

iOS 直播 UI SDK

变更记录: <https://git2.baijiashilian.com/open-ios/BaijiaYun/-/blob/master/BaijiaYun-CHANGELOG.md>

Git SDK 链接: <https://git2.baijiashilian.com/open-ios/BaijiaYun>

Git 源码链接: <https://git2.baijiashilian.com/open-ios/BJLiveUI>

源码对接文档: [https://git2.baijiashilian.com/open-ios/BJLiveUI/blob/master/wiki/直播 UI SDK 源码集成 & 自定义修改.md](https://git2.baijiashilian.com/open-ios/BJLiveUI/blob/master/wiki/直播UI SDK源码集成&自定义修改.md)

APP 下载:

<https://itunes.apple.com/cn/app/id1146697098?ls=1&mt=8>

旧版 SDK 文档: [iOS 直播 UI SDK 1.x 2.x](#)

功能简介

直播 UI SDK 在直播 Core SDK 的基础上提供了一个直播间互动模板, 主要包括一对一音视频互动, 多人音视频互动, 文档展示、文字聊天等功能, 包含一套完整的直播间 UI, 集成工作量小, 便于快速接入、开发。

如果需要较大程度自定义直播间的样式, 可以在通读本文档的

SDK 集成和使用部分后，参考下面的源码集成部分，源码集成 UI SDK，或者直接集成 [iOS 直播 Core SDK](#) 实现直播间界面。

1. 直播相关功能

直播的功能细节参考 [iOS 直播 Core SDK](#)。

2. UI

SDK 提供了三种不同班型的 UI 界面，三种 UI 都可以单独集成，不互相依赖。

- **大班课** 包括普通一对一直播、大小班分组直播、双摄像头直播。大班模板主要用于主讲人单向宣讲，偶尔连麦互动等场景。

进教室后在 SDK 获取到的班型为 `BJLRoomType_1vNClass` 或者 `BJLRoomType_1v1Class`，对应后台教室类型 2 或 6。此外，旧版本的班型也使用这个界面。一般情况下，除 `BJLRoomType_interactiveClass` 之外，即后台类型为 4 的以外的班型都使用这个界面。

- **竖屏大班课** 是 **大班课** UI 界面的竖屏样式，包含 **大班课** 的课件、音视频、聊天、用户等主要功能。主要用于习惯于竖屏操作的直播场景。**至少需要 3.8.0 以上的 SDK 版本。**

所有大班课都可以使用竖屏大班课的风格，主要兼容手机设备，在 `Pad` 设备上建议使用横屏的大班课。

- **小班课** 是专业多人直播互动 UI 界面，包括专业一对一直播，标准 UI 仅支持横屏显示。小班模板主要用于互动性较多，用户之间会经常连麦沟通等场景。

用于 `BJLRoomType_interactiveClass` 类型的教室，对应后台类型为 4 的教室使用这个界面。

这三种班型的代码和 UI 资源在 3.x 版本之后是完全分离的，可以按需集成，参考下面的集成方式部分。

此外，如果需要带货直播的竖屏 UI 界面，可以参考 [iOS 直播带货 UI SDK](#) 去集成。

表格列举了 SDK 提供的 UI 界面的主要的功能视图。

直播主界面	功能
采集视图	用于显示摄像头采集到的视频
播放视图	用于显示主播的视频
课件视图	用于显示白板、课件、画笔
聊天视图	用于发送聊天消息、显示直播间内的聊天消息列
用户列表	用于显示直播间内的所有用户
设置面板	用于设置音视频、课件等功能模块的相关参数
自定义按钮	将自定义按钮添加到直播间主界面，SDK 提供相应的点击回调方法
自定义弹框	点击自定义按钮之后显示在直播间主界面的弹窗可自定义样式

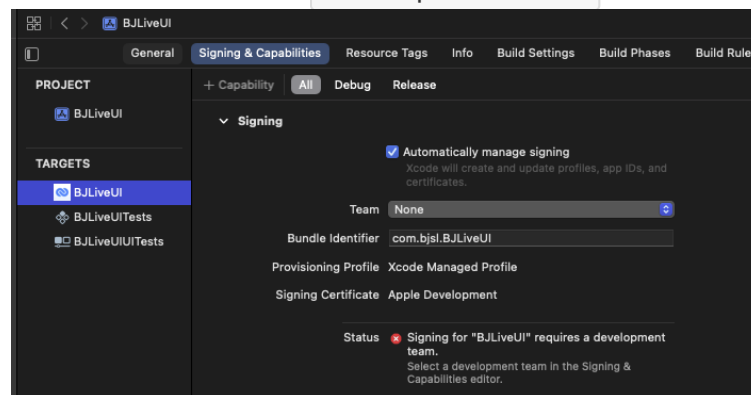
Demo

1. Demo 源文件

在 <https://git2.baijiashilian.com/open-ios/BaijiaYun/-/tags> 上下载最新的 SDK，SDK 提供了三种 demo，分别是 OC 版本的，Swift 版本的，Flutter 版本的 demo，demo 源文件分别在 `BJLiveUI-demo`，`BJLiveUI-demo-swift`，`bjliveui_flutter_demo` 文件夹中。

2. Demo 编译、运行

- 在 demo 的工程目录下执行 `pod install`，初次集成需要执行 `pod update` 更新 CocoaPods 的索引。
- 使用 Xcode 打开 demo 文件夹下的 `BJLiveUI.xcworkspace` 文件。
- 选择运行设备：**模拟器运行 demo 时无法采集音视频**；真机运行时，需要设置好 `development team`：



需设置 Team

- 使用 Xcode 运行 demo。

3. Demo 体验

- demo 运行成功后将进入如下登录界面，需要输入机构代码（参考 [专属域名说明](#)），六位参加码及用户名才能进入教室。
- 其中参加码通过使用 [百家云后台](#) 或者 [API](#) 创建一个教室获得，如果是八位参加码的系列课程参加码，需要取到其中一个可进入教室的六位参加码，用户名可自定义。
- 目前 UI 主要分为：大班课新版三分屏样式、新版竖屏大班课样式，和专业小班课的样式，。文档中以三分屏样式为例。

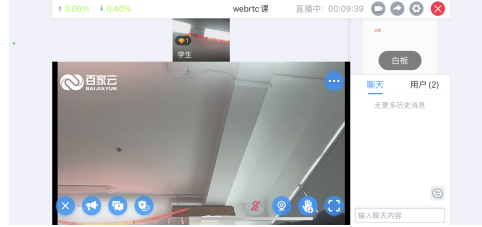


demo 登陆界面

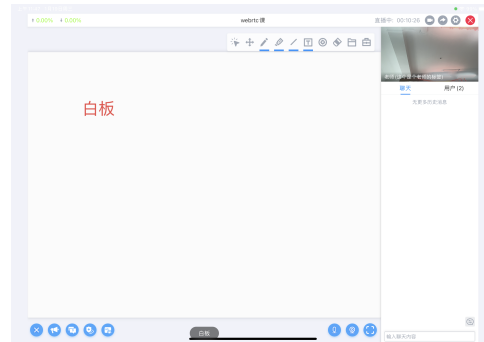
- 教室加载成功之后进入如下主界面，包含课件、画笔、采集音视频、播放音视频、聊天列表、用户列表等部分。下面是 SDK 的三分屏样式展示。



老师发言时学生界面



学生举手上麦时界面



iPad 老师界面

引入 SDK

1. 支持的设备和系统

- 4.x 版本 SDK 支持 **iOS 12.0** 及以上系统的 **iPhone**、**iPad** 等设备。3.x 版本 SDK 支持 **iOS 10.0** 及以上系统。集成 **4.0.0** 或以上版本的 **SDK** 尽量保持 **Xcode** 是最新稳定版。
- 小班课类型的教室，只支持 64-bit 设备进入教室，32-bit 设备进教室时通过 `enterRoomFailureWithError:` 返回错误码 `BJLErrorCode_enterRoom_unsupportedDevice`，参考 [iOS Device Summary](#)。

从 4.0.0 版本开始, SDK 最低支持版本升级到 iOS12.0 移除了对 32 位系统的支持, 且需要使用真机测试。

- **iPad**: 1、2、3、4、mini 1 是 32-bit, 其它都是 64-bit
- **iPhone**: 5、5C 之前的设备是 32-bit, 5S 开始是 64-bit。
- **iPod Touch**: 1、2、3、4、5 是 32-bit, 目前只有 6 是 64-bit。

2. 集成方式

SDK 依赖一些第三方库, 依赖关系复杂、手动配置繁琐, 建议使用 CocoaPods 方式引入。

注意: 文档中的版本号只是示意, 引入时的版本号建议参考 [Git](#) 上的最新版本。

- `Podfile` 设置 `source`

```
1. source 'https://github.com/CocoaPods/Specs.git'  
2. source 'https://git2.baijiashilian.com/open-  
ios/specs.git'
```

- `Podfile` 中引入 `BJLiveUI`, **SDK** 对于集成进行了调整, 如果不需要使用 **AVSDK**, 直接引入 **SDK** 即可, 如果需要使用 **AVSDK**, 仍然需要将引入 **SDK** 之后的语句加入。特别的, 点播和回放 **SDK** 的脚本语句不需要写入了, 可以直接移除。
- 最新版4.0已经移除对AVSDK的支持

```
1. pod 'BaijiaYun/BJLiveUI', '~> 4.0'
```

- 对于3.0及以下的SDK, 如果在后台创建的课程使用的都是 webrtc底层, 则不需要集成

BJHLMediaplayer.framework:

```
1. pod 'BaijiaYun/BJLiveUI', '~> 3.0'
```

- 对于3.0及以下的SDK，如果在后台创建的课使用的是 WebRTC 或者 AVSDK 底层，或者不确定使用的是什么底层，则需要集成 AVSDK（即 BJHLMediaplayer.framework）。

```
1. pod 'BaijiaYun/BJLiveUI', '~> 3.0'
2.
3. # 用于动态引入 Framework，避免冲突问题
4. script_phase \
5. :name => '[BJLiveCore] Embed Frameworks',
6. :script =>
   'Pods/BaijiaYun/frameworks/EmbedFrameworks.sh',
7. :execution_position => :after_compile
8.
9. # 用于清理动态引入的 Framework 用不到的架构，避免发布 AppStore 时发生错误，需要写在动态引入 Framework 的 script 之后
10. script_phase \
11. :name => '[BJLiveBase] Clear Archs From Frameworks',
12. :script =>
   'Pods/BaijiaYun/frameworks/ClearArchsFromFramework
   "BJHLMediaPlayer.framework"',
13. :execution_position => :after_compile
```

- 在工程目录下执行 `pod install`，初次集成需要执行 `pod update` 更新 CocoaPods 的索引。

大小班课的代码和资源文件是分离式的设计，可以按需集成。

- 如果仅需要大班课，可以这么集成：


```
1. pod 'BaijiaYun/BJLiveUIBigClass', '~> 4.0'
```

- 如果仅需要小班课，可以这么集成：

```
1. pod 'BaijiaYun/BJLiveUISmallClass', '~> 4.0'
```

- 如果仅需集成竖屏大班课，可以这么集成：

```
1. pod 'BaijiaYun/BJLiveUIEE', '~> 4.0'
```

版本升级

版本号格式为 `大版本.中版本.小版本[-alpha(测试版本)/beta(预览版本)]`：

- 测试版本和预览版本可能很不稳定，请勿随意尝试；
- 小版本升级只改 **BUG**、**UI** 样式优化，不会影响功能；
- 中版本升级、修改功能，更新 **UI** 风格、布局，会新增 **API**、标记 **API** 即将废弃，但不会导致现有 **API** 不可用；
- 大版本任何变化都是有可能的；

首次集成建议选择最新正式版本(版本号中不带有 `alpha`、`beta` 字样)，版本升级后请仔细阅读 **ChangeLog**，指定版本的方式有以下几种：

- 固执型：`pod update` 时不会做任何升级，但可能无法享受到最新的 **BUG** 修复，建议用于 **0.x** 版本；

```
1. pod 'BaijiaYun/BJLiveUI', '4.0.0'
```

- 稳妥型(推荐): `pod update` 时只会升级到更稳定的小版本, 而不会升级中版本和大版本, 不会影响功能和产品特性, 升级后需要 **适当测试**;

```
1. pod 'BaijiaYun/BJLiveUI', '~> 4.0.0'
```

- 积极型: `pod update` 时会升级中版本, 但不会升级大版本, 及时优化, 但不会导致编译出错不可用, 升级后需要 **全面测试**;

```
1. pod 'BaijiaYun/BJLiveUI', '~> 4.0'
```

- 激进型(不推荐): `pod update` 时会升级大版本, 可能导致编译出错、必须调整代码, 升级后需要 **严格测试**;

```
1. pod 'BaijiaYun/BJLiveUI'
```

工程设置

- 隐私权限: 在 `Info.plist` 中添加麦克风、摄像头、相册、蓝牙访问描述。

Privacy - Bluetooth Always Usage Description	String	用于辅助书写教学内容
Privacy - Bluetooth Peripheral Usage Description	String	用于辅助书写教学内容
Privacy - Camera Usage Description	String	用于视频上课、发言, 或者拍照上传课件、发送聊天图片
Privacy - Microphone Usage Description	String	用于语音上课、发言
Privacy - Photo Library Usage Description	String	用于上传课件、发送聊天图片

隐私权限

1. **Privacy - Microphone Usage Description** 用于语音上课、发言
2. **Privacy - Camera Usage Description** 用于视频上课、发言, 拍照传课件、聊天发图
3. **Privacy - Photo Library Usage Description** 用于上传课件、聊天发图

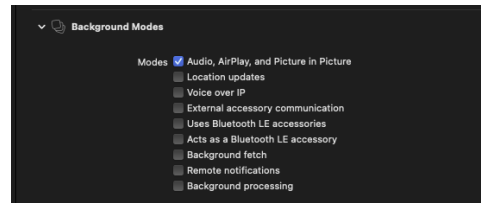
4. Privacy - Bluetooth Always Usage Description

用于辅助书写教学内容

5. Privacy - Bluetooth Peripheral Usage Description

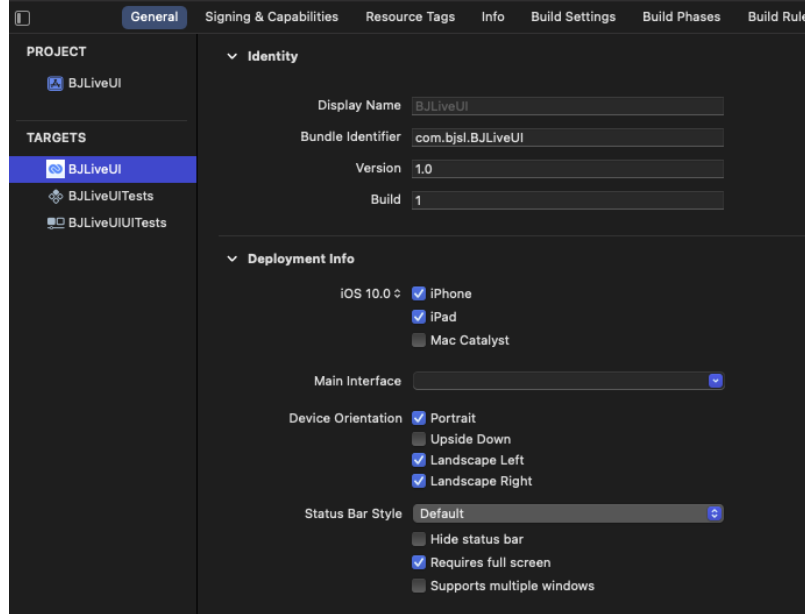
用于辅助书写教学内容

- 后台任务（打开这一选项之后，在 **App** 提交审核时，强烈建议录制一个视频，说明 **App** 确实用到了后台播放，否则审核很有可能不通过）：在 **Project > Target > Capabilities** 中打开 **Background Modes** 开关，选中 **Audio, Airplay, and Picture in Picture**。



后台音频

- Requires full screen（如果需要在 **iPad** 上运行，需要勾选此选项，保证在支持分屏的 **iPad OS** 上也能正常工作）：在 **Project > Target > General** 中勾选 **Requires full screen** 开关。



Requires full screen

Hello World

可参考 [demo](#)。

集成的整体流程如下：

- 在自己定义的相关文件中定义一个 `BJLRoomViewController` 的属性 `roomViewController`，用于管理教室视图。
- 使用教室相关信息（通过百家云后台或者 API 创建教室后获取）将 `roomViewController` 属性实例化。
- 为教室的进入、退出等事件添加监听和相应的回调处理，回调处理可以根据自身需求进行自定义，为教室视图的管理做好准备。
- 将 `roomViewController` 的 `view` 显示出来（可使用 `present`、`addChildViewController` 等方式）
- 如果 APP 不支持横屏，又需要使用 SDK 的横屏样式，需要在工程设置中支持横竖屏，在使用

`BJLRoomViewController` 的时候允许横屏，在APP的其他界面限制使用竖屏

1. 准备创建教室

- 引入头文件。

```
1. #import <BJLiveUIBase/BJLiveUIBase.h>
```

- 定义 `roomViewController`

```
1. @property (nonatomic) BJLRoomViewController  
   *roomViewController;
```

- 设置专属域名前缀，需要在创建 `BJLRoom` 实例之前设置。例如专属域名为 `demo123.at.baijiayun.com`，则前缀为 `demo123`，参考 [专属域名说明](#)。

```
1. NSString *domainPrefix = @"yourDomainPrefix";  
2. [BJLRoom setPrivateDomainPrefix:domainPrefix];
```

2. 创建、进入教室

可通过教室 ID 或参加码两种方式进行：

- 教室 ID 方式：教室 ID 通过使用 [百家云后台](#) 或者 [API](#) 创建一个教室获得；签名参数通过 [签名参数 sign 计算方法](#) 获得。

```
1. /**  
2. 通过 roomID 创建教室  
3. #param type 教室类型  
4. #param roomID roomID对象
```

```
5. #return 直播教室，如果 SDK 没有集成 type 对应的
   UISDK，返回 nil
6. */
7. + (__kindof instancetype)instanceWithRoomType:
   (BJLRoomVCType)type roomID:(BJLRoomID
   *)roomID;
```

- 参加码方式：参加码同样通过使用 [百家云后台](#) 或者 [API](#) 创建一个教室获得

```
1. /**
2. 通过参加码创建教室
3. #param type 教室类型
4. #param roomCode 参加码对象
5. #return 直播教室，如果 SDK 没有集成 type 对应的
   UISDK，返回 nil
6. */
7. + (__kindof instancetype)instanceWithRoomType:
   (BJLRoomVCType)type roomCode:(BJLRoomCode
   *)roomCode;
```

- 进入教室界面

```
1.
   [UIWindow.bjl_keyWindow.bjl_visibleViewController
   bjl_presentFullScreenViewController:self.roomViewC
2.
   animated:YES
3.
   completion:nil];
```

由于直播间横屏或者竖屏都有，需要在进教室之前的页面明确好原课表页面的方向，以免受直播间横竖屏的影响，导致退教室之

后可能出现的方向不一致导致崩溃:

1. 例如是从课表页面**present** 直播间，如果课表页面仅支持竖屏，则课表页面明确仅支持竖屏
2.

```
- (UIInterfaceOrientationMask)supportedInterfaceOrientations {
```
3.

```
    return UIInterfaceOrientationMaskPortraitUpsideDown;
```
4.

```
}
```

3. 监听进出教室事件

进出教室事件的监听可以即时获取教室的动态变化，便于添加自定义的回调处理。监听通过两种方式实现：

现：`BJLRoomVCDelegate`；监听 `BJLRoomViewController` 的方法调用。

3.1 BJLRoomVCDelegate

- 设置 `delegate`

1. `self.roomViewController.delegate = self;`

- `BJLRoomVCDelegate` 方法

1. */** 进入教室 - 成功 */*
2.

```
- (void)roomViewControllerEnterRoomSuccess: (BJLRoomViewController *)roomViewController {
```
3.

```
    NSLog(@"[%@ %@]", NSStringFromSelector(_cmd), roomViewController);
```
4.

```
}
```

```
1. /** 进入教室 - 失败 */
2. - (void)roomViewController:
   (BJLRoomViewController *)roomViewController
   enterRoomFailureWithError:(BJLError *)error {
3.     NSLog(@"[%@ %@, %@]",
           NSStringFromSelector(_cmd),
           roomViewController, error);
4. }
```

```
1. /**
2. 即将退出教室 - 正常/异常
3. 正常退出 `error` 为 `nil`，否则为异常退出
4. 参考 `BJLErrorCode` */
5. - (void)roomViewController:
   (BJLRoomViewController *)roomViewController
   willExitWithError:(nullable BJLError *)error {
6.     NSLog(@"[%@ %@, %@]",
           NSStringFromSelector(_cmd),
           roomViewController, error);
7. }
```

```
1. /**
2. 退出教室 - 正常/异常
3. 正常退出 `error` 为 `nil`，否则为异常退出
4. 参考 `BJLErrorCode` */
5. - (void)roomViewController:
   (BJLRoomViewController *)roomViewController
   didExitWithError:(nullable BJLError *)error {
6.     NSLog(@"[%@ %@, %@]",
           NSStringFromSelector(_cmd),
           roomViewController, error);
7. }
```


4. 分享教室

仅大班课支持分享教室，可以在后台配置是否允许分享。点击分享将会回调 `roomViewControllerToShare:` 方法，SDK 接收一个需要分享的视图控制器。

```
1. - (nullable UIViewController
   *)roomViewControllerToShare:
   (BJLRoomViewController *)roomViewController {
2.     if (roomViewController ==
       self.roomViewController) {
3.         return self.shareViewController;
4.     }
5.     return nil;
6. }
```

5. 退出教室

```
1. [self.roomViewController exitWithCompletion:nil];
```

6. 其他功能

SDK 除了提供标准的直播间教室界面外，还提供了以下的页面，可以根据实际需求使用。

- 外接设备作为辅助摄像头的界面，参考 `BJLAsCameraViewController`。
- 进入教室的二维码扫码界面，参考 `BJLQRCodeScanner`。
- 进教室前媒体权限自检界面，参考 `BJLCheckGuideViewController`。
- 选取相册中图片的界面，参考 `BJLPhotoListViewController`。

常见问题

pod 集成时报错。

命令行中一般会有错误的具体原因，可以根据错误原因搜索网上相关资料解决。下面给出部分错误的排查方法：

1、Failed to connect to GitHub to update the CocoaPods.

一般是由于本地的 pod 无法成功更新，连接不上 Github，需要使用确保能连接上 [Spec](#) 的网络。

2、could not find compatible version.

未执行过 `pod update`，导致本地不存在 SDK 的新版本，或者 Podfile 中项目支持的版本过低，或者存在不兼容的 SDK 依赖。

3、could not determine repo type for URL.

无法添加 repo，可以使用以下命令将 source repo 添加到 `~/cocoapods/repos/` 目录中。如果是 2.6.0 以下的 ruby 版本，`baijiashilian-open-ios` 改为 `baijiashilian-open-ios-specs`。

```
1. pod repo add baijiashilian-open-ios
   https://git2.baijiashilian.com/open-ios/specs
```

进入教室过程中出现错误提示：

错误提示	排查方法
未标记 nullable 的参数不能为空	检查创建用户、教室时的传参，确保参数非
不支持进入，	检查进入的教室类型和创建的教室类型是否

请使用其他端进入教室	
无法进入教室	检查在进入教室是否未正确设置域名前缀

教室用户之间音视频、消息未互通。

检查在进入教室是否未正确设置域名前缀。

是否可以修改 UI 样式。

修改 UI SDK 的样式需要源码集成。如果 UI 样式与 SDK 完全不一致，在开发能力允许的情况下集成 Core SDK 自行实现教室界面。

集成 3.8.0 以及之后的版本的问题。

1、Could not find or use auto-linked framework “.

在 Swift 5.1 支持了 ABI 稳定和 module stability 之后集成 Swift binary Framework 时，需要明确设置这个 target 的 Build Settings 的 Build Libraries for Distribution 为 YES ，如果依赖了其他的 Swift 三方库，也需要同样的设置。

使用 1.10.0 以及之后版本的 CocoaPods 支持了这个特性，详情见

<https://github.com/CocoaPods/CocoaPods/pull/9693>，但是第一次执行 pod install 集成时会出现未自动设置成功的情况，需要执行二次才能成功设置。

如果上述方法都无法正常通过的情况下，可以补充下面的脚本到 Podfile 中：

```
1. post_install do |installer|
2.   installer.pods_project.targets.each do |target|
3.     target.build_configurations.each do |config|
```

```
4.     config.build_settings['BUILD_LIBRARY_FOR_DISTRIBUTION']  
       = 'YES'  
5.   end  
6. end  
7. end
```

2、error: using bridging headers with module interfaces is unsupported.

SDK 在 3.8.0 开始使用 Swift 实现，支持直接作为 module 在需要使用的 Swift 文件中使用 `@import module` 的写法引入，因此需要移除在项目的 bridge header 文件中的 `#import <Framework/Header.h>` 的引入方式。

此外，如果是 Flutter 项目或者有修改过一些配置的主项目，需要在 iOS 的项目的主工程的 target 中的 **Build Settings** 搜索到 **Build Libraries for Distribution**，配置为 NO。

Swift 集成 OC 相关问题。

参考 Core SDK 的常见问题模块，此外，可以在 <https://git2.baijiashilian.com/open-ios/BaijiaYun/-/tags> 上下载最新的 SDK，在 `BJLiveUI-demo-swift` 文件夹中有 Swift 版本的 demo。

附录

1、源码集成

- [源码集成参考文档](#)

2、源码编译 Framework

- [源码打包 Framework 集成文档](#)



下载为pdf格式