

短视频处理 LanSoEditor-SDK 之 Android 平台集成步骤

一，集成方式（Android Studio 环境下）：

说明 1，为了方便您在实际编程中查看调用相关的方法，我们开放了一些 native 的接口方法的各种源代码，因此需要拷贝 lansosdk 源代码文件夹，如后期你们混淆代码，JAVA 的 JNI 机制是一定要保留 native 字符串方法名字的，故请在脚本中增加：

```
-keep public class com.lansosdk.videoeditor.** {  
    <fields>;  
    <methods>;  
}  
-keep public class com.lansosdk.videoplayer.** {  
    <fields>;  
    <methods>;  
}  
-keep class com.lansosdk.box.**{*;  
}
```

我们其他的 jar 文件，已经在 sdk 提供给您的时候，做过混淆操作，请勿再次混淆。

说明 2，当前 SDK 提供 armeabi-v7a 指令集的库文件，如果您实际项目需要 armeabi 和 arm64-v8a 的指令集，请联系我们，我们另外编译提供给您。

说明 3，此集成方式为大部分开发者常用方式，如果您清晰 Android Studio 的 module 机制，也可以整理成一个单独的 module，以方便后期更新升级。

说明 4，如果您在未授权的情况下，集成我们 SDK 到您项目中，请修改您 app 的名字为我们 demo 的名字方可运行，不然底层无法运行。授权后将不再限制。

SDK 由 4 部分组成：

1. Armeabi-v7a： native 层的库文件。

拷贝《Demo 工程》中 libs 下的 armeabi-v7a 文件夹到您工程中的 app/src/main 下的 jniLibs。

2. Lansosdk 文件夹： 开放源代码的各种方法和类。

拷贝《Demo 工程》中 src/com 文件夹下的 lansosdk 文件夹 到您工程中的 app/src/main/java/com 文件夹里。如果没有 com 文件夹，则新建，里面的源代码

与 java 的 JNI native 相关联, 包名和路径不许更改, 请注意!!
【请务必确认我们的 lansosdk 文件是在 app/src/main/java/com 下】。

3. lansongeditor_hx.jar: java 层代码。

拷贝《Demo 工程》中 libs 下的 jar 文件到您工程的 app/libs 下, 需要在 build.gradle 中声明编译此 jar。

4. LSResource 文件夹 (Android 高级版本使用)

拷贝 assets 文件夹下的整个 LSResource 文件夹到您工程的 assets 文件夹下。

5. 其他建议: (Android 高级版本使用)

我们的大部分前台播放和演示采用 DrawPadView 来做的, 建议您保留 DrawpadView 类和 app\src\main\java\com\example\advanceDemo\view 文件夹里的代码, 以方便后期的更新和升级。

6. 在您的主启动加载库的代码里, 一般是 MainActivity 或 Application.java 中, 加载和初始化我们的 sdk 底层 so 库文件:

```
LoadLanSongSdk.loadLibraries();  
LanSoEditor.initSo(getApplicationContext(), null);
```

7. 在您的 Manifest.xml 文件里增加读写权限:

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

8. 刷新您的工程后, 可以按照我们《Demo 工程》里的调用方法, 来实现您的各种功能。
9. 请不要修改我们的 videoeditor 文件夹中的内容, 以方便后期的更新, 您可以在外部集成我们的类, 然后在里面写相关的方法。
10. 遇到问题, 请首先执行 MediaInfo.checkFile(“您视频的路径”), 看打印什么信息。

11. 强烈建议您在先看明白我们的 SDK 演示调用流程后, 再集成到您项目中, 这样可事半功倍, 并且我们的 SDK 每个 Activity 的演示是尽量简单的, 一般 3 个步骤就可完成。