鸿蒙 点播回放 UI SDK

- 包名：`@baijia/videoplayerui`（har 包）
- **OHPM 仓库**：`https://ohpm.openharmony.cn/#/cn/detail/@baijia%2Fvideoplayerui`
- **Core 仓库**：`https://ohpm.openharmony.cn/#/cn/detail/@baijia%2Fvideoplayer`
- 依赖：`@baijia/videoplayer`（Core SDK，详见 Core 文档）
- Platform：HarmonyOS NEXT，runtimeOS: `HarmonyOS`，compileSdkVersion: `6.0.2(22)`
- IDE：**DevEco Studio 6.0.2 Release**

1. 简介

带 UI 的点播 / 回放 SDK 基于 Core SDK，提供标准的 ArkUI 实现，方便快速集成。鸿蒙端使用 `@Entry({ routeName })` 装饰器注册命名路由，由 `router.pushNamedRoute` 调起，等价于 Android 通过 `Intent` 调起 `Activity`。

> *鸿蒙端实现与 Android `videoplayer-ui` 接口逐行对齐，平台差异主要体现在：*
>
> - *Android `Activity` → 鸿蒙 `@Entry({ routeName }) struct Page`*
> - *Android `Intent.putExtra` → 鸿蒙 `router.pushNamedRoute({ name, params })`*

- *Android* `Fragment` *+ XML 布局 → 鸿蒙* `@Component struct` *（ArkTS 声明式 UI）*
- *Android* `BJYPlayerSDK.Builder` *全局静态字段 → 鸿蒙* `BJYPlayerSDK` *（静态字段）+* `BJYPlayerSDKBuilder`

2. 快速集成

2.1. 包依赖

`oh-package.json5` 引入：

```
{
  "dependencies": {
    "@baijia/videoplayerui": "^1.0.0"
}
```

2.2. 公开导出（`VideoPlayerUi/Index.ets`）

类别	导出符号
Page	`VideoPlayPage` / `VideoPlayTriplePage` /
入口工具类	`PBRoomUI`、`OnEnterPBRoomFailedListe
配置	`VideoPlayerConfig`、`SignModel`、`Visitc
全局 SDK	`BJYPlayerSDK`、`BJYPlayerSDKBuilder`
Core 再导出	`VideoPlayerFactory`、`IBJYVideoPlayer`、

2.3. 初始化 SDK

参考 Core SDK 的 `BJYPlayerConfig`，或使用 UI 层提供的 `BJYPlayerSDKBuilder` 链式 API（对齐 Android `BJYPlayerSDK.Builder`）：

```
import { BJYPlayerSDKBuilder } from '@baijia/videoplayerui';

new BJYPlayerSDKBuilder()
  .setDevelopMode(true)        // 开发者模式（正式发版请关闭）
  .setEncrypt(true)            // 加密
  .setCustomDomain('demo123')  // 专属域名前缀
  .enablePlaybackUserSignalSetting() // 解析回放 user 信令
  .build();                    // 构建并同步到 BJYPlayerConfig / PlayerDataLoader
```

`BJYPlayerSDKBuilder` API（`common/constant/BJYPlayerSDK.ets`）：

方法	说明
`setDevelopMode(isDevelopMode: boolean): BJYPlayerSDKBuilder`	设置开发者模式
`setCustomDomain(domain: string): BJYPlayerSDKBuilder`	设置专属域名
`setEncrypt(isEncrypt: boolean): BJYPlayerSDKBuilder`	设置加密
`enablePlaybackUserSignalSetting(): BJYPlayerSDKBuilder`	开启回放 user 信
`build(): void`	完成构建，同步配置到 `BJYPlayerConfi`

	与 `PlayerDataLoa`

> `build()` 中已跳过 Android 端的 X5、MMKV、Glide、RxJavaPlugins 等无对应能力项。

2.4. BJYPlayerSDK 全局静态配置

除 Builder 外，`BJYPlayerSDK` 提供大量公开静态字段供高级定制（`common/constant/BJYPlayerSDK.ets:13`）：

字段	类型
`CUSTOM_DOMAIN`	string
`customEnvironmentInfix`	string
`customEnvironmentSuffix`	string
`customAPIPrefix`	string
`IS_ENCRYPT`	boolean
`DEPLOY_TYPE`	number
`IS_DEVELOP_MODE`	boolean
`enablePlaybackUserSignal`	boolean
`STATIC_PPT_DEFAULT_SIZE`	number
`DISABLE_ANIM_PPT`	boolean
`MAX_SUPPORT_SHAPE_APPEND_COUNT`	number
`supportPiP`	boolean
`isVideoList`	boolean

enableSubtitleShadow	boolean
subtitleForegroundColor	number
playerVolume	number
enableScreenshot	boolean \
enableScreenRecording	boolean \
bjyUnknownError / bjyVideoUnknownError / bjyVideoPlayError / bjyVideoUrlError / bjyTokenInvalidError / bjyDefaultError	string
playerNumber	string
supportAD	boolean
customPlaybackStr / customPlaybackExtData	string
enableCustomPlaybackReport	boolean
hideChatLandscape	boolean
filterDefinitionList	VideoDefi
supportBackgroundAudio	boolean
supportLooping	boolean
supportBreakPointPlay	boolean

工具方法：

```
BJYPlayerSDK.getWebHost(): string;  // 拼接当前环境 web host，line 111
```

3. 快速集成

`PBRoomUI`（`VideoPlayerUi/src/main/ets/PBRoomUI.ets`）是 UI SDK 的唯一入口工具类，提供一系列静态方法。所有方法内部通过 `router.pushNamedRoute` 跳转到对应命名路由。

3.1. 调起点播播放页面

在线点播（路由 `VideoPlayPage`，行号 217）：

```
import { PBRoomUI, VideoPlayerConfig } from '@baijia/videoplayerui';

PBRoomUI.startPlayVideo(videoId, token, playerConfig /* 可传 null */);
```

```
/**
 * 进入标准在线点播播放界面
 * 对应 Android startPlayVideo(context, videoId, token, playerConfig)
 */
static startPlayVideo(videoId: number, token: string,
  playerConfig: VideoPlayerConfig | null): void
```

点播专辑（行号 236）：

```
PBRoomUI.startPlayVideoAlbum(albumId, playerConfig);
```

三分屏点播（路由 `VideoPlayTriplePage`，行号 253）：

```
PBRoomUI.startPlayVideoTriple(videoId, token, playerConfig);
PBRoomUI.startPlayVideoTripleAlbum(albumId, playerConfig);   // line 272
```

离线点播（行号 289）：

```
PBRoomUI.startPlayLocalVideo(downloadModel, playerConfig);
```

视频列表（行号 311，路由 `VideoListPage`）：

```
PBRoomUI.startPlayVideoList(videoInfoModelList: VideoInfoModel[],
                            playerConfig, position);
```

> ⚠ `VideoListPage` ***路由已注册但页面实现未交付***（鸿蒙端 `pages/` 目录下无 `VideoListPage.ets`，参考 `pages/` 仅有 `PBRoomPage.ets`、`VideoPlayPage.ets`、`VideoPlayTriplePage.ets` 三个）。`startPlayVideoList` 当前调用会因找不到命名路由而失败。

3.2. 调起回放播放页面

所有回放接口都跳到路由 `PBRoomPage`。

标准回放（无配置，sessionId 非长期课传 `'-1'`，行号 48）：

```
import { PBRoomUI, OnEnterPBRoomFailedListener } from '@baijia/videoplayerui';

class MyFailListener implements OnEnterPBRoomFailedListener {
 onEnterPBRoomFailed(msg: string): void { /* ... */ }
}

PBRoomUI.enterPBRoom(roomId, roomToken, sessionId, new MyFailListener());
```

标准回放 with config（行号 58）：

```
PBRoomUI.enterPBRoomWithConfig(roomId, roomToken, sessionId,
                               playerConfig, failListener);
```

裁剪回放（行号 69）：

```
PBRoomUI.enterPBRoomWithVersion(roomId, roomToken, sessionId,
                                version /* 0=原视频，-1=主版本 */,
                                playerConfig, failListener);
```

合并回放（行号 106 / 116）：

```
PBRoomUI.enterMixedPBRoom(mixedId, mixedToken, failListener);
PBRoomUI.enterMixedPBRoomWithConfig(mixedId, mixedToken, playerConfig, failListener);
```

回放合集（行号 148）：

```
PBRoomUI.enterAlbumPBRoom(albumNo, playerConfig, failListener);
```

离线回放（行号 171 / 179）：

```
PBRoomUI.enterLocalPBRoom(videoModel, signalModel);
PBRoomUI.enterLocalPBRoomWithConfig(videoMode
 signalModel, playerConfig);
```

> *`videoModel` / `signalModel` 来自 Core SDK 的 `DownloadManager.newPlaybackDownloadTask`，鸿蒙 UI 入口统一使用 `@baijia/videoplayer` 中的 `DownloadModel`（参见 `PBRoomUI.ets:13`）。*

`OnEnterPBRoomFailedListener` 接口（行号 19）：

```
export interface OnEnterPBRoomFailedListener {
  onEnterPBRoomFailed(msg: string): void;
}
```

`VideoInfoModel` 视频列表项（行号 27）：

```
export class VideoInfoModel {
  videoId: number = 0;
  token: string = '';
  title: string = '';
}
```

3.3. Page 路由注册

Page	路由名	文件
VideoPlayPage	'VideoPlayPage'	pa
VideoPlayTriplePage	'VideoPlayTriplePage'	pa
PBRoomPage	'PBRoomPage'	pa
VideoListPage	'VideoListPage'	未实

`PBRoomUI.buildPBRoomParams()`（行号 334）将 `VideoPlayerConfig` 扁平化为 router params，主要透传字段：`userName`、`userId`、`videoTitle`、`enableDragController`、`supportBackgroundAudio`、`userGroup`、`defaultAlbumIndex`、`isLandscape`、`isVideoMain`、`visitorSeconds/positiveText/negativeText/content`、`dragControllerContent/positiveText`、`horseLampContent/fontSize/fontColor/fontAlpha`。

3.4. VideoPlayerConfig

定义在 `VideoPlayerUi/src/main/ets/bean/VideoPlayerConfig.ets:98`。

```
import { VideoPlayerConfig, HorseLampConfig, SignModel,
        VisitorModel, DragControllerModel } from '@baijia/videoplayerui';

const cfg = new VideoPlayerConfig('userName', 'userId')
  .setHorseLamp(new HorseLampConfig())
  .setSignModelList([new SignModel()])
  .setVisitorModel(new VisitorModel())
  .setEnableDragController(true)
```

```
9.  .setDragControllerModel(new
    DragControllerModel())
10.  .setUserGroup(1);
```

全部字段：

字段	类型
supportBackgroundAudio	boolean
supportLooping	boolean
supportBreakPointPlay	boolean
userName	string
userId	string
userGroup	number
horseLamp	`HorseLampConfig \
notification	NotificationConfig
supportSeek	boolean
maxWatchTime	number
supportVideoRate	boolean
isVideoMain	boolean
isLandscape	boolean
enableToggleScreen	boolean
loginToken	string
avatar	string

`signModelList`	`SignModel[]`	
`defaultAlbumIndex`	number	
`visitorModel`	`VisitorModel \	
`enableDragController`	boolean	
`dragControllerModel`	`DragControllerModel \	
`videoTitle`	string	

构造函数（行号 147）：

```
1. constructor(userName: string = '', userId: string = '');
```

Builder 链式 setter（均返回 `VideoPlayerConfig`）：

方法	行号
`setHorseLamp(horseLamp: HorseLampConfig)`	152
`setSignModelList(signModelList: SignModel[])`	157
`setVisitorModel(visitorModel: VisitorModel)`	162
`setEnableDragController(enable: boolean)`	167
`setDragControllerModel(dragControllerModel: DragControllerModel)`	172
`setUserGroup(userGroup: number)`	177

> *Android 端 `VideoPlayerConfig.albumItemDataList` 在鸿蒙端**未在 `VideoPlayerConfig` 类内提供**，回放*

> *合集请用 `PBRoomUI.enterAlbumPBRoom(albumNo, ...)` 直接传 `albumNo` 字符串。*

3.5. 回放合集

使用专辑号入口（详见 §3.2 `enterAlbumPBRoom`），由 Core SDK 内部从服务端拉取专辑详情。

3.6. 点播签到

通过 `VideoPlayerConfig.setSignModelList()` 配置签到打点；通过 `CallbackManager.setSignListener()` 监听签到事件。

`SignModel` 字段（`bean/VideoPlayerConfig.ets:19`）：

字段	类型	默认值	说明	行号
`seconds`	number	`0`	触发时间点（秒）	21
`logo`	string	`''`	弹窗 logo URL	23
`title`	string	`''`	标题	25
`content`	string	`''`	正文	27
`btnText`	string	`''`	按钮文案	29

监听签到按钮点击：

```
import { CallbackManager, SignConsumer } from '@baijia/videoplayerui';
import { SignModel } from '@baijia/videoplayerui';
```

```
3.
4. class MySignConsumer implements
   SignConsumer {
5.   accept(model: SignModel): void { /* ... */ }
6. }
7. CallbackManager.getInstance().setSignListener(new
    MySignConsumer());
```

3.7. 点播分享

注册分享回调后，UI 上会显示分享按钮（默认不显示）。

```
1. import { CallbackManager, ShareListener } from
   '@baijia/videoplayerui';
2.
3. class MyShareListener implements ShareListener
   {
4.   onShareClicked(videoId: string): void { /* ... */ }
5. }
6. CallbackManager.getInstance().setShareListener(ne
    MyShareListener());
```

接口签名（ `common/CallbackManager.ets:35` ）：

```
1. export interface ShareListener {
2.   onShareClicked(videoId: string): void;
3. }
```

3.8. 点播/回放试看

`VideoPlayerConfig.visitorModel` 设置试看， `VisitorModel` （ `bean/VideoPlayerConfig.ets:36` ）字段：

字段	类型	默认值	说明
seconds	number	2147483647	试看时长（秒）
positiveText	string	''	确认按钮
negativeText	string	''	取消按钮
navigateText	string	''	跳转按钮
content	string	''	弹窗正文
navigateLink	string	''	跳转链接

监听试看弹窗按钮：

```
import { CallbackManager, VisitorListener } from '@baijia/videoplayerui';
import { IBJYVideoPlayer } from '@baijia/videoplayer';

class MyVisitorListener implements VisitorListener {
  onPositiveClick(dialogId: string, player: IBJYVideoPlayer): void { }
  onNavigateClick(dialogId: string, player: IBJYVideoPlayer): void { }
}
CallbackManager.getInstance().setVisitorListener(new MyVisitorListener());
```

> *Android 端 `onPositiveClick(DialogFragment, IBJYVideoPlayer)` 在鸿蒙端用 `dialogId: string`（`CustomDialog` 控制器 ID）替代 `DialogFragment`。*

3.9. 点播/回放禁止拖拽

```
const cfg = new VideoPlayerConfig()
  .setEnableDragController(false)   // 禁止拖拽
  .setDragControllerModel(new
DragControllerModel());
```

`DragControllerModel`（`bean/VideoPlayerConfig.ets:55`）：

字段	类型	默认值	说明	行号
`positiveText`	string	`''`	按钮文案	57
`content`	string	`''`	弹窗正文	59

监听用户拖拽尝试：

```
import { CallbackManager, DragConsumer } from
'@baijia/videoplayerui';
import { DragControllerModel } from
'@baijia/videoplayerui';

class MyDragConsumer implements
DragConsumer {
  accept(model: DragControllerModel): void { /*
自行弹窗 */ }
}
CallbackManager.getInstance().setDragListener(new
 MyDragConsumer());
```

3.10. HorseLampConfig / NotificationConfig

`HorseLampConfig`（`bean/VideoPlayerConfig.ets:85`）跑马灯：

字段	类型	默认值	行号
`content`	string	`''`	86
`fontSize`	number	`14`	87
`fontColor`	string	`'#FFFFFF'`	88
`fontAlpha`	number	`1.0`	89
`speed`	number	`50`	90
`displayStyle`	number	`0`	91

`NotificationConfig`（`bean/VideoPlayerConfig.ets:66`）后台播放通知（鸿蒙系统对后台媒体通知能力有限制）：

字段	类型	默认值	行号
`smallIcon`	string	`''`	68
`largeIcon`	string	`''`	70
`contentString`	string	`''`	72

构造函数（行号 74）：`new NotificationConfig(smallIcon, largeIcon, contentString)`。

4. 功能介绍

4.1. 包结构

```
VideoPlayerUi/src/main/ets/
├── Index.ets                    # 公开导出
├── PBRoomUI.ets                 # UI 入口工具类
```

```
├── bean/
│   └── VideoPlayerConfig.ets          # 配置模型（VideoPlayerConfig / SignModel / VisitorModel / DragControllerModel / HorseLampConfig / NotificationConfig）
├── common/
│   ├── CallbackManager.ets            # 全局回调注册
│   ├── LPError.ets
│   └── constant/
│       └── BJYPlayerSDK.ets           # 全局静态配置 + Builder
├── pages/                             # 路由页面（@Entry 装饰）
│   ├── VideoPlayPage.ets
│   ├── VideoPlayTriplePage.ets
│   └── PBRoomPage.ets
├── component/                         # 通用 UI 组件
│   ├── LoadingComponent.ets
│   ├── GestureComponent.ets
│   ├── ControllerComponent.ets
│   ├── ErrorComponent.ets
│   ├── MenuComponent.ets
│   ├── LampComponent.ets              # 跑马灯
│   ├── CommentComponent.ets
│   ├── MediaPlayerDebugComponent.ets
│   ├── playback/                      # 回放专用：PBChatComponent / AnnouncementComponent / PBToolboxComponents / PPTTripleComponent 等
│   ├── toolbox/                       # 答题、问答、测验
│   ├── subtitle/                      # 字幕组件
│   ├── roomoutline/                   # 大纲
│   ├── studyreport/
│   └── keyframe/
└── widget/                            # 弹窗、合集等小部件（如 VideoAlbumDialog）
```

4.2. 自定义点播 UI

鸿蒙端使用 ArkTS 声明式 UI，在 `VideoPlayPage` / `VideoPlayTriplePage` / `PBRoomPage` 中通过 `@Component struct` 组合各个组件。**未提供** Android 端 `ComponentManager` / `ComponentContainer` / `BaseVideoView` / `BJYVideoView` 等运行时容器化注册体系；如需定制 UI，请 fork 对应 Page 自行修改。

4.3. UI 组件清单

通用组件（ `component/` ）：

组件	文件
LoadingComponent	LoadingComponen
GestureComponent	GestureComponen
ControllerComponent	ControllerCompone
ErrorComponent	ErrorComponent.et
MenuComponent	MenuComponent.e
LampComponent	LampComponent.e
CommentComponent	CommentCompone
MediaPlayerDebugComponent	MediaPlayerDebug

回放专用（`component/playback/`、`component/toolbox/`、`component/subtitle/`、`component/roomoutline/`）：聊天、公告、答题、问答、测验、字幕、大纲、三分屏 PPT、学习报告、关键帧等。

弹窗与小部件（`widget/`）：合集对话框 `VideoAlbumDialog` 等。

5. 回调注入 - CallbackManager

`CallbackManager`（`common/CallbackManager.ets:91`）单例，对齐 Android `com.baijiayun.videoplayer.ui.listener.CallbackManager`。

5.1. 获取实例

```
1. const cb = CallbackManager.getInstance();  // line 106
```

5.2. 注册 / 注销 API

回调接口	注册方法
SignConsumer	setSignListener(consu
ShareListener	setShareListener(liste
VisitorListener	setVisitorListener(liste
DragConsumer	setDragListener(consu
OuterPlayerListener	setOuterPlayerListene

`ExternalAlbumListener`	`setExternalAlbumListe`
`OnVideoListRefreshListener`	`setOnVideoListRefresh`
`OnVideoListCallback`	`setOnVideoListCallbac`
全部清空	`clearListener()`

每个 set 方法都有对应的 getter（命名 `getXxx()`）供 SDK 内部调用。

5.3. 接口签名

```
// 签到（line 28）
export interface SignConsumer { accept(model: SignModel): void; }

// 分享（line 35）
export interface ShareListener {
onShareClicked(videoId: string): void; }

// 试看（line 44）
export interface VisitorListener {
  onPositiveClick(dialogId: string, player: IBJYVideoPlayer): void;
  onNavigateClick(dialogId: string, player: IBJYVideoPlayer): void;
}

// 拖拽（line 52）
export interface DragConsumer { accept(model: DragControllerModel): void; }

// 播放器事件外发（line 59）
export interface OuterPlayerListener {
```

```
  onPlayerStatusChanged(status: PlayerStatus): void;
  onPlayTimeChanged(current: number, duration: number): void;
  onPlayerError(error: LPError): void;
}

// 外部合集按钮（line 68）
export interface ExternalAlbumListener {
  onExternalAlbumClick(): void;
}

// 视频列表刷新（line 75）
export interface OnVideoListRefreshListener {
  onRefresh(): void;
  onLoadMore(): void;
}

// 视频列表数据（line 83）
export interface OnVideoListCallback {
  onDataChange(videoInfoModelList: VideoInfoModel[]): void;
  onLoadNoMoreData(): void;
}
```

6. 平台适配差异（鸿蒙 vs Android）

概念	Android
页面承载	`Activity` + `Intent`
Fragment / Layout XML	`Fragment` + XML

上下文参数	入口方法首参 `Context context`
`VideoPlayerConfig` 传递	`Intent.putExtra(ConstantUtil.VI config)`
回调实现	匿名内部类 / lambda
试看弹窗参数	`DialogFragment`
组件容器化	`ComponentManager` / `Compo` `BaseVideoView` / `BJYVideoVie`
视频列表	`VideoListActivity`

7. 已知限制

1. **`VideoListPage` 未实现**：`PBRoomUI.startPlayVideoList` 已写入路由 `'VideoListPage'`，但 `pages/` 目录无对应文件。
2. **ArkTS 严格模式**：所有 listener 必须用具名 `class implements`，不允许匿名对象字面量。
3. **`ComponentManager` 体系未提供**：自定义 UI 组件需直接 fork Page 并改写 `@Component`，无法在运行时插拔。
4. **后台播放通知**：`NotificationConfig` 字段保留但鸿蒙系统对后台音频通知限制较严，需结合 `ContinuousTask`（持续任务）申请。
5. **签名 / 证书**：`videoplayeruiDebug.p7b` / `videoplayeruiRelease.p7b` / `.cer` / `.p12` 已包含在仓库根，正式发布需自行申请并替换。

8. 完整对外导出清单

符号	类别	来源文件
`VideoPlayPage`	Page	pages/V
`VideoPlayTriplePage`	Page	pages/V
`PBRoomPage`	Page	pages/I
`PBRoomUI`	入口	PBRoor
`OnEnterPBRoomFailedListener`	接口	PBRoor
`VideoInfoModel`	模型	PBRoor
`VideoPlayerConfig`	配置	bean/V
`SignModel` / `VisitorModel` / `DragControllerModel` / `HorseLampConfig`	配置子项	bean/V
`BJYPlayerSDK` / `BJYPlayerSDKBuilder`	全局配置	commo
Core 再导出（`VideoPlayerFactory` / `IBJYVideoPlayer` / `PlayerStatus` / `PBRoom` / `PBRoomImpl` / `DownloadManager` / `DownloadModel` / `DownloadTask` 等）	透传	@baijia

> *`CallbackManager` 及其接口（`SignConsumer` / `ShareListener` / `VisitorListener` / `DragConsumer` / `OuterPlayerListener` 等）**未在 `Index.ets` 顶层 re-export***，需要通过相对路径或从 `@baijia/videoplayerui` 内部模块路径引入（视项目打

> *包时是否扁平化导出而定，必要时可在 `Index.ets` 追加一行 `export *` ）。*

下载为pdf格式