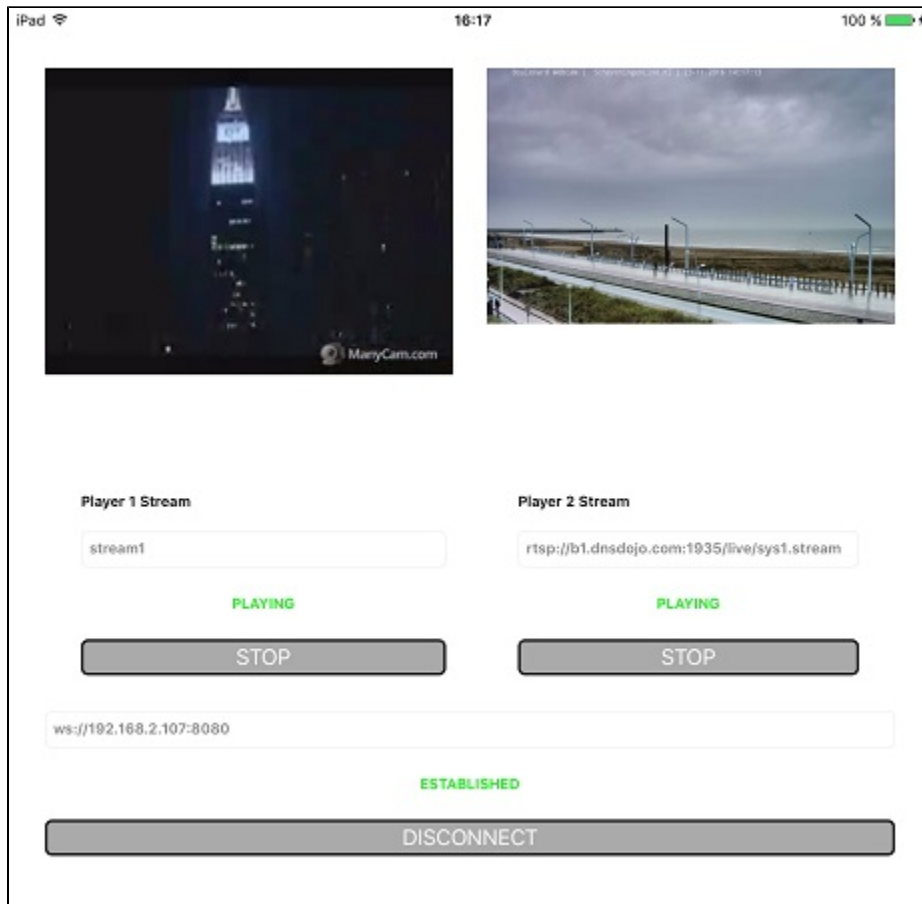


iOS 2 Players

Пример iOS-приложения с двумя плеерами

Данный пример показывает, как отобразить в одном приложении более одного плеера, каждый из которых может играть свой видеопоток.



Работа с кодом примера

Для разбора кода возьмем версию примера Player, которая доступна для скачивания в сборке [2.5.2](#).

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

1. Импорт API. [код](#)

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

2. Создание сессии

FPWCSApi2 createSession [код](#)

В параметрах сессии указываются:

- URL WCS-сервера
- имя серверного приложения defaultApp

```
FPWCSApi2SessionOptions *options = [[FPWCSApi2SessionOptions alloc] init];
options.urlServer = _connectUrl.text;
options.appKey = @"defaultApp";
NSError *error;
FPWCSApi2Session *session = [FPWCSApi2 createSession:options error:&error];
```

3. Подключение к серверу

FPWCSEApi2Session connect [код](#)

```
[session connect];
```

4. Получение от сервера события, подтверждающего успешное соединение.

ViewController onConnected [код](#)

```
- (void)onConnected:(FPWCSEApi2Session *)session {
    [_connectButton setTitle:@"DISCONNECT" forState:UIControlStateNormal];
    [self changeViewState:_connectButton enabled:YES];
    [self onStoped1];
    [self onStoped2];
}
```

5. Воспроизведение первого видеопотока.

FPWCSEApi2Session createStream, FPWCSEApi2Stream play [код](#)

Методу createStream передаются параметры:

- имя воспроизводимого потока
- вид для отображения потока

```
- (FPWCSEApi2Stream *)play1Stream {
    FPWCSEApi2Session *session = [FPWCSEApi2 getSessions][0];
    FPWCSEApi2StreamOptions *options = [[FPWCSEApi2StreamOptions alloc] init];
    options.name = _player1StreamName.text;
    options.display = _player1Display;
    NSError *error;
    player1Stream = [session createStream:options error:nil];
    ...
    if(![player1Stream play:&error]) {
        UIAlertController * alert = [UIAlertController
            alertControllerWithTitle:@"Failed to play"
            message:error.localizedDescription
            preferredStyle:UIAlertControllerStyleAlert];

        UIAlertAction* okButton = [UIAlertAction
            actionWithTitle:@"Ok"
            style:UIAlertActionStyleDefault
            handler:^(UIAlertAction * action) {

            }];

        [alert addAction:okButton];
        [self presentViewController:alert animated:YES completion:nil];
    }
    return player1Stream;
}
```

6. Воспроизведение второго видеопотока.

FPWCSEApi2Session createStream, FPWCSEApi2Stream play [код](#)

Методу createStream передаются параметры:

- имя воспроизводимого потока
- вид для отображения потока

```

- (FPWCSApi2Stream *)play2Stream {
    FPWCSApi2Session *session = [FPWCSApi2 getSessions][0];
    FPWCSApi2StreamOptions *options = [[FPWCSApi2StreamOptions alloc] init];
    options.name = _player2StreamName.text;
    options.display = _player2Display;
    NSError *error;
    player2Stream = [session createStream:options error:nil];
    ...
    if(![player2Stream play:&error]) {
        UIAlertController * alert = [UIAlertController
                                     alertControllerWithTitle:@"Failed to play"
                                     message:error.localizedDescription
                                     preferredStyle:UIAlertControllerStyleAlert];

        UIAlertAction* okButton = [UIAlertAction
                                    initWithTitle:@"Ok"
                                    style:UIAlertActionStyleDefault
                                    handler:^(UIAlertAction * action) {

                                        }]];

        [alert addAction:okButton];
        [self presentViewController:alert animated:YES completion:nil];
    }
    return player2Stream;
}

```

7. Остановка воспроизведения первого видеопотока.

FPWCSApi2Stream stop [код](#)

```

- (void)player1Button:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"STOP"]) {
        if ([FPWCSApi2 getSessions].count) {
            NSError *error;
            [player1Stream stop:&error];
        } else {
            NSLog(@"Stop playing, no session");
            [self onStopped1];
        }
        ...
    }
}

```

8. Остановка воспроизведения второго видеопотока.

FPWCSApi2Stream stop [код](#)

```

- (void)player2Button:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"STOP"]) {
        if ([FPWCSApi2 getSessions].count) {
            NSError *error;
            [player2Stream stop:&error];
        } else {
            NSLog(@"Stop playing, no session");
            [self onStopped2];
        }
        ...
    }
}

```

9. Закрытие соединения.

FPWCSSession disconnect [код](#)

```
- (void)connectButton:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"DISCONNECT"]) {
        if ([FPWCSSession getSession].count) {
            FPWCSSession *session = [FPWCSSession getSession][0];
            NSLog(@"Disconnect session with server %@", [session getServerUrl]);
            [session disconnect];
        } else {
            NSLog(@"Nothing to disconnect");
            [self onDisconnected];
        }
    } else {
        //todo check url is not empty
        [self changeViewState:_connectUrl enabled:NO];
        [self connect];
    }
}
```

10. Получение события, подтверждающего разъединение.

ViewController onDisconnected [код](#)

```
- (void)onDisconnected {
    [_connectButton setTitle:@"CONNECT" forState:UIControlStateNormal];
    [self changeViewState:_connectButton enabled:YES];
    [self changeViewState:_connectUrl enabled:YES];
    [self onStoped1];
    [self onStoped2];
}
```