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。

1. 对接方式说明

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

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”

2. 对接步骤说明

2.1 在appliaction的onCreate方法里进行如下配置

```
Application {

    @Override
    public void onCreate() {
        super.onCreate();
        VenvyUIUtil.runOnUIThreadDelay(new Runnable() {
            @Override
            public void run() {
                // VideoType 中标识本次接入业务类型，具体接入请咨询商务同学
                // VideoType.BOTH 表示接入点播和直播
                // VideoType.LIVEOS 表示只接入直播
                // VideoType.VIDEOOS 表示只接入点播
                // VideoType.OTT 表示只接入OTT
                VideoPlus.appCreate(MyApp.this, VideoPlus.VideoType.BOTH);
            }
        }, 3000);
    }
```

- 另外如果app对接的是1.8.0.x版本的话，因1.8.0.x用x5webView,所以需要在Application里面进行如下配置：

```
@Override
public void onCreate() {
    super.onCreate();

    initX5();
}

private void initX5() {
    Intent intent = new Intent(this, AdvanceLoadX5Service.class);
    startService(intent);
}
```

2.2 在项目 build.gradle 中配置相关的compile.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])

    //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)
    compile 'com.videoli:video_pub:1.5.0.4'

}
```

2.3 在AndroidManifest.xml中配置如下权限

```
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

3. 直播互动配置步骤

3.1 直播互动层创建

- 在 xml 文件中配置互动层

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

- 在代码中找到控件

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

- 在代码中手动创建

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

3.2 初始化 Adapter

- 创建 Adapter

```
 private class LiveAdapter extends VideoPlusAdapter {

    @Override
    public Provider createProvider() {
    final int width = VenvyUIUtil.getVerticalScreenWidth(LiveActivity.this);
    final int height =VenvyUIUtil.getVerticalScreenHeight(LiveActivity.this);

    Provider provider = new Provider.Builder()
        .setUserId(mRoomId)//roomId 或者userId
        .setPlatformId(getPlatformId())//videojj直播后台平台Id
        .setHorVideoHeight(height)//横屏视频的高
        .setHorVideoWidth(width)//横屏视频的宽
        .setVerVideoHeight(screenHeightSmall)//small 视频小屏视频的高
        .setVerVideoWidth(width)//small视频小屏视频的宽
        .setVerticalFullVideoWidth(height)//Full 视频全屏视频的高
        .setVerticalFullVideoHeight(width)//视频全屏视屏的宽
        .setVerticalType(1)//1 竖屏小屏，0竖屏全屏
        .setDirection(2) //2横竖屏，0竖屏，1是横屏
        .build();
    return provider;
    }
}

```

- 将 Adapter 对象设置给 VideoLiveView

```
LiveAdapter liveAdatper = new LiveAdapter();
videoLiveView.setVideoOSAdapter(liveAdatper);
```

3.3 直播互动开启

```
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

```
protected void onDestroy() {
    super.onDestroy();
    //销毁直播互动
    if (videoLiveView != null) {
        videoLiveView.destroy();
    }
}
```

3.6 抽奖及分区功能的使用

- 使用抽奖功能需要用户登录，传入用户 ID，用户名，在收到抽奖广告后即可参与抽奖
- 使用分区也需要传入对应的分区名
- 以上配置请在 Adapter 中重写 buildLoginInterface() 方法

```
@Override
public IPlatformLoginInterface buildLoginInterface() {
    return new PlatFormUserInfoImpl() {
        @Override
        public PlatformUserInfo getLoginUser() {
            PlatformUserInfo userInfo = new PlatformUserInfo();
            userInfo.cate = mCate; // 设置分区 e.g. lol, hearthston
e, dota1 ...
            userInfo.setUid(userId); // 用户 id
            userInfo.setUserName(userName); // 用户名
            return userInfo;
        }
    };
}
```

4. 点播互动配置步骤

4.1 点播互动层创建

- 在xml中配置互动层的VideoOsView：

```
 <cn.com.videopls.pub.os.VideoOsView
     android:id="@+id/video"
     android:layout_width="match_parent"
     android:layout_height="match_parent" />
```

- 在代码中找到控件

```
VideoOsView videoOsView = (VideoOsView) findViewById(R.id.video);
```

- 手动创建 VideoOsView :

```
VideoOsView videoOsView = new VideoOsView(context);
videoOsView.setLayoutParams(new LayoutParams(Layoutparams.MATCH_PARENT,Layoutparams.MATCH_PARENT));
addview(videoOsView);
```

4.2 创建 Adapter

```
private class MyAdapter extends VideoPlusAdapter {

    @Override
    public Provider createProvider() {

        final int width = VenvyUIUtil.getScreenWidth(context);
        final int height = VenvyUIUtil.getScreenHeight(context);
        Provider provider = new Provider.Builder()
                .setAppKey(String appkey) // appkey
                .setHorVideoHeight(Math.min(width, height)) // 横屏视频的高
                .setHorVideoWidth(Math.max(width, height)) // 横屏视频的宽
                .setVerVideoWidth(Math.min(width, height)) // small视频小屏视频的宽
                .setVerVideoHeight(screenHeightSmall) //small 视频小屏视频的高
                .setVideoPath(String url) // 视频地址
                .setVideoType(3) // 固定格式
                .setVideoTitle(String videoTitle) // 视频标题 不必须
                .build();
        return provider;
        }

        @Override
        public IMediaControlListener buildMediaController() {
            // 点播必须实现
            return new MyMediaController();
        }
}

```

- 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 Adapter 对象设置给 VideoOsView

```
MyAdapter myAdapter = new MyAdapter();
videoOsView.setVideoOSAdapter(myAdapter);
// 此方法为开启互动层 ， 请根据实际业务在适当的位置开启
videoOsView.Start();
myAdapter.onCreate();
```

4.4 切换视频 (更新 Adapter)

```
// 重置互动层 ， 清空上个视频的互动
videoOsView.stop();
Provider provider = new Provider.Builder()
        .setVideoPath(url)//重新视频地址
        .build();
// 更新配置信息
myAdapter.updateProvider(provider);
// 重新开启互动
videoOsView.start();
```

- adapter.updateProvider(provider) 亦可更新 view 的宽高

```
// 更新宽高不需要调用 stop() ， 可及时生效
Provider provider = new Provider.Builder()
    .setHorVideoHeight(width)//横屏视频的高
    .setHorVideoWidth(height)//横屏视频的宽
    .setVerVideoWidth(height)//small视频小屏视频的宽
    .setVerVideoHeight(screenHeightSmall)//small 视频小屏视频的高
    .build();
// 更新配置信息
myAdapter.updateProvider(provider);
```

4.5 在页面 Activity 的生命周期中调用点播 Adapter 对应的生命周期：

```java
@Override
protected void onStart() {
    super.onStart();
    if (myAdapter != null) {
        myAdapter.onStart();
    }
}

@Override
protected void onPause() {
    super.onPause();
    if (myAdapter != null) {
        myAdapter.onPause();
    }
}

@Override
protected void onResume() {
    super.onResume();
    if (myAdapter != null) {
        myAdapter.onResume();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    // 销毁点播互动
    if (videoOsView != null) {
        videoOsView.destroy();
    }
    if (myAdapter != null) {
        myAdapter.onDestroy();
    }
}
```

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：

```
public IWidgetClickListener<WidgetInfo> buildWidgetClickListener() {
    return new IWidgetClickListener<WidgetInfo>() {
        @Override
        public void onClick(@Nullable WidgetInfo widgetInfo) {
            //点击监听
        }
    };
}

public IWidgetShowListener<WidgetInfo> buildWidgetShowListener() {
    return new IWidgetShowListener<WidgetInfo>() {
        @Override
        public void onShow(WidgetInfo widgetInfo) {
            //展示监听
        }
    };
}

public IWidgetCloseListener<WidgetInfo> buildWidgetCloseListener() {
    return new IWidgetCloseListener<WidgetInfo>() {
        @Override
        public void onClose(WidgetInfo widgetInfo) {
            //关闭监听
        }
    };
}
```

6. 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