

鸿蒙 直播 UI SDK

- 包名: `@baijiayun/liveui` (大班课横版三分屏模板, har 包), 当前版本 `1.9.1`
- 公共组件包: `@baijiayun/liveuibase`, 当前版本 `1.9.1`
- 源码仓库 (开源): `https://git2.baijiashilian.com/open-android/harmonyos/baijiayun/ohos_liveui`
- **OHPM 仓**
库: `https://ohpm.openharmony.cn/#/cn/detail/@baijiayun%2Fliveui`
- Platform: HarmonyOS NEXT, runtimeOS: HarmonyOS, compatibleSdkVersion: `5.0.0(12)`
- IDE: **DevEco Studio 5.0** 及以上
- 底层依赖: `@baijiayun/livecore` (Core SDK), 详见《鸿蒙 直播 Core SDK》

1. 简介

UI SDK 基于 Core SDK 实现, 提供面向教育场景的**大班课横版三分屏**师生互动模板, 包括音视频互动、课件展示与画笔、文字聊天、在线人员、问答以及各类互动玩法, 集成工作量小, 适合需要快速上线的场景。

界面中的业务逻辑均由 Core SDK 提供, UI SDK 负责布局与交互。教室页面是一个 ArkUI 组件 `LiveRoomTripleUI`, 由宿主应用放入自己的 `@Entry` 页面中渲染; 老师、助教、学生使用同一个组件, 由 SDK 按角色展示不同功能。



说明：UI SDK 用于快速集成，界面本身不提供定制接口。
如需定制界面，可 fork 源码后修改（见 §4.3）。对外的稳定契约为：§3.2 的进房参数、§3.3 的组件参数、§5 的 `CallbackManager`。

鸿蒙端与 Android 直播 UI SDK 的主要差异：

- Android 通过 `LiveSDKWithUI.enterRoom()` 启动教室 Activity → 鸿蒙端由宿主在 `@Entry` 页面中放置 `LiveRoomTripleUI` 组件
- 系统返回键、页面导航、分享、后台运行由宿主处理（见 §3.4）

2. 快速集成

2.1. 包依赖

`oh-package.json5` 引入：

```
1. {  
2.   "dependencies": {  
3.     "@baijiayun/liveui": "^1.9.1"  
4.   }  
5. }
```

`@baijiayun/liveuibase`、`@baijiayun/livecore` 以及 Core SDK 的底层依赖均由 `@baijiayun/liveui` 传递依赖，无需额外声明。

2.2. 权限与后台运行

权限声明与 Core SDK 相同，见《鸿蒙 直播 Core SDK》§2.3。其中 `VIBRATE`（翻页到边界时的震动反馈）与 `KEEP_BACKGROUND_RUNNING`（退到后台继续运行）为 UI SDK 相关功能所需。

摄像头、麦克风权限的申请时机由宿主决定，**建议在进入教室前完成申请**。UI SDK 在进入教室前的准备页中会检测权限状态，并提供权限引导提示。

退到后台后继续播放声音，需要宿主申请长时任务并激活媒体会话，完整代码见《鸿蒙 直播 Core SDK》§16。

2.3. 初始化 SDK

UI SDK 无需初始化。建议在 `UIAbility.onCreate()` 中开启 Core SDK 的日志落盘与崩溃采集，便于问题排查：

```
1. import { LPFileLog, LPCrashCollector } from
   '@baijiayun/livecore';
2.
3. onCreate(): void {
4.   LPFileLog.init(this.context);
5.   LPCrashCollector.start();
6. }
```

专属域名在每次进房时通过进房参数 `domain` 传入，无需全局设置。

3. 调起教室页面

3.1. LiveRoomTripleUI

新建一个 `@Entry` 页面，放入 `LiveRoomTripleUI`：

```
1. import { router } from '@kit.ArkUI';
2. import { LiveRoomTripleUI, LiveRoomEntry } from
   '@baijiayun/liveui';
3.
4. interface RoomRouteParams {
```

```
5.   domain: string; code: string; userName: string;
    roomTitle: string;
6. }
7.
8. @Entry
9. @Component
10. struct LiveRoomPage {
11.   @State entry: LiveRoomEntry = { domain: "",
    code: "", userName: "", roomTitle: "" };
12.   @State ready: boolean = false;
13.   /** UI SDK 提供的返回键处理函数（普通成员，无需
    @State） */
14.   private backHandler: (() => boolean) | null =
    null;
15.
16.   /** 系统返回键 / 侧滑返回，交给 UI SDK 处理 */
17.   onBackPressed(): boolean {
18.     return this.backHandler !== null ?
    this.backHandler() : false;
19.   }
20.
21.   aboutToAppear(): void {
22.     const p = router.getParams() as
    RoomRouteParams;
23.     this.entry = { domain: p.domain, code: p.code,
    userName: p.userName, roomTitle: p.roomTitle };
24.     this.ready = true;
25.   }
26.
27.   build() {
28.     Column() {
29.       if (this.ready) {
30.         LiveRoomTripleUI({
31.           entry: this.entry,
32.           onExit: () => { router.back(); },
```

```
33.         onShare: (shareLink: string) => {
            this.shareRoom(shareLink); }, // 传 null 则不显示
            分享按钮
34.         onBackHandlerReady: (h: () => boolean)
            => { this.backHandler = h; }
35.     })
36. }
37. }
38. .width('100%').height('100%')
39. }
40. }
```

注意：@Entry 页面的 build() 根节点需为容器组件（如 Column ），不能直接使用自定义组件作为根节点。

3.2. 进房参数

entry 参数支持两种类型，按进房方式二选一：

- LiveRoomEntry：参加码进房
- LiveRoomSignEntry：签名进房，适用于已有自己用户体系、不希望用户输入参加码的场景

公共字段：

字段	类型	必填	说明
domain	string	是	机构专属域名，形如 https://demo123.at.
userName	string	是	用户昵称
roomTitle	string	是	房间标题，用于顶栏显示进房成功后会替换为服务，未知时可传空字符串

brtcApiUrl	string	否	音视频服务地址，一般会根据 domain 自动私有化部署时请联系技术
------------	--------	---	------------------------------------

LiveRoomEntry （参加码）：

字段	类型	必填	说明
code	string	是	参加码， 用户角色由参加码决定

LiveRoomSignEntry （签名）：

字段	类型	必填	说明
roomId	string	是	房间号
userNumber	string	是	用户编号， 请使用业务系统定义的用户 ID， 关系到计费统计
userType	LPUserType	是	用户角色
sign	string	是	签名
userAvatar	string	否	头像地址
groupId	number	否	分组 ID， 不分组时不传

```
1. import { LiveRoomTripleUI, LiveRoomSignEntry }
   from '@baijiayun/liveui';
2. import { LPUserType } from '@baijiayun/livecore';
3.
4. LiveRoomTripleUI({
5.   entry: {
6.     domain: 'https://demo123.at.baijiayun.com',
```

```
7.     roomId: '房间号',
8.     userNumber: '10086',
9.     userName: '张三',
10.    userType: LPUserType.Student,
11.    sign: '业务服务端计算的签名',
12.    roomTitle: ""
13.  } as LiveRoomSignEntry,
14.  onExit: () => { router.back(); }
15. })
```

注意：`sign` 必须由业务服务端计算，切勿在客户端计算或存放密钥。计算规则参考 [签名规则](#)。

3.3. 组件参数

参数	类型	说明
<code>entry</code>	<code>`LiveRoomEntry`</code>	LiveRoom
<code>onExit</code>	<code>() => void</code>	退出教室时 SDK 不会
<code>onShare</code>	<code>`((shareLink: string) => void) `</code>	null`
<code>onBackHandlerReady</code>	<code>(handler: () => boolean) => void</code>	SDK 提供 <code>onBackl</code>

注意：`entry` 在进入教室后不应再变化。参数变化会导致组件重建，教室需重新进入。

3.4. 宿主需要处理的事项

--	--	--

事项	做法	原
返回键	在 <code>onBackPressed()</code> 中调用 SDK 提供的处理函数	Ar 页
页面导航	在 <code>onExit</code> 中返回上一页	页
分享	在 <code>onShare</code> 中调起系统分享	不
后台运行	进入教室时申请长时任务并激活媒体会话，退出时释放	应 需

后台运行：在教室页面的 `aboutToAppear()` / `aboutToDisappear()` 中成对调用，`BackgroundTask` 与 `AvSessionKeeper` 的完整实现见《鸿蒙 直播 Core SDK》§16：

```
1. aboutToAppear(): void {
2.     // ...
3.     BackgroundTask.start(getContext(this) as
        common.UIAbilityContext);
4.     AvSessionKeeper.start(getContext(this) as
        common.UIAbilityContext, roomTitle);
5. }
6. aboutToDisappear(): void {
7.     BackgroundTask.stop(getContext(this) as
        common.UIAbilityContext);
8.     AvSessionKeeper.stop();
9. }
```

分享：请使用系统分享能力 `ShareKit`：

```
1. import { systemShare } from '@kit.ShareKit';
2. import { uniformTypeDescriptor as utd } from
    '@kit.ArkData';
3.
```

```
4. const record: systemShare.SharedRecord = {
5.   utd: utd.UniformDataType.PLAIN_TEXT, content:
    text, title: title
6. };
7. const controller = new
    systemShare.ShareController(new
    systemShare.SharedData(record));
8. await controller.show(ctx, {
9.   selectionMode:
    systemShare.SelectionMode.SINGLE,
10.  previewMode:
    systemShare.SharePreviewMode.DETAIL
11. });
```

说明：`ShareKit` 为 HarmonyOS 系统能力，建议对调用做异常处理。

3.5. 横屏与全屏

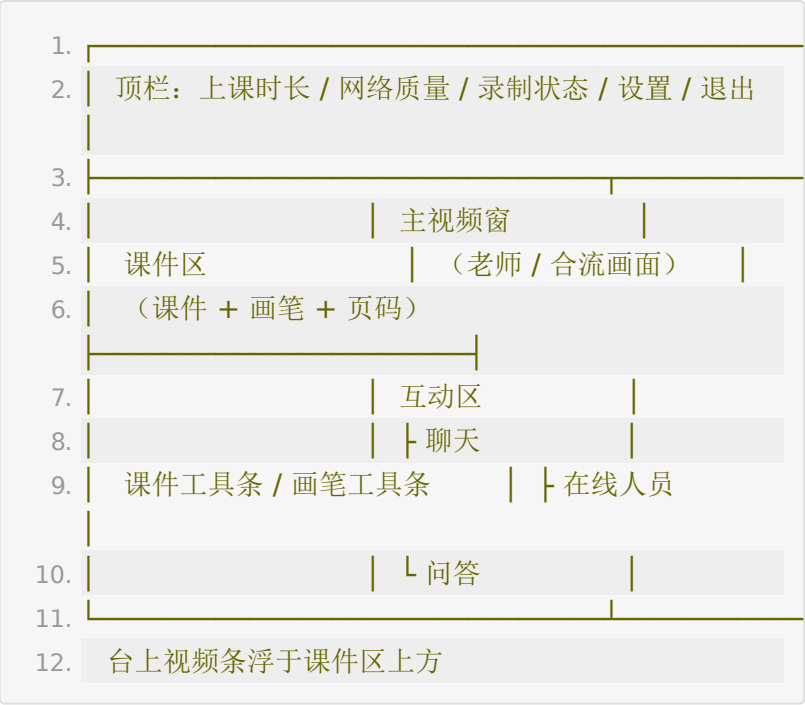
教室页面的横屏与全屏由 `LiveRoomTripleUI` 自行处理，宿主无需设置：

- 支持正、反两个横屏方向，跟随系统「自动旋转」开关；
- 屏幕旋转 180° 时，SDK 会自动校正本地预览与发布出去的画面；
- 已按系统安全区域处理内容边距。

注意：教室页面使用横屏。退出教室返回竖屏页面时，请在该页面的 `onPageShow()` 中恢复竖屏方向。

4. 功能介绍

4.1. 界面构成



主要功能:

分类	功能
音视频	举手上台、老师邀请上台、 开关摄像头与麦克风、清晰度切换、 屏幕共享、画中画、台上视频与主讲人切换
课件	静态 / 动画 PPT 、翻页、缩放、快速切页、 12 种画笔、激光笔、课件管理、作业区
聊天	文字、表情、图片、私聊、@、引用、 撤回、禁言、置顶、翻译、全屏聊天飘窗
教室工具	计时器、点名、答题器、抢答器、红包雨、 网页课件、问答、点赞奖励、公告
学生端互动	测验、抽奖、 PK 、专注度检测、 红包雨排行榜、活动推送
其他	商品展示与讲解、直播点赞、学情报告、

4.2. 公开导出

@baijiayun/liveui :

类别	符号
教室模板	LiveRoomTripleUI
进房参数	LiveRoomEntry LiveRoomSignEntry LiveRoomEntryBase EnterRoomParams
子组件	TopMenuBar LiveStatus formatElapsed MainVideoWindow InteractivePanel ChatPanel UserListPanel WhiteboardArea PageIndicator pageIndicatorText

@baijiayun/liveuibase 为各业务 UI 模板共用的组件库，包括主题与尺寸、视频窗口件、通用弹窗、教室工具窗口、聊天、课件管理、设置、商品等组件。

说明：@baijiayun/liveuibase 中只有 \$5 的 **CallbackManager** 及其回调接口属于稳定接口。其余导出的组件、常量与工具函数供各 UI 模板复用及 fork 后参考，参数可能随版本调整，不承诺兼容。

4.3. 自定义 UI

UI SDK 不提供运行时替换组件的机制，如需定制界面：

1. 修改配色与尺寸：fork @baijiayun/liveuibase，修改 LiveTheme / LiveDimens 中的集中配置；
2. 替换某个面板：fork ohos_liveui，在 LiveRoomTripleUI 中替换对应子组件。ChatPanel、UserListPanel、MainVideoWindo

w、WhiteboardArea、InteractivePanel 已单独导出，便于复用与替换；

3. 完全自定义界面：直接使用 Core SDK，课件与画笔组件 WhiteboardPanel 由 Core SDK 提供，详见《鸿蒙 直播 Core SDK》。

5. 回调注入 - CallbackManager

组件参数 (§3.3) 负责教室页面与宿主的对接；如需替换 SDK 的默认行为，例如账号在其他端登录时跳转到自己的页面、退出前做二次确认、点击商品跳转到自己的商品页等，请使用 CallbackManager。不注册时使用 SDK 的默认实现。

5.1. 获取实例

```
1. import { CallbackManager } from
   '@baijiayun/liveuibase';
2.
3. const cb = CallbackManager.getInstance();
```

5.2. 注册 / 注销 API

```
1. // 账号在其他端登录（不注册时弹出 SDK 默认提示）
2. cb.setEnterRoomConflictListener({
3.   onConflict: (endType: number, endTypeName:
   string, callback: RoomExitCallback) => {
4.     callback.exit(); // 或 callback.cancel()
5.   }
6. });
7.
8. // 用户主动退出教室时的二次确认（不注册时弹出 SDK
   默认确认框）
```

```
9. cb.setExitListener({
10.   onRoomExit: (callback: RoomExitCallback) => {
      callback.exit(); }
11. });
12.
13. // 进入 / 退出教室状态通知（四个方法均可选）
14. cb.setRoomStateListener({
15.   onEnterRoomSuccess: () => { },
16.   onEnterRoomError: (e: LPRoomError) => { },
17.   onExitRoomSuccess: () => { },
18.   onExitRoomError: (e: LPRoomError) => { }
19. });
20.
21. // 老师打开的自定义网页（注册后 SDK 不再使用内置窗口打开）
22. cb.setCustomPageListener({
23.   onOpenCustomPage: (url: string) => { },
24.   onCloseCustomPage: (url: string) => { }
25. });
26.
27. // 商品点击（注册后 SDK 不再使用内置浏览器打开，并自动收起商品窗口）
28. cb.setShopProductCallback({
29.   onOpenBrowserToBuy: (product:
      LiveProductModel) => { }
30. });
31.
32. // 退出教室时清空
33. cb.clearListener();
```

API	说明
setEnterRoomConflictListener	账号在其他端登录。 此时当前教室已失效， 仅表示不退出页面

setExitListener	用户主动退出教室。老师「下课并退出」仍使用的确认框，不经过此回调。
setRoomStateListener	状态通知，不影响 SDK 断线重连成功时也会回调。 onEnterRoomSuccess 用于统计时请自行去重。
setCustomPageListener	回调参数为网页地址，宿主需自行使用 Web 端。
setShopProductCallback	商品点击。buyUrl 为空的商品也会回调，便于宿主按商品 ID 跳转。
clearListener	清空全部回调，退出教室。

5.3. 接口签名

符号	类别	说明
CallbackManager	类	单例，通过 getI
RoomExitCallback	接口	exit() / cancel 必须调用其中之一。
RoomEnterConflictListener	接口	onConflict(end endTypeName,
RoomExitListener	接口	onRoomExit(ca
RoomStateListener	接口	onEnterRoomS onEnterRoomE onExitRoomSu onExitRoomErr
RoomCustomPageListener	接口	onOpenCuston onCloseCuston

ShopProductCallback	接口	onOpenBrowse onOpenProduc 均可选
---------------------	----	-------------------------------------

```
1. import {  
2.   CallbackManager, RoomExitCallback,  
   RoomEnterConflictListener, RoomExitListener,  
3.   RoomStateListener, RoomCustomPageListener,  
   ShopProductCallback  
4. } from '@baijiayun/liveuibase';
```

6. 已知限制

分类	说明
模板	当前提供大班课横版三分屏模板
进房方式	支持参加码与签名进房，暂不支持链接进房
设备检测	设备检测合并在进入教室前的准备页中，不提供独立页面
互动玩法	测验、抽奖、PK、专注度检测、活动推送由老师在 Web 端发起，客户端支持学生端参与；暂不支持转盘
界面定制	不提供运行时定制接口，需 fork 源码修改

7. 示例工程

开源仓库 `ohos_liveui` 的 `entry` 模块为 UI SDK 的源码调试工程。完整的接入示例请参考本文 §3.1 的教室页面代码，并按《鸿蒙 直播 Core SDK》§2.3、§2.4、§16 补充权限声明、日志初始化与后台运行。

