

iOS Stream Recording

Пример iOS-приложения для записи видеопотока

Данный пример может использоваться с Web Call Server для публикации и записи WebRTC-видеопотока.

На скриншоте ниже (слева направо)

- публикуется видеопоток
- публикация потока завершена и соединение с сервером закрыто



В URL в поле ввода

- 192.168.2.107 - адрес WCS-сервера
- testStream - имя потока

Над полем ввода отображается видео с камеры.

По завершении публикации появляется ссылка для скачивания записи потока.

Работа с кодом примера

Для разбора кода возьмем версию примера StreamRecording, 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 <FPWCSEApi2/FPWCSEApi2.h>
```

2. Создание сессии и подключение к серверу.

FPWCSEApi2 createSession, FPWCSEApi2Session connect [код](#)

В параметрах сессии указываются:

- URL WCS-сервера
- имя серверного приложения defaultApp

```

- (FPWCSApi2Session *)connect {
    FPWCSApi2SessionOptions *options = [[FPWCSApi2SessionOptions alloc] init];
    url =[[NSURL alloc] initWithString:_connectUrl.text];
    options.urlServer = [NSString stringWithFormat:@"%s://%s:%s", url.scheme, url.host, url.port];
    streamName = [url.path.stringByDeletingPathExtension stringByReplacingOccurrencesOfString: @"/" withString:
@" "];
    options.appKey = @"defaultApp";
    NSError *error;
    FPWCSApi2Session *session = [FPWCSApi2 createSession:options error:&error];
    ...
    [session connect];
    return session;
}

```

3. Получение от сервера события, подтверждающего успешное соединение.

ViewController onConnected [код](#)

При получении данного события вызывается метод публикации потока ViewController publishStream

```

- (void)onConnected:(FPWCSApi2Session *)session {
    [self publishStream];
}

```

4. Публикация видеопотока.

FPWCSApi2Session createStream, FPWCSApi2Stream publish [код](#)

Методу createStream передаются параметры:

- имя публикуемого потока
- вид для локального отображения
- параметр record = true для записи потока при публикации
- размеры и FPS публикуемого видео при публикации с iPad

```

- (FPWCSEApi2Stream *)publishStream {
    FPWCSEApi2Session *session = [FPWCSEApi2 getSessions][0];
    FPWCSEApi2StreamOptions *options = [[FPWCSEApi2StreamOptions alloc] init];
    options.name = streamName;
    options.display = _remoteDisplay;
    options.record = true;
    if ( UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad ) {
        options.constraints = [[FPWCSEApi2MediaConstraints alloc] initWithAudio:YES videoWidth:640 videoHeight:
480 videoFps:15];
    }
    NSError *error;
    FPWCSEApi2Stream *stream = [session createStream:options error:&error];
    ...
    if(![stream publish:&error]) {
        UIAlertController * alert = [UIAlertController
            alertControllerWithTitle:@"Failed to publish"
            message:error.localizedDescription
            preferredStyle:UIAlertControllerStyleAlert];

        UIAlertAction* okButton = [UIAlertAction
            actionWithTitle:@"Ok"
            style:UIAlertActionStyleDefault
            handler:^(UIAlertAction * action) {
                [session disconnect];
            }];

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

```

5. Получение от сервера события, подтверждающего успешную публикацию.

ViewController onPublishing [код](#)

При получении данного события определяется имя файла записи потока с помощью вызова метода FPWCSEApi2Stream getRecordName

```

- (void)onPublishing:(FPWCSEApi2Stream *)stream {
    [_startButton setTitle:@"STOP" forState:UIControlStateNormal];
    [self changeViewState:_startButton enabled:YES];
    recordName = [stream getRecordName];
}

```

6. Закрытие соединения.

FPWCSEApi2Session disconnect [код](#)

```

- (void)startButton:(UIButton *)button {
    [self changeViewState:button enabled:NO];
    if ([button.titleLabel.text isEqualToString:@"STOP"]) {
        if ([FPWCSEApi2 getSessions].count) {
            FPWCSEApi2Session *session = [FPWCSEApi2 getSessions][0];
            NSLog(@"Disconnect session with server %@", [session getServerUrl]);
            [session disconnect];
        } else {
            NSLog(@"Nothing to disconnect");
            [self onDisconnected];
        }
    } else {
        [self changeViewState:_connectUrl enabled:NO];
        [self connect];
    }
}

```

7. Получение события, подтверждающего разъединение.

ViewController onDisconnected [код](#)

При получении данного события формируется ссылка для скачивания файла записи и вызывается метод ViewController playVideo для проигрывания записанного видео

```

- (void)onDisconnected {
    [self changeViewState:_connectUrl enabled:YES];
    [self onUnpublished];
    if (url && recordName) {
        NSString *urlString = @"http://www.sample-videos.com/video/mp4/720/big_buck_bunny_720p_1mb.mp4";
        NSString *urlString = [NSString stringWithFormat:@"http://%@:9091/client/records/%@", url.host,
recordName];
        _recordLink.text = urlString;
        [self playVideo: urlString];
    }
}

```

8. Воспроизведение записанного видео

AVPlayer play [код](#)

```

- (void)playVideo:(NSString *)urlString {
    NSURL *url = [NSURL URLWithString:urlString];
    AVURLAsset *movieAsset = [AVURLAsset URLAssetWithURL:url options:nil];
    [movieAsset.resourceLoader setDelegate:self queue:dispatch_get_main_queue()];

    AVPlayerItem *playerItem = [AVPlayerItem playerItemWithAsset:movieAsset];
    _player = [AVPlayer playerWithPlayerItem:playerItem];
    _playerViewController.player = _player;
    [_player play];
}

```