

- 1 阅读准备
- 2 环境要求
- 3 目录结构
- 4 场景模块集成
  - 4.1 导入通用模块(必须)
  - 4.2 云课堂场景模块集成(可选)
  - 4.3 直播带货场景模块集成(可选)
  - 4.4 手机开播场景模块集成(可选)
  - 4.5 纯视频开播场景模块集成(可选)
- 5 工程配置
  - 5.1 配置模块编译版本
  - 5.2 配置maven地址
  - 5.3 配置abiFilters
  - 5.4 配置sdkVersion
  - 5.5 配置renderScript
  - 5.6 配置方法数超过 64K 的应用
- 6 初始化sdk
- 7 登录直播/回放及进入不同场景
  - 7.1 观看端（云课堂场景/直播带货场景）
  - 7.2 开播端（手机开播场景）

## 1 阅读准备

提前下载保利威多场景Demo，请先下载Demo。

准备好一个保利威账号，Demo中登录直播需要账号直播系统中的 `appId`（应用ID），`appSecret`（应用密匙），`userId`（账号ID），`channelId`（频道号），登录回放再额外需要频道号对应回放列表里的 `vid`（回放视频id）。

## 2 环境要求

| 名称                            | 要求                                 |
|-------------------------------|------------------------------------|
| <code>minSdkVersion</code>    | <code>&gt;=21</code> (Android 5.0) |
| <code>targetSdkVersion</code> | <code>27</code> (Android 8.1)      |

| 名称          | 要求                        |
|-------------|---------------------------|
| abiFilters  | arm64-v8a、armeabi-v7a、x86 |
| jdk version | >=1.8.0                   |
| 集成工具        | Android Studio            |

注1：开发者如需兼容minSdkVersion >= 16 (Android 4.1设备)，请按[OkHttp4.x版本降级到3.x兼容方案](#)处理。

注2：如果要兼容targetSdkVersion至28，需要在AndroidManifest.xml文件中添加 android:usesCleartextTraffic="true"。

注3：so库的配置只能使用[arm64-v8a、armeabi-v7a、x86]的一种或几种，否则可能在部分机型上出现崩溃或者音画不同步、跳秒播放等情况。

注4：如需targetSdkVersion设置为29或以上，且需要访问手机外部存储公共目录下的文件，需要在AndroidManifest声明 requestLegacyExternalStorage

```
<application
    android:requestLegacyExternalStorage="true">
    ...
</application>
```

注5：如需targetSdkVersion设置为30或以上，且需要访问手机外部存储公共目录下的文件，需要在AndroidManifest声明管理外部存储权限 android.permission.MANAGE\_EXTERNAL\_STORAGE

```
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
```

并且需要在运行时动态判断是否已经给予了该权限

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
    if (!Environment.isExternalStorageManager()) {
        Intent intent = new Intent(android.provider.Settings.ACTION_MANAGE_ALL_FILES_ACCESS_PERMISSION);
        activity.startActivity(intent);
    }
}
```

注6：1.7.2及以前版本，如需targetSdkVersion设置为30或以上，需要关闭指针标记功能（[Android官方关于指针标记的说明](#)），参考以下代码修改AndroidManifest.xml以关闭指针标记。

```
<application
    android:allowNativeHeapPointerTagging="false">
    ...
</application>
```

### 3 目录结构

| 模块名                      | 作用                           |
|--------------------------|------------------------------|
| demo                     | 演示如何初始化sdk、以及登录直播/回放及进入不同场景  |
| polyvLiveCloudClassScene | 云课堂场景模块，包含播放、聊天、连麦、互动等功能的演示  |
| polyvLiveEcommerceScene  | 直播带货场景模块，包含播放、聊天、商品、打赏等功能的演示 |
| polyvLiveStreamerScene   | 手机开播场景模块，包含推流、连麦、聊天、文档等功能的演示 |
| polyvStreamerAloneScene  | 纯视频开播场景模块，包含推流、连麦、聊天等功能的演示   |
| polyvLiveCommonModul     | 通用模块，提供给各个场景实现所需依赖的基础库、工具类等  |

### 4 场景模块集成

#### 4.1 导入通用模块(必须)

拷贝polyv demo项目中的 polyvLiveCommonModul 模块至您项目的根目录下，接着打开您项目的 settings.gradle 文件，添加如下代码：

```
include ':polyvLiveCommonModul'
```

#### 4.2 云课堂场景模块集成(可选)

拷贝polyv demo项目中的 polyvLiveCloudClassScene 模块至您项目的根目录下，接着打开您项目的 settings.gradle 文件，添加如下代码：

```
include ':polyvLiveCloudClassScene'
```

### 4.3 直播带货场景模块集成(可选)

拷贝polyv demo项目中的 `polyvLiveEcommerceScene` 模块至您项目的根目录下，接着打开您项目的 `settings.gradle` 文件，添加如下代码：

```
include ':polyvLiveEcommerceScene'
```

### 4.4 手机开播场景模块集成(可选)

拷贝polyv demo项目中的 `polyvLiveStreamerScene` 模块至您项目的根目录下，接着打开您项目的 `settings.gradle` 文件，添加如下代码：

```
include ':polyvLiveStreamerScene'
```

### 4.5 纯视频开播场景模块集成(可选)

拷贝polyv demo项目中的 `polyvStreamerAloneScene` 模块至您项目的根目录下，接着打开您项目的 `settings.gradle` 文件，添加如下代码：

```
include ':polyvStreamerAloneScene'
```

## 5 工程配置

### 5.1 配置模块编译版本

打开您项目的 `build.gradle` 文件，添加如下代码：

```
ext {  
    compileSdkVersion = 27  
    minSdkVersion = 16  
    targetSdkVersion = 27  
}
```

### 5.2 配置maven地址

打开您项目的 `build.gradle` 文件，添加如下代码：

```

allprojects{
    repositories {
        maven { url "https://jitpack.io" }
        maven { url 'http://maven.aliyun.com/nexus/content/repositories/releases/' }
        mavenCentral()
        maven { url 'https://maven.aliyun.com/repository/public' }
        // 保利威阿里云效
        maven {
            credentials {
                username '609cc5623a10edbf36da9615'
                password 'EbkbzTNHRJ=P'
            }
            url 'https://packages.aliyun.com/maven/repository/2102846-release-8EVsoM/'
        }
    }
}

```

### 5.3 配置abiFilters

打开您项目的 `app/build.gradle` 文件，添加如下代码：

```

//依赖的包可能支持很多类型的ABI
//为了避免打包了我们没有支持的ABI，指定需要打包的ABI目录
ndk {
    abiFilters 'arm64-v8a', "armeabi-v7a", "x86"
}

```

### 5.4 配置sdkVersion

打开您项目的 `app/build.gradle` 文件，把里面的 `minSdkVersion` 设置为21或以上，把 `targetSdkVersion` 设置为27或以下。

如果要兼容 `minSdkVersion` 为16，可以参考[OkHttp4.x版本降级到3.x兼容方案](#)处理。

如果要兼容 `targetSdkVersion` 至28，需要在 `AndroidManifest.xml` 文件中添加

```
android:usesCleartextTraffic="true"。
```

### 5.5 配置renderScript

RenderScript 是在polyv demo项目中高斯模糊背景使用的，demo中为了兼容到api 16，使用到了 `support` 包中的 `RenderScript`。如果您项目的 `minSdkVersion` 为16，并且需要使用高斯模糊功能，则需要在 `app/build.gradle` 文件中添加如下配置：

```
renderscriptSupportModeEnabled true
```

如果您项目的 `minSdkVersion` 大于 16，或者不需要使用polyv项目demo中的高斯模糊功能，则需要做以下的操作：

- 1、移除polyvLiveCommonModul模块中build.gradle文件的`renderscriptSupportModeEnabled true`配置
- 2、移除polyvLiveCommonModul模块中`SupportRenderScriptBlur`类
- 3、把polyvLiveCommonModul模块中`PLVBlurUtils`类中的`SupportRenderScriptBlur`修改为`RenderScriptBlur`

## 5.6 配置方法数超过 64K 的应用

如果您项目的 `minSdkVersion` 设为21或更高的值，则默认情况下会启用MultiDex，并且您不需要MultiDex支持库。

否则可以参考[配置方法数超过 64K 的应用](#)里的方式添加MultiDex。

## 6 初始化sdk

打开您项目的 `Application` 类，在 `onCreate` 方法里添加如下代码：

```
PLVLiveSDKConfig.init(  
    new PLVLiveSDKConfig.Parameter(this)//sdk初始化所需参数  
        .isOpenDebugLog(true)//是否打开调试日志  
        .isEnabledHttpDns(false)//是否使用httpdns  
);
```

## 7 登录直播/回放及进入不同场景

### 7.1 观看端（云课堂场景/直播带货场景）

进入不同场景的观看页前，需要先调用登录直播/回放的方法进行校验。登录校验的目的是检验参数是否正确，以及sdk内部会保存从登录接口获取到的一些信息。sdk提供了 `IPLVSceneLoginManager` 进行直播/回放的登录，接口定义如下：

```

/**
 * 登陆直播观看
 *
 * @param appId      直播账号appId
 * @param appSecret   直播账号appSecret
 * @param userId      直播账号userId
 * @param channelId   频道号
 * @param loginListener 监听器
 */
void loginLiveNew(String appId,
                  String appSecret,
                  String userId,
                  String channelId,
                  OnLoginListener<PLVLiveLoginResult> loginListener);

/**
 * 登陆回放观看
 *
 * @param appId      直播账号appId
 * @param appSecret   直播账号appSecret
 * @param userId      直播账号userId
 * @param channelId   频道号
 * @param videoId     视频id
 * @param loginListener 监听器
 */
void loginPlaybackNew(String appId,
                     String appSecret,
                     String userId,
                     String channelId,
                     String videoId,
                     OnLoginListener<PLVPlaybackLoginResult> loginListener);

```

登录验证的结果会通过监听器的方法回调，可以在登录成功的回调中，选择进入所需的场景页。下面示例如何登录直播及进入场景页，代码如下：

```

//登录直播
loginManager.loginLiveNew(appId, appSecret, userId, channelId, new
IPLVSceneLoginManager.OnLoginListener<PLVLiveLoginResult>() {
    @Override
    public void onLoginSuccess(PLVLiveLoginResult plvLiveLoginResult) {
        loginProgressDialog.dismiss();
        PLVLiveChannelConfigFiller.setupAccount(userId, appId, appSecret);
        PLVLiveChannelType channelType = plvLiveLoginResult.getChannelTypeNew();
        switch (curScene) {
            //进入云课堂场景
            case CLOUDCLASS:
                PLVLaunchResult launchResult =
                PLVLCCloudClassActivity.launchLive(PLVLoginWatcherActivity.this, channelId, channelType, getViewerId(),
                getViewerName(), getViewerAvatar());
                break;
            //进入直播带货场景
            case ECOMMERCE:
                PLVLaunchResult launchResult =
                PLVECLiveEcommerceActivity.launchLive(PLVLoginWatcherActivity.this, channelId, channelType, getViewerId(),
                getViewerName(), getViewerAvatar());
                break;
            default:
                break;
        }
    }
}

@Override
public void onLoginFailed(String msg, Throwable throwable) {
    //...
}
});

```

上述方法的具体用例可以在polyv demo项目中的 `PLVLoginActivity` 类找到。

## 7.2 开播端（手机开播场景）

在进入手机开播页面，需要先调用登录频道的方法进行校验。登录校验的目的是检验参数是否正确，以及 sdk 内部会保存从登录接口获取到的一些信息。sdk 提供了 `IPLVSceneLoginManager` 进行频道开播的登录，接口定义如下：

```

/**
 * 登录手机开播
 *
 * @param channelId 频道号
 * @param pwd 频道密码
 * @param loginListener 监听器
 */
void loginStreamerNew(String channelId,
                      String pwd,
                      OnLoginListener<PLVLoginVO> loginListener);

```

登录验证的结果会通过监听器的方法回调，可以在登录成功的回调中，进入手机开播场景。下面示例如何登录频道并进入手机开播场景，代码如下：

```
loginManager.loginStreamerNew(channelId, password, new
IPLVSceneLoginManager.OnStringCodeLoginListener<PLVSLoginVO>() {
    @Override
    public void onLoginSuccess(PLVSLoginVO loginVO) {
        // 其它处理
        // 进入手机开播场景
        PLVLaunchResult launchResult = PLVSAStreamerAloneActivity.launchStreamer(
            PLVLoginStreamerActivity.this,
            loginVO.getChannelId(),
            loginVO.getInteractUid(),
            loginNick,
            loginVO.getTeacherAvatar(),
            loginVO.getTeacherActor(),
            loginVO.getChannelName(),
            loginVO.getRole(),
            loginVO.getColinMicType()
        );
    }
    @Override
    public void onLoginFailed(String msg, Throwable throwable) {
        // 登录失败的处理
    }
    @Override
    public void onLoginFailed(String msg, String code, Throwable throwable) {
        // 登录失败的处理
    }
});
```

上述方法的具体用例可以在polyv demo项目中的 PLVLSLoginStreamerActivity 类找到。