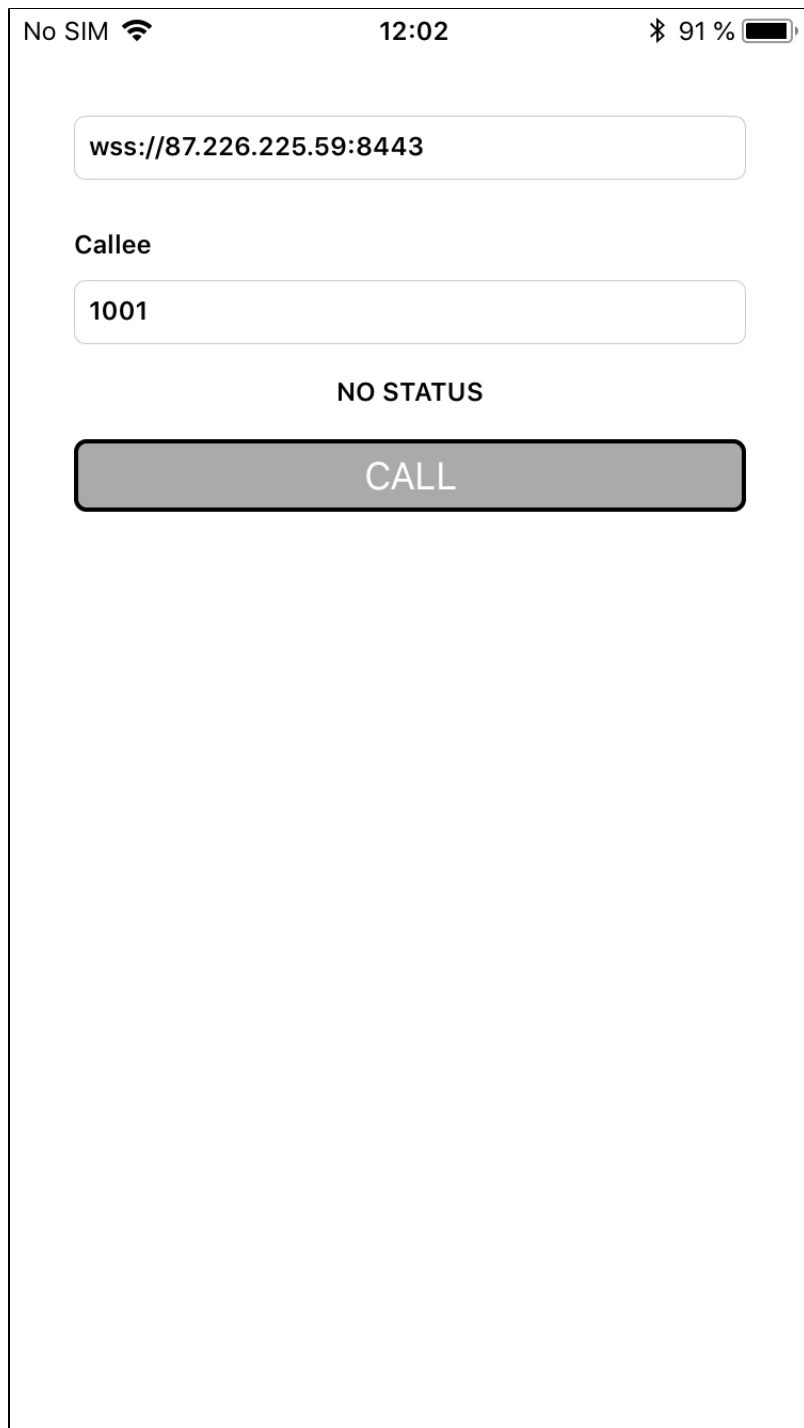


iOS Click to Call

Example of Click to Call application for iOS

This example allows to place outgoing audio call with one button click using account specified in server config file

/usr/local/FlashphonerWebCallServer/conf/apps/click-to-call/accounts.xml



The screenshot shows an iOS application interface for making a call. At the top, the status bar displays "No SIM" with a Wi-Fi icon, the time "12:02", and a battery level of "91 %". The main interface consists of three input fields and a call button. The first input field contains the text "wss://87.226.225.59:8443". The second input field is labeled "Callee" and contains the text "1001". Below these fields, the text "NO STATUS" is displayed. At the bottom, there is a large, grey, rectangular button with the text "CALL" in white capital letters.

Analyzing the code of the example

To analyze the code, let's take ClickToCallexample version, which can be downloaded with build [2.5.2](#).

View class for the main view of the application: ViewController (header file [ViewController.h](#); implementation file [ViewController.m](#)).

1. Import of API [code](#)

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

2. Connection to the server.

FPWCSApi2 createSession, FPWCSApi2Session connect [code](#)

FPWCSApi2SessionOptions object with the following parameters is passed to createSession() method

- URL of WCS server
- appKey of internal server-side application(clickToCallApp)

```
- (FPWCSApi2Session *)connect {
    FPWCSApi2SessionOptions *options = [[FPWCSApi2SessionOptions alloc] init];
    options.urlServer = _connectUrl.text;
    options.appKey = @"clickToCallApp";
    NSError *error;
    session = [FPWCSApi2 createSession:options error:&error];
    ...
    [session connect];
    return session;
}
```

3. Receiving the event confirming successful connection.

ViewController onConnected, ViewController call [code](#)

On this event, ViewController call method is called to make outgoing call.

```
- (void)onConnected:(FPWCSApi2Session *)session {
    [self onHangup];
    [self call];
}
```

4. Outgoing call.

FPWCSApi2Session createCall, FPWCSApi2Call call [code](#)

The next options are passed to the createCall() method:

- callee SIP username
- call constraints: audio only

```
- (FPWCSApi2Call *)call {
    FPWCSApi2CallOptions *options = [[FPWCSApi2CallOptions alloc] init];
    options.callee = _callee.input.text;
    options.localConstraints = [[FPWCSApi2MediaConstraints alloc] initWithAudio:YES video:NO];
    NSError *error;
    call = [session createCall:options error:&error];
    ...
    [call call];
    return call;
}
```

5. Call hangup.

FPWCSApi2Call hangup [code](#)

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