

iOS 直播 UI SDK

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

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

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

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针对教育场景下师生互动模板, 主要包括师生一对一音视频互动, 多人音视频互动, 课件展示、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。

1. 直播相关功能

参考 [iOS 直播 Core SDK](#)



2. UI

SDK 提供了二种不同班型的 UI 界面。

- **大班课** 包括普通一对一教室、大小班分组教室、双摄像头教室，仅支持横屏显示。大班课主要用于主讲人单向宣讲，偶尔连麦互动等场景的教学。
- **小班课** 是专业UI 界面，包括专业一对一教室，仅支持横屏显示。专业小班课主要用于互动性较多，学生、老师会经常连麦沟通等场景的教学。

这两种班型的代码和UI资源是完全分离的，可以按需集成。

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

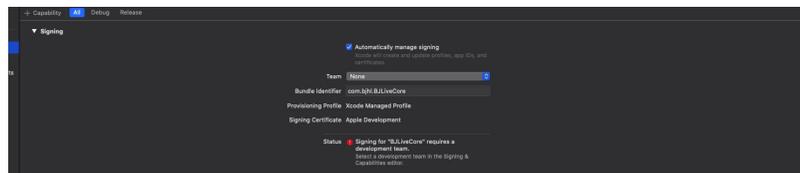
Demo

1. Demo 源文件

在 [git](#) 上下载最新的 SDK, demo 源文件在 `BJLiveUI/demo/BJLiveUI` 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` :



- 使用 Xcode 运行 demo

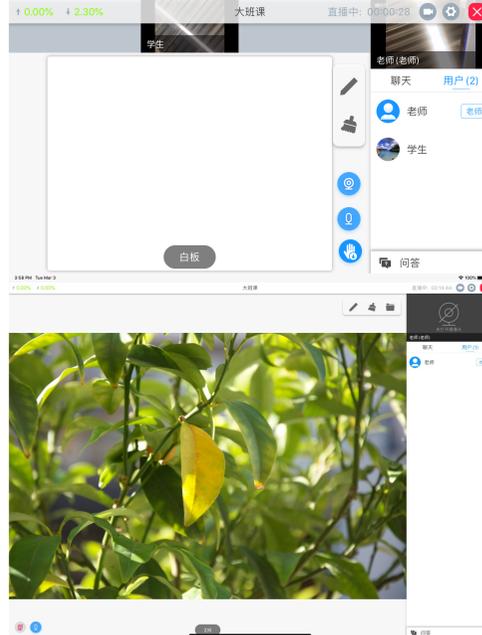
3. Demo 体验

- 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, 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 主要分成三种样式, 大班课新版三分屏样式以及旧版的支持竖屏显示的样式, 以及专业小班课的样式。文档中以三分屏样式为例。



- 教室加载成功之后进入如下主界面，包含课件、画笔、采集、播放、聊天列表等部分。下面分别是手机上只有老师发言，学生上麦发言以及平板上的三分屏样式。





引入 SDK

1. 支持的设备和系统

- 3.x版本 SDK 支持 **iOS 10.0** 及以上系统的**iPhone**、**iPad** 等设备，集成 **3.0.0** 或以上版本的 **SDK** 要求 **Xcode** 的版本至少为 **12.1**。教室类型为新版小班课的教室，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](#)。
- **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，建议使用 CocoaPods 方式引入：

- 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。
- 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'
```

- 如果在后台创建的课​​程使用的是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），写法如下：

```
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', '~> 3.0'
```

如果仅需要小班课，可以这么集成：

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

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', '3.0.0'
```

- 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', '~> 3.0.0'
```

- 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', '~> 3.0'
```

- 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、相册访问描述

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

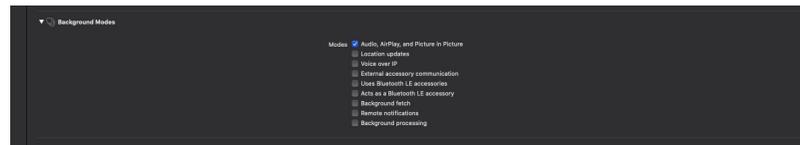
用于上传课件、聊天发图

4. Privacy - Bluetooth Always Usage Description

用于连接蓝牙手写板设备

Privacy - Camera Usage Description	String	视频
Privacy - Microphone Usage Desc...	String	音频
Privacy - Photo Library Usage Des...	String	课件

- 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`



- 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

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 `roomVlewController`，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 `roomVlewController` 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

- （可选）添加需要显示在教室主界面的自定义按钮，设置好点击按钮之后的响应回调和需要显示的自定义弹窗
- 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，如果sdk没有集成type对应的
   uidk，返回nil
3. #param type 教室类型
4. #param roomId roomId对象
5. #return 教室
6. */
7. + (__kindofinstancetype)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，如果sdk没有集成type对应的
   uidk，返回nil
3. #param type 教室类型
4. #param roomCode 参加码对象
5. #return 教室
6. */
7. + (__kindofinstancetype)instanceWithRoomType:
   (BJLRoomVCType)type roomCode:(BJLRoomCode
   *)roomCode;
```

- 进入教室界面

```
1. [self
   presentViewController:self.roomViewController
   animated:YES completion:nil];
```

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：`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。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];
```

附录

源码集成

- [源码集成参考文档](#)
- [源码打包framework集成文档](#)



下载为pdf格式