

# Video++互动层Android SDK对接指南

---

Video++ 互动层目前支持所支持的业务为：

- 直播互动
- 点播互动
- OTT互动
- Video++商城
- 互动娱乐

## 快速集成

### 1.1 Android SDK Demo

SDK配置的demo代码[点击此处](#)，该Demo上面有对接的详细代码可供参考使用

- 直播互动 LiveOsActivity.java
- 点播互动 VideoOsActivity.java
- OTT互动 VideoOTTActivity.java
- Video++商城 MallOsActivity.java
- 互动娱乐 HuyuOsActivity.java

### 1.2 SDK 下载

#### 1.2.1 远程依赖

gradle 远程仓库依赖，打开项目找到 build.gradle 文件，添加 maven 仓库地址：

```
1. repositories {
2.     maven{
3.         url 'https://dl.bintray.com/video-li/maven/'
4.     }
5. }
```

添加 SDK 版本依赖：

```
1. compile 'com.video-li:video_pub:2.0.2'
```

- 如若仓库访问失败，那么用本地依赖的方式进行依赖，aar 包请找对接同学提供

添加其他三方依赖

```
1. compile 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.0.2'
2. compile 'com.github.bumptech.glide:glide:GLIDE_VERSION'
3. compile 'com.squareup.okhttp3:okhttp:OKHTTP_VERSION'
```

### 1.2.2 本地依赖导入

将 aar 包复制到 libs 目录，打开项目的 build.gradle 文件，添加 libs 路径

```
1. repositories {
2.     flatDir {
3.         dirs 'libs'
4.     }
5. }
```

添加本地依赖

```
1. compile(name: 'venvy_pub-release-2.0.2', ext: 'aar')
```

### 1.3 权限说明

video++ SDK 使用到一下权限

```
1. <uses-permission android:name="android.permission.INTERNET"/>
2. <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
3. <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
4. <uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"/>
5. <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

### 1.4 混淆规则

```

1. # Glide
2. -keep public class * implements com.bumptechn.glide.module.GlideModule
3. -keep public class * extends com.bumptechn.glide.AppGlideModule
4. -keep public enum com.bumptechn.glide.load.resource.bitmap.ImageHeaderPa
  rser$** {
5.     **[] $VALUES;
6.     public *;
7. }
8.
9. # okhttp
10. -dontwarn okio.**
11. -dontwarn com.squareup.okhttp.**
12. -dontwarn okhttp3.**
13. -dontwarn javax.annotation.Nullable
14. -dontwarn javax.annotation.ParametersAreNonnullByDefault
15.
16. # Mqtt
17. -keep class org.eclipse.paho.client.mqttv3.** {
18.     <fields>;
19.     <methods>;
20. }
21.
22. # video++
23. -keep class cn.com.venvy.keep.** {
24.     <fields>;
25.     <methods>;
26. }

```

## 2 对接准备:

在 Appliaction 的 onCreate 方法里进行如下配置:

```

1. @(文档)Override
2. public void onCreate() {
3.     super.onCreate();
4.     VenvyUIUtil.runOnUiThreadDelay(new Runnable() {
5.         @Override
6.         public void run() {
7.             // VideoType 中标识本次接入业务类型, 具体接入请根据实际的业务需求
8.             // VideoType.BOTH 表示接入点播和直播
9.             // VideoType.LIVE0S 表示只接入直播
10.            // VideoType.VIDE00S 表示只接入点播
11.            // VideoType.OTT 表示只接入OTT
12.            // VideoType.MALL 表示只接入子商城
13.            // VideoType.HUYU 表示只接入互动娱乐
14.            VideoPlus.appCreate(MyApp.this, VideoPlus.VideoType.BOTH);
15.        }
16.    }, 3000);

```

- appCreate 的第二个为可变参数，可多选

### 3 直播互动配置步骤:

#### 3.1 直播互动层创建

在xml中配置互动层的VideoLiveView:

```
1. <cn.com.videopls.pub.live.VideoLiveView
2.     android:id="@+id/venvyLive"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent" />
```

在代码中找到控件

```
1. VideoLiveView videoLiveView = (VideoLiveView) findViewById(R.id.venvyLive);
```

如果您希望在代码中手动创建直播的互动层

```
1. VideoLiveView videoLiveView = new VideoLiveView(context);
2. videoLiveView.setLayoutParams(new LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
3. addview(videoLiveView);
```

#### 3.2 初始化 Adapter

继承 VideoPlusAdapter 以创建 Adapter

```

1. private class LiveAdapter extends VideoPlusAdapter {
2.
3.     @Override
4.     public Provider createProvider() {
5.         int width = VenvyUIUtil.getScreenWidth(LiveActivity.this);
6.         int height = VenvyUIUtil.getScreenHeight(LiveActivity.this);
7.
8.         Provider provider = new Provider.Builder()
9.             .setHorVideoHeight(height) // 横屏视频的高
10.            .setHorVideoWidth(width) // 横屏视频的宽
11.            .setVerVideoHeight(screenHeightSmall) // small 视频小屏视频
            的高
12.            .setVerVideoWidth(width) // small 视频小屏视频的宽
13.            .setVerticalFullVideoWidth(height) // Full 视频全屏视频的高
14.            .setVerticalFullVideoHeight(width) // 视频全屏视频的宽
15.            .setVerticalType(1) // 1 竖屏小屏, 0 竖屏全屏
16.            .setDirection(2) // 2 横竖屏, 0 竖屏, 1 是横屏
17.            .build();
18.         return provider;
19.     }
20. }

```

- createProvider 方法用来设置互动层各种情况下的显示大小
- VenvyUIUtil.getScreenWidth 、 VenvyUIUtil.getScreenHeight 为获取屏幕宽高的方法，可自行选择调用
- **注意：** 如果进入直播间默认为横屏状态，请调用 VenvyUIUtil.getVerticalScreenHeight 、 VenvyUIUtil.getVerticalScreenWidth 获取宽高

将Adapter对象设置给 VideoLiveView:

```

1. LiveAdapter liveAdatper = new LiveAdapter();
2. videoLiveView.setVideoOSAdapter(liveAdatper);

```

更新 Adapter

```

1. videoPlusView.stop();
2. Provider provider = new Provider.Builder()
3.     .setHorVideoWidth(Math.max(mScreenWidth, mScreenHeight))
4.     .setHorVideoHeight(Math.min(mScreenWidth, mScreenHeight))
5.     .setVerticalFullVideoWidth(Math.min(mScreenWidth, mScreenHeight))
6.     .setVerticalFullVideoHeight(Math.max(mScreenWidth, mScreenHeight))
7.     .setVerVideoWidth(Math.min(mScreenWidth, mScreenHeight))
8.     .setVerVideoHeight(mWidowPlayerHeight)
9.     .setVerticalType(0)
10.    .setDirection(2)
11.    .build();
12. liveAdapter.updateProvider(provider);
13. videoPlusView.start();

```

- liveAdapter.updateProvider(Provider provider) 可以更新 provider 中的设置参数
- 在调用 updateProvider 之前需要调用 videoLiveView.stop(), update 之后调用 videoLiveView.start();
- 当然你也可以将 prvider 提取出成员变量

### 3.3 直播互动开启

```

1. videoLiveView.start();

```

### 3.4 直播互动关闭

当页面 (Activity) 销毁的时候, 需要在 onDestroy() 方法里面销毁直播互动

```

1. @Override
2. protected void onDestroy() {
3.     super.onDestroy();
4.     if (videoLiveView != null) {
5.         videoLiveView.destroy();
6.     }
7. }

```

### 3.5 传入用户信息

在 Adapter 中重写 buildLoginInterface 方法

```
1. @Override
2. public IPlatformLoginInterface buildLoginInterface() {
3.     return new PlatformUserInfoImpl() {
4.         @Override
5.         public PlatformUserInfo getLoginUser() {
6.             PlatformUserInfo userInfo = new PlatformUserInfo();
7.             userInfo.roomId = ""; // 房间号 (必要)
8.             userInfo.platformId = ""; // platformId (必要)
9.             userInfo.uid = ""; // 设置用户ID
10.            userInfo.cate = ""; // 设置分区
11.            userInfo.userName = ""; // 设置用户名
12.            userInfo.isAnchor = true; // 是否为主播端
13.            userInfo.isPush = true; // 是否正在推流
14.            return userInfo;
15.        }
16.    };
17. }
```

### 3.6 中插视频广告事件监听

在 Adaptor 中重写 buildWedgeListener，返回 WedgeListener 对象，选择需要的方法重写

```
1.      /**
2.       * 中插开始播放
3.       */
4.     void onStart();
5.
6.     /**
7.      * 中插播放完毕通知，播放完毕分3种状态：
8.      * 1:中插视频自然播放到指定时间长度
9.      * 2:中插在播放期间由后台广告管理撤销当前中插的播放，导致当前中插页面关闭
10.     * 3:中插在播放时间，配置的展示时间段过期
11.     */
12.    void onFinish();
13.
14.    /**
15.     * 没有中插广告
16.     */
17.    void onEmpty();
18.
19.    /**
20.     * 点击中插返回按钮
21.     */
22.    void goBack();
23.
24.    /**
25.     * 恢复播放
26.     */
27.    void onResume();
28.
29.    /**
30.     * 暂停播放
31.     */
32.    void onPause();
33.
34.    /**
35.     *
36.     * @return 从后台返回后是否需要显示中插
37.     */
38.    boolean needShowVideo();
39.
40.    /**
41.     *
42.     * @return 设置中插缓存的最大值 单位：M
43.     */
44.    boolean downloadSwitch();
45.
46.    /**
47.     *
48.     * @return 移动网络是否下载视频 true:开启下载，false 关闭下载
49.     */
50.    int maxCacheSize();
```



以上是直播接入步骤，如果有不清楚的地方可以咨询对接的同学

#### 4.点播互动配置步骤:

##### 4.1 在xml中配置互动层的 VideoOsView:

```
1. <cn.com.videoplus.pub.os.VideoOsView
2.     android:id="@+id/video"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent" />
```

##### 4.2 初始化点播互动层 VideoOsView 对象的 Adapter:

```
1. private class MyAdapter extends VideoPlusAdapter {
2.
3.     @Override
4.     public Provider createProvider() {
5.         int width = VenvyUIUtil.getScreenWidth(MainActivity.this);
6.         int height = VenvyUIUtil.getScreenHeight(MainActivity.this);
7.         Provider provider = new Provider.Builder()
8.             .setAppKey(getAppKey()) // appkey
9.             .setHorVideoHeight(Math.min(width, height)) // 横屏视频的高
10.            .setHorVideoWidth(Math.max(width, height)) // 横屏视频的宽
11.            .setVerVideoWidth(Math.min(width, height)) // small 视频小屏
   视频的宽
12.            .setVerVideoHeight(screenHeightSmall) // small 视频小屏视频
   的高
13.            .setVideoPath(getVideoPath()) // 视频地址
14.            .setVideoType(3) //
15.            .setVideoTitle("ttt") // 视频标题 可选
16.            .build();
17.         return provider;
18.     }
19.
20.     @Override
21.     public IMediaController buildMediaController() {
22.         return new MyMediaController();
23.     }
24. }
25.
```

由于点播互动需要获得视频的播放时间，所以在接入点播互动构造 Adapter 的时候 buildMediaController 法必须重载，其中 getCurrentPosition 方法为获取当前播放器时间，单位为毫秒级。其他方法建议实现（比如start(), pause() 等等，方便以后我们SDK做扩展，我们目前没有对接入方的播放器做控制。

##### 4.3 将4.2的Adapter对象设置给VideoOsView:

```
1. videoOsView = (VideoOsView) findViewById(R.id.video);
2. adapter = new MyAdapter();
3. videoOsView.setVideoOSAdapter(adapter);
4. adapter.onCreate();
```

4.4 在页面Activity的生命周期中调用点播Adapter对应的生命周期：(如果您有查看我们的demo，请关注下点播页面VideoOsActivity的父类BasePlayerActivity的生命周期调用)

```
1. @Override
2.     protected void onStart() {
3.         super.onStart();
4.         //开启点播互动
5.         adapter.onStart();
6.
7.     }
8.
9.     @Override
10.    protected void onPause() {
11.        super.onPause();
12.        //暂停点播互动
13.        adapter.onPause();
14.
15.    }
16.
17.    @Override
18.    protected void onDestroy() {
19.        super.onDestroy();
20.        //销毁点播互动
21.        adapter.onDestroy();
22.    }
23.
24.    @Override
25.    protected void onResume() {
26.        super.onResume();
27.        //点播互动resume
28.        adapter.onResume();
29.    }
```

## 5.OTT互动配置步骤

OTT互动的配置与点播基本相同（请详细参照步骤4.X以及DEMO代码VideoOTTActivity），请将VideoOsView替换为VideoOTTView

## 6.子商城配置

### 6.1.1 在xml布局文件中添加子商城布局

```
1. <cn.com.videopls.pub.mall.VideoMallView
2.     android:id="@+id/venvyMall"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent" />
```

### 6.1.2 初始化VideoMallView并配置adapter

```
1. videoMallView = (VideoMallView) findViewById(R.id.venvyMall);
2. mallAdapter = new MallAdapter();
3. videoMallView.setVideoOSAdapter(mallAdapter);
4. videoMallView.start();
5.
```

### 6.1.3 继承 EnjoyAdapter 以创建 D

```

1. private class MallAdapter extends EnJoyAdapter {
2.     @Override
3.     public IPlatformLoginInterface buildLoginInterface() {
4.         return new PlatFormUserInfoImpl() {
5.             @Override
6.             public PlatformUserInfo getLoginUser() {
7.                 PlatformUserInfo userInfo = new PlatformUserInfo();
8.                 userInfo.roomId = ""; // 房间号 (必要)
9.                 userInfo.platformId = ""; // platformId (必要)
10.                platformUserInfo.uid = ""; // 用户唯一标志
11.                platformUserInfo.nickName = ""; // 用户昵称
12.                platformUserInfo.userName = ""; // 用户名
13.                platformUserInfo.phoneNum = ""; // 电话号码
14.                platformUserInfo.userToken = ""; // token信息
15.                return userInfo;
16.            }
17.            @Override
18.            public void userLoggedIn(PlatformUserInfo userInfo) {
19.                if (userInfo != null) {
20.                    String token = userInfo.getUserToken();
21.                }
22.            }
23.
24.            /**
25.             * 商城h5页面用户登陆, 回调给第三方app
26.             */
27.            @Override
28.            public void login(LoginCallback loginCallback) {
29.
30.            }
31.
32.            /**
33.             * h5 点击立即购买等需要转竖屏交互时通知app
34.             */
35.            @Override
36.            public void screenChanged(ScreenChangedInfo changedInfo) {
37.
38.            }
39.        };
40.    }

```

#### 6.1.4 打开子商城页面的方式

点击子商城入口（子商城入口需要第三方app自己配置),当点击入口按钮呼出子商城之后，入口需要隐藏。

```

1. doorBtn.setOnClickListener(new View.OnClickListener() {
2.     @Override
3.     public void onClick(View v) {
4.         //点击响应路由，呼出子商城
5.         Router.route(Scheme.MAILL);
6.
7.         //隐藏商城入口按钮
8.         doorBtn.setVisibility(View.GONE);
9.     }
10. });
11.

```

### 6.1.5 关闭子商城回调

当子商城页面关闭的时候需要通知第三方，第三方在接到回调通知后将入口显示，具体的回调在Adapter里面配置，其代码如下：

```

1.     @Override
2.     public IWidgetCloseListener<WidgetInfo> buildWidgetCloseListen
3.     er() {
4.         return new IWidgetCloseListener<WidgetInfo>() {
5.             @Override
6.             public void onClose(WidgetInfo widgetInfo) {
7.                 WidgetInfo.WidgetType widgetType = widgetInfo.getWi
8.                 dgetType();
9.                 if(widgetType == WidgetInfo.WidgetType.MAILL) { //子商
10.                    城页面关闭
11.                    //显示子商城入口
12.                    doorBtn.setVisibility(View.VISIABLE);
13.                }
14.            }
15.        }
16.    }
17.

```

## 6.2 竖屏小屏幕子商城配置：

6.2.1在竖屏小屏幕状态下，需要在播放器下面配置SDK提供的MallWebView布局页面( 其具体类路径为cn.com.venvy.mall.view.MallWebView) ,用来打开子商城h5页面,另外初始化WebView完毕后需要设置如下配置：

```

1. //也可以在xml里配置
2. MallWebView mallWebView = new MallWebView(context);
3. //步骤1创建的IPlatformLoginInterface对象
4. mallWebView.setPlatformLoginInterface(mPlatformLoginInterface);
5.

```

6.2.2 注意在竖屏状态下响应screenChanged里面添加如下代码：

```

1. public void screenChanged(changedInfo changedInfo) {
2.     //url为小屏幕URL
3.     mallWebView.setSsId(changedInfo.getSsid());
4.     mallWebView.setVisibility(View.VISIBLE);
5.     mallWebView.loadUrl(changedInfo.getUrl());
6. }
7.

```

6.2.3 在竖屏点击入口，打开MallWebView调用loadUrl方法来获取商品列表：

```

1. //获取商品货架的URL
2. String mallUrl=MallConfig.getMallUrl();
3. mallWebView.setSsId(System.currentTimeMillis()+"");
4. //拼接第三方房间号
5. mallWebView.loadUrl(mallUrl+"?video="+ 第三方app的房间ID)
6.

```

### 6.3 我的订单

根据产品定义，需要在第三方app的用户信息页面配置“我的订单”选项，当用户点击此选项的时候业务逻辑如下：

- 1、判断第三方app的用户是否登录，如果没有登录则跳转到第三方的登录页面，进行用户的登录操作。
- 2、跳转独立的activity，该activity的contentView为全屏竖屏的VenvyWebView，该activity的配置示例代码请参考Demo中MyOrderActivity。

### 6.4 Webview的初始化工作

因为sdk的webView用的是腾讯的x5webView,所以对其初始化需在Appcation中做如下配置：所以需要libs文件配置x5WebView的jar，同时需要在src/jniLibs文件放x5的so库，x5jar包[点击此处](#)

```

1. MyApp extends Application {
2.     @Override
3.     public void onCreate() {
4.         super.onCreate();
5.         VideoPlus.appCreate(MyApp.this,VideoPlus.VideoType.MALL);
6.
7.     }
8.

```

### 7.互娱配置步骤

相关 View 的设置同直播，继承 EnjoyAdapter 以创建 Adapter

```

1. class MyAdapter extends EnjoyAdapter {
2.
3.     @Override
4.     public IPlatformLoginInterface buildLoginInterface() {
5.         return new PlatFormUserInfoImpl() {
6.             @Override
7.             public PlatformUserInfo getLoginUser() {
8.                 PlatformUserInfo platformUserInfo = new PlatformUserInf
9. o();
10.                 // 是否为主播
11.                 platformUserInfo.isAnchor = true;
12.                 // 是否正在推流
13.                 platformUserInfo.isPush = true;
14.                 // 用户 id
15.                 platformUserInfo.uid = "";
16.                 // 房间号
17.                 platformUserInfo.roomId = "";
18.                 // 平台 id
19.                 platformUserInfo.platformId = "";
20.                 return platformUserInfo;
21.             }
22.         };
23.     }
24. }

```

#### 8.互动的相关监听方法:

如果您想要对互动层上面所展示的广告行为做监听，比如展示、点击、关闭，则可以按照需求复写 Adapter的buildWidgetShowListener、buildWidgetClickListener、buildWidgetCloseListener方法，如下所示：

```
1. public IWidgetClickListener<WidgetInfo> buildWidgetClickListener() {
2.     return new IWidgetClickListener<WidgetInfo>() {
3.         @Override
4.         public void onClick(@Nullable WidgetInfo widgetInfo) {
5.             //点击监听
6.         }
7.     };
8. }
9.
10. public IWidgetShowListener<WidgetInfo> buildWidgetShowListener() {
11.     return new IWidgetShowListener<WidgetInfo>() {
12.         @Override
13.         public void onShow(WidgetInfo widgetInfo) {
14.             //展示监听
15.         }
16.     };
17. }
18.
19. public IWidgetCloseListener<WidgetInfo> buildWidgetCloseListener() {
20.     return new IWidgetCloseListener<WidgetInfo>() {
21.         @Override
22.         public void onClose(WidgetInfo widgetInfo) {
23.             //关闭监听
24.         }
25.     };
26. }
```

通过 `widgetInfo.getWidgetType()` 获取广告的类型



```
1. public enum WidgetType {
2.     TAG, //热点类型
3.     CLOUD, // 互娱配置页
4.     INFO, //link, 或者信息层
5.     VIDEOCLIP, //中插
6.     NEWS, //新闻
7.     LOTTERY, //抽奖
8.     BUBBLE, //气泡
9.     EASYSHOP, //轻松购
10.    BASICWIKI, //百科
11.    BASICMIX, //轮播广告
12.    ADINFO, //海报
13.    VOTE, //投票
14.    COUPONGIFT, //优惠券红包
15.    CODEGIFT, //口令红包
16.    ADGIFT, //视频红包（二维码红包）
17.    CARDGAME, //卡牌
18.    GIFT, //红包
19.    TEXT, //文字链
20.    GOODS, //商品
21.    PRAISE, //点赞
22.    PIC, //图片广告
23.    MAILL //子商城
24. }
```