

Video++互动层Android SDK对接指南

Video++ 互动层目前支持直播和点播功能的同时集成。其中具体集成直播、点播、或者直播点播融合的版本请先和商务同学确认。

对接方式说明：

- 1.SDK配置的demo代码[点击此处](#)，该Demo上面有直播(demo中LiveActivity.java)和点播(MainActivity.java)对接的详细代码可供参考使用
- 2.除了参考上面的Demo中的配置外，还可以参考下文“对接步骤说明”

对接步骤说明：

1. 在项目 build.gradle 中配置相关的compile.

```
1. dependencies {
2.     compile fileTree(dir: 'libs', include: ['*.jar'])
3.
4.     //Video++ 互动层依赖:(版本号只是举例，具体版本请咨询对接技术同学)
5.     compile 'com.videoli:video_pub:1.5.0.4'
6.
7. }
```

- 2.在AndroidManifest.xml中配置如下权限：

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

- 3.直播互动配置步骤：

- 3.1 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" />

```

3.2 初始化直播互动层VideoLiveView对象的Adapter :

```

1.  private class LiveAdapter extends VideoPlusAdapter {
2.
3.         @Override
4.         public Provider createProvider() {
5.             final int width = VenvyUIUtil.getScreenWidth(LiveActi
6. vity.this);
7.             final int height =VenvyUIUtil.getScreenHeight(LiveAct
8. ivity.this);
9.
10.            Provider provider = new Provider.Builder()
11.                .setUserId(mRoomId)//roomId 或者userId
12.                .setPlatformId(getPlatformId())//videojj直播后
13.                .setHorVideoHeight(height)//横屏视频的高
14.                .setHorVideoWidth(width)//横屏视频的宽
15.                .setVerVideoHeight(screenHeightSmall)//small
16.                .setVerVideoWidth(width)//small视频小屏视频的宽
17.                .setVerticalFullVideoWidth(height)//Full 视频
18.                .setVerticalFullVideoHeight(width)//视频全屏视
19.                .setVerticalType(1)//1 竖屏小屏, 0竖屏全屏
20.                .setDirection(2) //2横竖屏, 0竖屏, 1是横屏
21.                .setIsMango()
22.                .build();
23.            return provider;
24.        }
25.    }

```

3.3 将4.2的Adapter对象设置给VideoLiveView :

```
1.         videoLiveView = (VideoLiveView) findViewById(R.id.venvyLi  
        ve);  
2.         liveAdatper = new LiveAdapter();  
3.         videoLiveView.setVideo0SAdapter(liveAdatper);
```

3.4 直播互动开启：

```
1.         videoLiveView.start();
```

3.5 直播互动关闭：

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

```
1.     protected void onDestroy() {  
2.         super.onDestroy();  
3.         // 销毁直播互动  
4.         videoLiveView.destroy();  
5.     }
```

4. 点播互动配置步骤

4.1 在xml中配置互动层的VideoOsView:

```
1.     <cn.com.videopls.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.
6.         final int width = VenvyUIUtil.getScreenWidth(MainActi
vity.this);
7.         final int height = VenvyUIUtil.getScreenHeight(MainAc
tivity.this);
8.         Provider provider = new Provider.Builder()
9.             .setAppKey(getAppKey())//appkey
10.            .setHorVideoHeight(Math.min(width, heigh
t))//横屏视频的高
11.            .setHorVideoWidth(Math.max(width, height))//横
屏视频的宽
12.            .setVerVideoWidth(Math.min(width, height))//s
mall视频小屏视频的宽
13.            .setVerVideoHeight(screenHeightSmall)//small
视频小屏视频的高
14.            .setVideoPath(getVideoPath())//视频地址
15.            .setVideoType(3)//
16.            .setVideoTitle("ttt")//
17.            .build();
18.         return provider;
19.     }
20.
21.     @Override
22.     public IMediaControlListener buildMediaController() {
23.         return new MyMediaController();
24.     }
25. }
26.

```

与直播配置Adapter不用的是，在点播构造Adapter的时候buildMediaController（）方法必须重载，此方法返回接口为控制播放器行为。其中getCurrentPosition方法为获取当前播放器时间，单位为毫秒级。

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：

```

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.         //销毁调拨互动
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 互动的展示、点击、关闭监听：

如果客户端想要对互动层的展示、点击、关闭做监听可以复写Adapter的 buildWidgeShowListener、buildWidgetClickListener、buildWidgetCloseListener方法，如下所示：

```

1.      public IWidgetClickListener<WidgetInfo> buildWidgetClickListener() {
2.          return new IWidgetClickListener<WidgetInfo>() {
3.
4.              @Override
5.              public void onClick(@Nullable WidgetInfo widgetInfo) {
6.                  // 点击监听
7.              }
8.          };
9.      }
10.
11.     public IWidgetShowListener<WidgetInfo> buildWidgetShowListener() {
12.         return new IWidgetShowListener<WidgetInfo>() {
13.             @Override
14.             public void onShow(WidgetInfo widgetInfo) {
15.                 // 展示监听
16.             }
17.         };
18.     }
19.
20.     public IWidgetCloseListener<WidgetInfo> buildWidgetCloseListener() {
21.         return new IWidgetCloseListener<WidgetInfo>() {
22.             @Override
23.             public void onClose(WidgetInfo widgetInfo) {
24.                 // 关闭监听
25.             }
26.         };
27.     }

```

注意事项：

Video++互动层对于三方的引入，为了避免冲突和重复引入，导入三方功能开放给最外层调用，如项目中已配置，可按照项目中原有配置来，如没配置相关依赖，需要在项目中添加。具体依赖可看build.gradle文件注释说明。

V1.2.0 版本升级内容

1. android支持远程依赖接入
2. android直播优化内存占用
3. android点播优化内存占用
4. android点播横竖屏切换曝光投递优化

V1.2.0.3 版本升级内容

5. 增加中间层Provider动态更新功能
6. 增加图片下载成功监控

v1.2.0.4 版本升级内容

7. 将SDK编译版本下调，同时将依赖项目暴露给调用方
8. Fix 点播读取文件BUG

V1.2.0.6 版本升级内容

9. fix 部分机型上闪退的bug

v1.2.0.7版本内容

10. 修复config接口中返回的对外链处理