

1 集成准备

提前下载项目Demo，准备好保利威直播的观看端链接，可以体验到更完整的功能。

2 项目配置

2.1 配置支持系统版本

打开项目的 PROJECT - Deployment Target - iOS Deployment Target 改为 9.0 或更高。

打开项目的 TARGETS - General - Deployment Info，把 iOS 系统改为 9.0 或更高。

2.2 配置 App Transport Security (ATS)

打开项目的 `info.plist` 文件，添加如下内容：

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
  <key>NSAllowsArbitraryLoadsInWebContent</key>
  <true/>
</dict>
```

2.3 配置设备旋转

播放器支持全屏播放，需要在 TARGETS - General - Deployment Info 中，勾选支持的横屏旋转方向。

2.4 配置后台播放和画中画

打开项目的 TARGETS - Signings & Capabilities，点击 + Capability，选择 Background Modes，勾选 'Audio, AirPlay, and Picture in Picture'。

3 集成SDK

3.1 Pod方式集成

3.1.1 集成PLVWebViewSDK

在 Podfile 文件中，添加以下内容：

```
pod 'PLVWebViewSDK', '~> 3.2.4'
```

在终端执行 `pod install`

3.1.2 集成系统小窗

如果需要使用系统小窗功能，则需要在 Podfile 文件中另外添加以下内容

```
# 包含系统画中画 -- 如果不使用系统小窗 则不需要下面配置  
pod 'PLVLiveScenesWebViewPIPModule', '1.10.6.3'
```

并在终端执行 `pod install`。

3.2 离线包方式集成

3.2.1 集成PLVWebViewSDK

将 PLVWebViewSDK.framework 库添加到自己的项目中。如果是使用源码的方式集成，则需要将 PLVWebViewSDK 文件整个添加到自己的项目即可。

3.2.2 集成系统小窗

1.如果项目使用系统小窗功能则需要将以下库添加到项目中

```
PLVAliHttpDNS、PLVBusinessSDK、PLVFoundationSDK、PLVIJKPlayer、PLVLiveScenesSDK
```

2.在General => Frameworks, Libraries, and Embedded Content中移除已添加的

PLVFoundationSDK.framework、PLVLiveScenesSDK.framework、PLVBusinessSDK.framework 库，并添加系统库 `libresolv.tbd`。

3.设置PLVIJKPlayer为Embed & Sign

4.Build Settings, Other Linker Flags中添加 `-ObjC`，如果项目已配置有，则不需要重复添加

5.运行报错

如果模拟器运行报错需要在Build Settings 中设置Excluded Architecture => Any iOS Simulator SDK 为 arm64

如果运行报错

```
Building for iOS Simulator, but the linked and embedded framework '***' was built for iOS + tvOS Simulator
```

可在Build Settings 中 设置 VALIDATE_WORKSPACE 为 YES即可。

4 代码调用

创建想要展示 WebView 的控制器页面 DemoViewController 继承于 PLVWebViewBaseViewController，并通过初始化方法 `-initWithConfig:` 创建控制器页面。示例代码如下：

```
// PLVWebViewDemoViewController.h

#import <PLVWebViewSDK/PLVWebViewSDK.h>

@interface PLVWebViewDemoViewController : PLVWebViewBaseViewController

@end
```

初始化时可通过重写 `initWithHandleForDemo` 进行自定义配置，同时可以对Bridge事件进行自定义处理。示例代码如下：

```

// PLVWebViewDemoViewController.m

@implementation PLVWebViewDemoViewController

#pragma mark - [ Override ]

- (void)viewDidLoad {
    [super viewDidLoad];
}

/// 初始化时，可以进行的自定义配置
//~ (void)initHandleForDemo {
//
//
//}

#pragma mark - [ Delegate ]
#pragma mark PLVFloatableWebViewBridgeDelegate
/// 如需要监听 js 回调事件，可以通过Delegate事件进行相应的处理，例如webViewBridgeShare消息
//~ (void)webViewBridgeShare:(PLVFloatableWebViewBridge *)bridge {
//
//
//}

@end

```

创建、配置、打开直播页面控制器

```

PLVWebViewConfig *config = [[PLVWebViewConfig alloc] init];
config.urlString = urlString;
config.allowFloatingWindow = YES;
config.isSystemFloatingWindow = NO;
config.userAgent = self.uaTextView.text;
config.enableAutoFloatWindow = YES;

PLVWebViewDemoViewController *webVC = [[PLVWebViewDemoViewController alloc] initWithConfig:config];
[self.navigationController pushViewController:webVC animated:YES];

```