), за исключением того, что воспроизводятся два видео

- слева отображается видео с камеры данного пользователя
- справа воспроизводится видео от другой стороны

No SIM 

16:53

 42%  

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

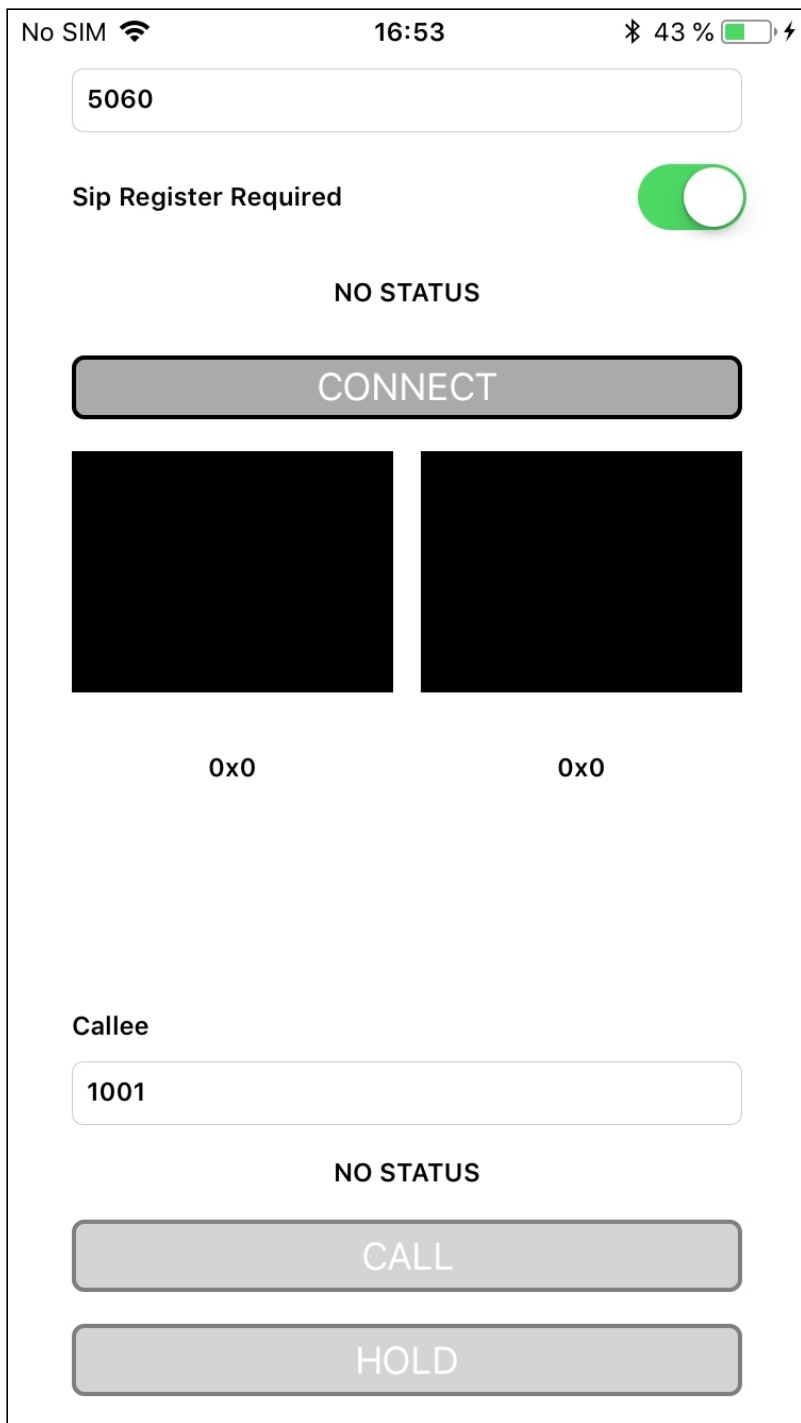
Sip Port

5060

Sip Register Required



NO STATUS



## Работа с кодом примера

Для разбора кода возьмем версию примера PhoneMinVideo, которая доступна [здесь](#).

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

### 1. Импорт API

code

```
#import <FPWCSApi2/FPWCSApi2.h>
```

## 2. Подключение к серверу

`FPWCSApi2.createSession`, `FPWCSApi2Session.connect` [code](#)

В параметрах сессии указываются:

- URL WCS-сервера
- параметры SIP-аккаунта для совершения исходящих и приема входящих звонков
- имя серверного приложения defaultApp

```
FPWCSApi2SessionOptions *options = [[FPWCSApi2SessionOptions alloc] init];
options.urlServer = _connectUrl.text;
options.sipRegisterRequired = _sipRegRequired.control.isOn;
options.sipLogin = _sipLogin.input.text;
options.sipAuthenticationName = _sipAuthName.input.text;
options.sipPassword = _sipPassword.input.text;
options.sipDomain = _sipDomain.input.text;
options.sipOutboundProxy = _sipOutboundProxy.input.text;
options.sipPort = [NSNumber numberWithInt: [_sipPort.input.text
integerValue]];
options.appKey = @"defaultApp";
NSError *error;
...
session = [FPWCSApi2 createSession:options error:&error];
...
[session connect];
```

## 3. Исходящий звонок

`FPWCSApi2Session.createCall`, `FPWCSApi2Call.call` [code](#)

При создании звонка в метод `createCall` передаются параметры:

- имя вызываемого SIP-аккаунта
- вид для отображения локального видеопотока
- вид для отображения видеопотока, получаемого от собеседника
- параметры аудио и видео

```
- (FPWCSApi2Call *)call {
    FPWCSApi2Session *session = [FPWCSApi2 getSessions][0];
    FPWCSApi2CallOptions *options = [[FPWCSApi2CallOptions alloc] init];
    options.callee = _callee.input.text;
    options.localDisplay = _videoView.local;
    options.remoteDisplay = _videoView.remote;
```

```

    options.localConstraints = [[FPWCSApi2MediaConstraints alloc]
initWithAudio:YES video:YES];
    options.remoteConstraints = [[FPWCSApi2MediaConstraints alloc]
initWithAudio:YES video:YES];
    NSError *error;
    call = [session createCall:options error:&error];
    ...
    [call call];
    return call;
}

```

#### 4. Получение от сервера события, сигнализирующего о входящем звонке

`FPWCSApi2Session.onIncomingCallCallback` code

```

[session onIncomingCallCallback:^(FPWCSApi2Call *rCall) {
    call = rCall;

    [call on:kFPWCSCallStatusBusy callback:^(FPWCSApi2Call *call){
        [self changeCallStatus:call];
        [self toCallState];
    }];

    [call on:kFPWCSCallStatusFailed callback:^(FPWCSApi2Call *call){
        [self changeCallStatus:call];
        [self toCallState];
    }];

    [call on:kFPWCSCallStatusRing callback:^(FPWCSApi2Call *call){
        [self changeCallStatus:call];
        [self toHangupState];
    }];

    [call on:kFPWCSCallStatusHold callback:^(FPWCSApi2Call *call){
        [self changeCallStatus:call];
        [self changeViewState:_holdButton enabled:YES];
    }];

    [call on:kFPWCSCallStatusEstablished callback:^(FPWCSApi2Call *call){
        [self changeCallStatus:call];
        [self toHangupState];
        [self changeViewState:_holdButton enabled:YES];
    }];

    [call on:kFPWCSCallStatusFinish callback:^(FPWCSApi2Call *call){
        [self changeCallStatus:call];
        [self toCallState];
        [self dismissViewControllerAnimated:YES completion:nil];
    }];
    ...
}];

```

## 5. Ответ на входящий звонок

`FPWCSApi2Call.answer` code

```
alert = [UIAlertController
    alertControllerWithTitle:[NSString stringWithFormat:@"Incoming call
from '%@'", [rCall getCallee]]
    message:error.localizedDescription
    preferredStyle:UIAlertControllerStyleAlert];

UIAlertAction* answerButton = [UIAlertAction
    initWithTitle:@"Answer"
    style:UIAlertActionStyleDefault
    handler:^(UIAlertAction * action) {
        [call getLocalConstraints].video =
[[FPWCSApi2VideoConstraints alloc] init];
        [call setLocalDisplay:_videoView.local];
        [call setRemoteDisplay:_videoView.remote];
        [call answer];
    }];

[alert addAction:answerButton];
UIAlertAction* hangupButton = [UIAlertAction
    initWithTitle:@"Hangup"
    style:UIAlertActionStyleDefault
    handler:^(UIAlertAction * action) {
        [call hangup];
    }];

[alert addAction:hangupButton];
[self presentViewController:alert animated:YES completion:nil];
```

## 6. Удержание звонка

`FPWCSApi2Call.hold`, `FPWCSApi2Call.unhold` code

```
- (void)holdButton:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"UNHOLD"]) {
        if (call) {
            [call unhold];
            [_holdButton setTitle:@"HOLD" forState:UIControlStateNormal];
        }
    } else {
        if (call) {
            [call hold];
            [_holdButton setTitle:@"UNHOLD" forState:UIControlStateNormal];
        }
    }
}
```

## 7. Завершение исходящего звонка

FPWCSApi2Call.hangup code

```
- (void)callButton:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"HANGUP"]) {
        if ([FPWCSApi2 getSessions].count) {
            [call hangup];
        } else {
            [self toCallState];
        }
        ...
    }
}
```

## 8. Завершение входящего звонка

FPWCSApi2Call.hangup code

```
UIAlertAction* hangupButton = [UIAlertAction
                               initWithTitle:@"Hangup"
                               style:UIAlertActionStyleDefault
                               handler:^(UIAlertAction * action) {
                                   [call hangup];
                               }];

[alert addAction:hangupButton];
```

## 9. Закрытие соединения

FPWCSApi2Session.disconnect code

```
- (void)connectButton:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"DISCONNECT"]) {
        if ([FPWCSApi2 getSessions].count) {
            FPWCSApi2Session *session = [FPWCSApi2 getSessions][0];
            NSLog(@"Disconnect session with server %@", [session
getServerUrl]);
            [session disconnect];
        } else {
            NSLog(@"Nothing to disconnect");
            [self onDisconnected];
        }
        ...
    }
}
```