

JdPlaySdk iOS 集成文档

1. JdPlaySdk 概述

JdPlaySdk.framework 用于局域网内媒体发现和媒体控制。第三方可以采用两种方式接入：

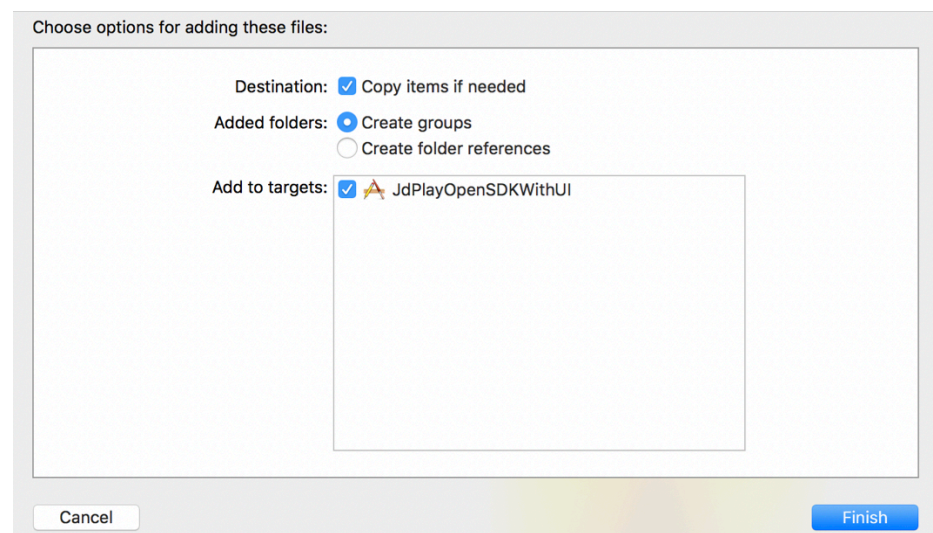
1.基于开放 demo 接入:接入简单,短时间内可接入完成(推荐方式)

2.基于开放 SDK 接口接入

2. 基于开放 SDK 的接入流程

1.将 JdPlaySdk.framework 拖拽到工程导航视图(project navigator)中的 Frameworks 虚拟目录下。

在弹出的对话框中按如下图所示勾选



2. 添加 SDK 依赖的系统库文件。分别是“**SystemConfiguration.framework**”、“**libstdc++.tbd**”、“**libz.1.2.5.tbd**”、“**AVFoundation.framework**”、“**Foundation.framework**”。另外还需添加的第三方库有 Reachability、MJRefresh、MJExtension、SDWebImage。

3. 禁用 Xcode7 下 iOS9App 传输安全, 修复无法连接服务器错误, 在 项目 info.plist 中添加一个键“App Transport Security Settings”, 类型为 Dictionary, 在该 dictionary 下添加一个键“Allow Arbitrary Loads”, 类型为 Boolean, 值为 YES, 然后保存文件。

3 注意事项

1. 本 SDK 只适用于真机调试
2. 在集成 SDK 的过程中中要确保 JdPlayManagerInit() 先于 updateInterfaceWithReachability: (监测网络状态发生改变调用的方法)调用, 这样才能成功开启底层服务。
3. 在 AppDelegate.m 文件的方法里加入以下代码, 确保程序进入后台后 socket 保持长连接, 以此达到流畅播控音乐的目的。

代码如下:

```
- (void)applicationDidEnterBackground:(UIApplication *)application  
{
```

```
JdPlay_appOnSuspend();
```

```
//程序进入后台保持 socket 长连接
```

```
UIApplication * app = [UIApplication sharedApplication]; __block
```

```
UIBackgroundTaskIdentifier bgTask;
```

```
bgTask = [app beginBackgroundTaskWithExpirationHandler:^(
```

```
dispatch_async(dispatch_get_main_queue(),^{
```

```
if(bgTask != UIBackgroundTaskInvalid) {
```

```
bgTask = UIBackgroundTaskInvalid;
```

```
}); };
```

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIOR
```

```
ITY_DEFAULT,0),^{ dispatch_async(dispatch_get_main_queue(),^{
```

```
if(bgTask != UIBackgroundTaskInvalid){ bgTask =
```

```
UIBackgroundTaskInvalid;
```

```
}); }); }
```

4.

```
JdPlayManagerInit();  
  
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector  
    (networkStateChange:) name:kReachabilityChangedNotification object:nil];  
self.conn = [Reachability reachabilityForInternetConnection];  
[self.conn startNotifier];  
[self updateInterfaceWithReachability:_conn];
```

上图中代码块不一定非要在

application:didFinishLaunchingWithOptions: 方法中调用。下图展

示了如何在设备列表界面初始化 SDK，接收设备列表回调。

```
@implementation JdMyDeviceViewController 显示设备列表的界面

- (void)viewDidLoad
{
    [super viewDidLoad];

    shareObj = [JdShareClass sharedInstance];
    mPresenter = [JdDeviceListPresenter sharedInstance];
    mPresenter.delegate = self; 设置代理 管理设备列表的单例类
    [self start];

    self.tableView.delegate = self;
    self.tableView.dataSource = self;
    self.tableView.tableFooterView = [[UIView alloc] init];

    [self.tableView registerNib:[UINib
        nibWithNibName:@"JdDeviceInfoTableViewCell" bundle:nil]
        forCellReuseIdentifier:@"cell"];
}

-(void)start {
    JdPlayManagerInit();

    [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector
        (networkStateChange:) name:kReachabilityChangedNotification object:nil];
    self.conn = [Reachability reachabilityForInternetConnection];
    [self.conn startNotifier];
    [self updateInterfaceWithReachability:_conn];

#pragma mark - 监听网路状态改变
    (void)networkStateChange:(NSNotification *)note{
        Reachability* curReach = [note object];
        NSParameterAssert([curReach isKindOfClass:[Reachability class]]);

        [self updateInterfaceWithReachability:curReach];
    }

    (void)updateInterfaceWithReachability:(Reachability *)reachability
```

你只需保证在start方法调用之前，初始化了JdDeviceListPresenter类即可

这一部分的代码必须是一起的，目的是开启SDK的服务，检测到网络状态发生改变时，SDK会有相关的设备信息回调。

设备列表回调方法：

```
#pragma mark - DeviceListView
-(void)onJdDeviceInfoChange:(NSArray *)infos
{
    self.deviceListArr = [NSMutableArray arrayWithArray:infos];
    [self performSelectorOnMainThread:@selector(reflashUI) withObject:nil
        waitUntilDone:YES];
}
```

5. SDK 在线文档地址:

<https://www.zybuluo.com/YangMu/note/784928>