

Android 点播回放 UI SDK

- [源码地址](#)
- Platform: API 21+
- CPU: ARM-v7a, ARM64-v8a
- IDE: **Android Studio** Recommend
- [Change Log](#)
- [点播 3.X UI SDK 文档](#)、[回放 3.X UI SDK](#)

1. 简介

带 UI 的点播 SDK 基于点播 Core SDK，提供标准的 UI 实现，方便用户快速集成投入使用。此 SDK 代码开源，开发者可以自己建立分支进行开发，也欢迎给我们提 issue。

2. 快速集成

2.1. 添加 maven 仓库

从 4.0 SDK 起，引入了新的 `nexus.baijiayun.com` 仓库。

gradle 7.1 及以上

```
1. maven {  
2.     allowInsecureProtocol true
```

```
3.     url
      'http://nexus.baijiayun.com/nexus/content/groups/a
public/'
4.   }
5. maven { url 'https://git2.baijiashilian.com/open-
android/maven/raw/master/' }
```

gradle 7.1 以下

```
1. maven {url
  'http://nexus.baijiayun.com/nexus/content/groups/a
public/'}
2. maven { url 'https://git2.baijiashilian.com/open-
android/maven/raw/master/' }
```

2.2. 版本号说明

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

2.3. 添加依赖

最新版本请点击自取，[Releases](#) · [Open Android / VideoplayerUI2.0Demo](#) · [GitLab](#)

3. 快速集成

3.1. 初始化 SDK

参考 [点播 Core SDK 初始化](#)

3.2. 调起点播播放页面

`VideoPlayActivity` 为标准的点播播放界面，该布局为全屏居中播放，用户可以参考实现任意组合下的点播播放。

`PBRoomUI` 为 UI SDK 入口工具类，提供一系列 `static` 函数方便用户调用。

在线播放

```
1. /**
2.  * 进入标准在线点播播放界面
3.  * @param context    context
4.  * @param videoid    点播vid
5.  * @param token      点播token
6.  * @param playerConfig 创建播放器的配置项，
   null则取默认配置项
7.  */
8. public static void startPlayVideo(Context
   context, long videoid, String token,
   VideoPlayerConfig playerConfig) {
9.     Intent intent = new Intent(context,
   VideoPlayActivity.class);
10.    intent.putExtra(ConstantUtil.IS_OFFLINE,
   false);
```

```

11.     intent.putExtra(ConstantUtil.VIDEO_ID,
        videoid);
12.     intent.putExtra(ConstantUtil.TOKEN, token);
13.     if (playerConfig == null) {
14.         playerConfig = new VideoPlayerConfig();
15.     }
16.
        intent.putExtra(ConstantUtil.VIDEO_PLAYER_CONFIG,
            playerConfig);
17.     context.startActivity(intent);
18. }

```

点播三分屏

```

1. /**
2.  * 进入标准在线点播三分屏播放界面
3.  * @param context    context
4.  * @param videoid    点播vid
5.  * @param token      点播token
6.  * @param playerConfig 创建播放器的配置项，
        null则取默认配置项
7.  */
8.  public static void startPlayVideoTriple(Context
        context, long videoid, String token,
        VideoPlayerConfig playerConfig) {
9.     Intent intent = new Intent(context,
        VideoPlayTripleActivity.class);
10.    intent.putExtra(ConstantUtil.IS_OFFLINE,
        false);
11.    intent.putExtra(ConstantUtil.VIDEO_ID,
        videoid);
12.    intent.putExtra(ConstantUtil.TOKEN, token);
13.    if (playerConfig == null) {
14.        playerConfig = new VideoPlayerConfig();
15.    }

```

```
16.         intent.putExtra(ConstantUtil.VIDEO_PLAYER_CONFIG,
17.             playerConfig);
18.     context.startActivity(intent);
19. }
```

离线播放

```
1. /**
2.  * 进入标准离线点播播放界面
3.  * @param context      context
4.  * @param downloadModel 下载model(包含离线视频的全部信息)
5.  * @param playerConfig 创建播放器的配置项,
6.  * null则取默认配置项
7.  */
8. public static void startPlayLocalVideo(Context
9.     context, DownloadModel downloadModel,
10.     VideoPlayerConfig playerConfig) {
11.     Intent intent = new Intent(context,
12.         VideoPlayActivity.class);
13.     intent.putExtra(ConstantUtil.IS_OFFLINE,
14.         true);
15.     intent.putExtra(ConstantUtil.VIDEO_DOWNLOAD_MODEL,
16.         downloadModel);
17.     if (playerConfig == null) {
18.         playerConfig = new VideoPlayerConfig();
19.     }
20.     intent.putExtra(ConstantUtil.VIDEO_PLAYER_CONFIG,
21.         playerConfig);
22.     context.startActivity(intent);
23. }
```

3.3. 调起回放播放页面

标准回放

```
1. /**
2.  * 进入回放标准UI界面
3.  *
4.  * @param context          Activity
5.  * @param roomId          房间id
6.  * @param roomToken       token
7.  * @param sessionId       sessionId 长期
   课才需要填，非长期课默认传-1
8.  * @param onEnterPBRoomFailedListener 进房
   间错误监听，可为null
9.  */
10. public static void enterPBRoom(Context
   context, String roomId, String roomToken, String
   sessionId,
11.     OnEnterPBRoomFailedListener
   onEnterPBRoomFailedListener)
```

标准回放 with videoConfig

```
1. /**
2.  * 进入回放标准UI界面
3.  *
4.  * @param context          Activity
5.  * @param roomId          房间id
6.  * @param roomToken       token
7.  * @param sessionId       sessionId 长期
   课才需要填，非长期课默认传-1
8.  * @param playerConfig    创建播放器的
   配置项， null则取默认配置项
```

```
9.    * @param onEnterPBRoomFailedListener 进房  
    间错误监听，可为null  
10.   */  
11.   public static void enterPBRoom(Context  
    context, String roomId, String roomToken, String  
    sessionId, VideoPlayerConfig playerConfig,  
    OnEnterPBRoomFailedListener  
    onEnterPBRoomFailedListener)
```

裁剪回放

```
1.   /**  
2.   * 进入回放标准UI界面  
3.   *  
4.   * @param context          Activity  
5.   * @param roomId          房间id  
6.   * @param roomToken       token  
7.   * @param sessionId       sessionId 长期  
    课才需要填，非长期课默认传-1  
8.   * @param version         裁剪版本号（不  
    传主版本，传0原视频）  
9.   * @param playerConfig    创建播放器的  
    配置项，null则取默认配置项  
10.  * @param onEnterPBRoomFailedListener 进房  
    间错误监听，可为null  
11.  */  
12.  public static void enterPBRoom(Context  
    context, String roomId, String roomToken, String  
    sessionId, int version, VideoPlayerConfig  
    playerConfig, OnEnterPBRoomFailedListener  
    onEnterPBRoomFailedListener)
```

合并回放

```
1.  /**
```

```
2. * 合并回放，多个回放合并在一起
3. * @param context
4. * @param mixedId
5. * @param mixedToken
6. * @param onEnterPBRoomFailedListener
7. */
8. public static void enterPBRoom(Context
context, String mixedId, String mixedToken,
OnEnterPBRoomFailedListener
onEnterPBRoomFailedListener)
```

合并回放 with videoConfig

```
1. /**
2. * 合并回放
3. * @param context
4. * @param mixedId
5. * @param mixedToken
6. * @param playerConfig
7. * @param onEnterPBRoomFailedListener
8. */
9. public static void enterPBRoom(Context
context, String mixedId, String mixedToken,
VideoPlayerConfig playerConfig,
OnEnterPBRoomFailedListener
onEnterPBRoomFailedListener)
```

离线回放

```
1. /**
2. * 进入离线回放标准UI界面
3. *
4. * @param context activity
5. * @param videoModel 视频 DownloadModel
6. * @param signalModel 信令 DownloadModel
```



```
7.    */
8.    public static void enterLocalPBRoom(Context
context, DownloadModel videoModel,
DownloadModel signalModel)
```

离线回放 with videoConfig

```
1.    /**
2.     * 进入离线回放标准UI界面
3.     * @param context    activity
4.     * @param videoModel  视频 DownloadModel
5.     * @param signalModel 信令 DownloadModel
6.     * @param playerConfig 创建播放器的配置项,
null则取默认配置项
7.     */
8.    public static void enterLocalPBRoom(Context
context, DownloadModel videoModel,
DownloadModel signalModel, VideoPlayerConfig
playerConfig)
```

3.4. VideoPlayerConfig

`VideoPlayerConfig` 提供一系列配置参数供外部控制播放器行为，如下

```
1.    // 是否后台播放
2.    public boolean supportBackgroundAudio =
false;
3.    // 是否循环播放
4.    public boolean supportLooping = false;
5.    // 是否记忆播放
6.    public boolean supportBreakPointPlay = true;
7.    // 用户名
8.    public String userName;
```

```
9. // 用户ID
10. public String userId;
11. // 跑马灯
12. public LPHorseLamp horseLamp;
13. // 是否支持进度条拖动
14. public boolean supportSeek = true;
15. // 最大可拖拽观看时间
16. public int maxWatchTime =
    Integer.MAX_VALUE;
17. // 是否支持倍速播放
18. public boolean supportVideoRate = true;
19. // 是否以视频为主 (true为以视频为主)
20. public Boolean isVideoMain;
21. // 是否默认横屏
22. public Boolean isLandscape;
23. // 是否允许切换横竖屏
24. public boolean enableToggleScreen = true;
25. // 点播评论 token
26. public String loginToken;
27. // 点播评论 用户avatar链接，null则显示本地默认头像
28. public String avatar;
29. // 签到
30. public List<SignModel> signModelList;
31.
32. /**
33.  * 点播合集 or 回放合集 数据源
34.  */
35. public List<AlbumItemData>
    albumItemDataList = new ArrayList<>();
36.
37. /**
38.  * 合集默认 index
39.  */
40. public int defaultAlbumIndex;
```

其中 `userName` 和 `userId` 用户播放行为上报，建议设置。

3.5. 回放合集

外界通过 `VideoplayerConfig.setAlbumItemDataList()` 传入合集列表，`VideoplayerConfig.defaultAlbumIndex` 为默认播放合集的下标。

```
1. /**
2.  * 回放专辑: roomId + sessionId + version +
3.  * 点播合集: videoid + token + prefaceUrl + name
4.  */
5. data class AlbumItemData(
6.     @SerializedName("room_id")
7.     val roomId: Long,
8.     @SerializedName("session_id")
9.     val sessionId: Long,
10.    val version: Int,
11.    @SerializedName("play_token")
12.    val token: String,
13.    @SerializedName("vid")
14.    val videoid: Long,
15.    // 封面
16.    @SerializedName("preface_url")
17.    val prefaceUrl: String?,
18.    // 标题
19.    val name: String,
20.    @SerializedName("is_effective_expired")
21.    val isExpired: Int
22. )
```

3.6. 点播签到

点播签到功能支持在指定时间点显示指定文案，并且回调签到事件。

外界通过 `VideoplayerConfig.setSignModelList()` 传入签到列表。

每一个 `SignModel` 代表一个签到打点，字段如下

```
1. public class SignModel implements Serializable {
2.     /**
3.      * 显示签到的时间点，单位秒
4.      */
5.     public int seconds;
6.     /**
7.      * 签到弹窗 logo，null 则为默认图标
8.      */
9.     public String logo;
10.    /**
11.     * 签到标题
12.     */
13.    public String title;
14.    /**
15.     * 签到内容
16.     */
17.    public String content;
18.    /**
19.     * 签到按钮文案
20.     */
21.    public String btnText;
22. }
```

设置点播签到按钮点击回调

```
1. CallbackManager.getInstance().setSignConfigListen
   Consumer<SignModel>() {
```

```

2.     @Override
3.     public void accept(SignModel
signModel) {
4.         LLogger.d(TAG, "sign:" +
PJsonUtils.toString(signModel));
5.         Toast.makeText(LauncherActivity.this,
PJsonUtils.toString(signModel),
Toast.LENGTH_SHORT).show();
6.     }
7. };

```

3.7. 点播分享

`CallbackManager.setShareListener(ShareListener shareListener)` 设置点播分享回调后，UI 界面上会显示分享按钮（默认不显示），点击回调 `onShareClicked(String videoid)`。

```

1. public interface ShareListener {
2.     /**
3.      * 点击分享回调
4.      *
5.      * @param videoid 点播视频 id
6.      */
7.     void onShareClicked(String videoid);
8. }

```

3.8. 点播/回放试看

`VideoPlayerConfig.visitorModel` 设置试看信息，包括允许试看的时长和弹框文案。

```
1. public class VisitorModel extends
   CustomUserModel {
2.     /**
3.      * 试看时长, 单位秒
4.     */
5.     public int seconds = Integer.MAX_VALUE;
6. }
```

CustomUserModel 字段如下

```
1. public class CustomUserModel implements
   Serializable {
2.     /**
3.      * logo url
4.     */
5.     public String logoUrl;
6.     /**
7.      * logo resource id, 与 logoUrl 互斥
8.     */
9.     public int logoResId;
10.    /**
11.     * 弹窗提示标题
12.    */
13.    public String title;
14.    /**
15.     * 弹窗提示正文
16.    */
17.    public String content;
18.    /**
19.     * 确认按钮文案
20.    */
21.    public String positiveText;
22.    /**
23.     * 取消按钮文案
24.    */
```

```
25.     public String navigateText;
26. }
```

3.9. 点播/回放禁止拖拽

`VideoPlayerConfig.enableDragController = false`，默认值为 `true`，允许拖拽。

```
1. /**
2.  * 是否允许拖拽
3.  */
4. public boolean enableDragController = true;
```

禁止拖拽后，用户拖拽会弹框提示，设置

`VideoPlayerConfig.dragControllerModel` 自定义文案。
`DragControllerModel` 继承自 `CustomUserModel`，详见 3.8 节。

4. 功能介绍

包结构



4.1. 自定义点播UI

component包下提供一系列标准播放器UI组件，用户可以在ComponentManager管理类中配置需要显示的组件。

```
1. /**
```



```

2.  * 默认组合的组件
3.  */
4.  public void generateDefaultComponentList(){
5.      componentMap.clear();
6.      componentChangeListeners.clear();
7.
8.      addComponent(UIEventKey.KEY_LOADING_COMPON
9.          new LoadingComponent(context));
10.
11.      addComponent(UIEventKey.KEY_GESTURE_COMPON
12.          new GestureComponent(context));
13.      //controller 需在gesture布局上方，否则会有事
14.      件冲突
15.      addComponent(UIEventKey.KEY_CONTROLLER_COM
16.          new ControllerComponent(context));
17.
18.      addComponent(UIEventKey.KEY_ERROR_COMPONEI
19.          new ErrorComponent(context));
20.
21.      addComponent(UIEventKey.KEY_MENU_COMPONEN
22.          new MenuComponent(context));
23.      if(BJYPlayerSDK.IS_DEVELOP_MODE){
24.
25.          addComponent(UIEventKey.KEY_VIDEO_INFO_COMPF
26.              new MediaPlayerDebugInfoComponent(context));
27.      }
28.  }

```

4.2. 自定义组件

1) ComponentContainer

ComponentContainer extends FrameLayout，持有
ComponentManager初始化各个播放器组件，实现手势监听，

并且控制自定义事件和触摸事件的分发。

2) BaseVideoView

BaseVideoView extends FrameLayout, 持有 IBJYVideoPlayer 引用获取播放器回调, 持有 ComponentContainer 通知各个 component 组件更新状态, 内部实现网络监听逻辑。

3) BJYVideoView

BJYVideoView extends BaseVideoView, 初始化 BJYPlayerView 和各个 component, 监听播放器回调通知各个 component。



下载为pdf格式