

1 SDK 信息

您可以在腾讯云官网更新 [小直播 SDK](#)，目前小直播 SDK 有如下几版本：

版本类型	功能
直播精简版	支持推流、直播、点播
独立播放器版	支持直播、点播
短视频功能版	支持短视频和点播
全功能专业版	支持推流、直播、点播、连麦、短视频
商用企业版	在全功能专业版基础上增加动效贴纸、美瞳瘦脸、绿幕抠图功能

以专业版为例，下载完的 SDK 解压后有以下几个部分：

名称	修改日期	类型
API文档	2017/8/11 12:32	文件夹
Demo	2017/8/11 12:32	文件夹
LiteAVSDK_Professional_2.1.1339.aar	2017/8/11 11:37	AAR 文件
LiteAVSDK_Professional_2.1.1339.zip	2017/8/11 11:37	ZIP 文件

文件名	说明
LiteAVSDK_Professional_3.0.1185.aar	aar 封装方式的 SDK，适用于 Android Studio 用户
LiteAVSDK_Professional_3.0.1185.zip	jar + so 封装方式的 SDK，适用于 Eclipse 用户，如果您觉得 SDK 全量打包进 apk 会增大安装包体积，可以将 zip 包中的 so 文件上传到服务器，通过使用时再下载 so 文件的方式减少 apk 的体积，具体使用方法见 如何减少 apk 体积 ？
Demo	基于 aar 方式的简化 Demo，包含简单的 UI 界面和 SDK 的主要功能演示，使用Android Studio可以快速导入并体验。
API文档	点击文件夹里的 index.html 文件可以查看本 SDK 的所有接口描述

2 系统要求

SDK 支持在 Android 4.0.3（API 15）及以上系统上运行，但只有（Android 4.3）API 18 以上的系统才能开启硬件编码。

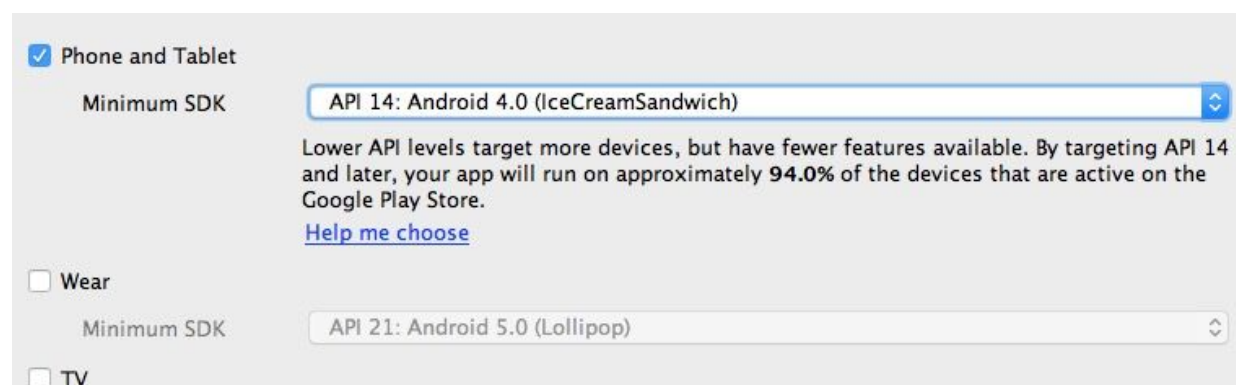
3 开发环境

以下是 SDK 的开发环境，APP 开发环境不需要与 SDK 一致，但要保证兼容：

- Android NDK: android-ndk-r12b
- Android SDK Tools: android-sdk_25.0.2
 - minSdkVersion: 15
 - targetSdkVersion: 21
- Android Studio（推荐您也使用 Android Studio，当然您也可以使用 Eclipse+ADT）

4 集成攻略（aar）

4.1 新建工程



4.2 拷贝文件

将 aar 包放在工程 libs 目录下即可

4.3 工程配置

- 在工程 app 目录下的 build.gradle 中，添加引用 aar 包的代码：

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    // 导入腾讯云直播 SDK aar
    compile(name: 'LiteAVSDK_Professional_3.0.1185', ext: 'aar')
}
```

- 在工程目录下的 build.gradle 中，添加 flatDir，指定本地仓库：

```
allprojects {
    repositories {
        jcenter()
        flatDir {
            dirs 'libs'
        }
    }
}
```

- 在工程目录下的 build.gradle 的 defaultConfig 里面，指定 ndk 兼容的架构：

```
defaultConfig {
    applicationId "com.tencent.liteav.demo"
    minSdkVersion rootProject.ext.minSdkVersion
    targetSdkVersion rootProject.ext.targetSdkVersion
    versionCode 1
    versionName "2.0"

    ndk {
        abiFilters "armeabi", "armeabi-v7a"
        // 如果您使用的是商业版，只能使用 armeabi 架构，即：
        // abiFilters "armeabi",
    }
}
```

- 最后编译一下工程 Rebuild Project。

5 集成攻略 (jar)

5.1 库说明

解压 LiteAVSDK_Professional_3.0.1185.zip 压缩包后得到 libs 目录，里面主要包含 so 文件和 jar 文件，文件清单如下：

so文件	说明
liteavsdk.jar	小直播 SDK android 核心库
libliteavsdk.so	小直播 SDK 核心组件
libsaturn.so	小直播 SDK 核心组件
libtraeimp-rtmp-armeabi.so	连麦功能所使用的声学组件库
libstlport_shared.so	C++ stl 基础库（请不要随意替换，版本不匹配亦产生崩溃）
libijkffmpeg.so	ffmpeg 基础库（ijk 版本），用于点播播放功能，解决一些奇葩视频格式的兼容问题
libijkplayer.so	ijkplayer 开源库，用于点播播放功能，解决一些奇葩视频格式的兼容问题
libijksdl.so	ijkplayer 开源库，用于点播播放功能，解决一些奇葩视频格式的兼容问题

5.2 拷贝文件

如果您的工程之前没有指定过 jni 的加载路径，推荐您将刚才解压的 jar 包和 so 库在 /src/main/jniLibs 目录下，这是 android studio 默认的 jni 加载目录。

5.3 工程配置

在工程 app 目录下的 build.gradle 中，添加引用 jar 包和 so 库的代码。

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    // 导入腾讯云直播 SDK jar  
    compile fileTree(dir: 'src/main/jniLibs', includes: ['*.jar'])  
}
```

6 配置 APP 权限

在 AndroidManifest.xml 中配置 APP 的权限，音视频类 APP 一般需要以下权限：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.Camera"/>
<uses-feature android:name="android.hardware.camera.autofocus" />
```

7 验证

在工程中调用 SDK 接口，获取 SDK 版本信息，以验证工程设备是否正确。

7.1 引用 SDK

在 MainActivity.java 中引用 SDK 的 class：

```
import com.tencent.rtmp.TXLiveBase;
```

7.2 调用接口

在 onCreate 中调用 getSDKVersion 接口获取版本号：

```
String sdkver = TXLiveBase.getSDKVersionStr();
Log.d("liteavsdk", "liteav sdk version is : " + sdkver);
```

7.3 编译运行

如果前面各步骤都操作正确，demo 工程将顺利编译通过，运行之后将在 logcat 中看到如下 log 信息：

```
08-10 19:30:36.547 19577-19577/ D/liteavsdk: liteav sdk version is : 3.0.1185
```

8 减少 APK 体积

整个 SDK 的体积主要来自于 so 文件，这些 so 文件是 SDK 正常运行所依赖的音视频编解码库、图像处理库以及声学处理组件，如果小直播 SDK 的功能不是 App 的核心功能，您可以考虑采用在线加载的方式减少最终 apk 安装包的大小。

8.1 上传 SO 文件

将 SDK 压缩包中的 so 文件上传到 CDN，并记录下载地址，比如

`http://xxx.com/so_files.zip`。

8.2 启动准备

在用户启动 SDK 相关功能前，比如开始播放视频之前，先用 loading 动画提示用户“正在加载相关的功能模块”。

8.3 下载 SO 文件

在用户等待过程中，APP 就可以到 `http://xxx.com/so_files.zip` 下载 so 文件，并存入应用目录下（比如应用根目录下的 files 文件夹），为了确保这个过程不受运营商 DNS 拦截的影响，请在文件下载完成后校验 so 文件的完整性。

8.4 加载 SO 文件

等待所有 so 文件就位以后，调用 TXLiveBase 的 `setLibraryPath` 将下载的目标 path 设置给 SDK，然后再调用 SDK 的相关功能。之后，SDK 会到这些路径下加载需要的 so 文件并启动相关功能。

9 LOG 打印

在 TXLiveBase 中可以设置 log 是否在控制台打印以及 log 的级别，具体代码如下：

- **setConsoleEnabled**
设置是否在 Android Studio 的控制台打印 SDK 的相关输出。
- **setLogLevel**
设置是否允许 SDK 打印本地 log，SDK 默认会将 log 写到 sdcard 上的 **log / tencent / liteav** 文件夹下。
如果您需要我们的技术支持，建议将开关打开，在重现问题后提供 log 文件，非常感谢您的支持。
- **Log 文件的查看**
小直播 SDK 为了减少 log 的存储体积，对本地存储的 log 文件做了加密，并且限制了 log 数量的大小，所以要查看 log 的文本内容，需要使用 log [解压缩工具](#)。

```
TXLiveBase.setConsoleEnabled(true);  
TXLiveBase.setLogLevel(TXLiveConstants.LOG_LEVEL_DEBUG);
```

常见问题排查

如果您将 SDK 导入到您的工程，编译运行出现类似以下错误：

```
Caused by: android.view.InflateException:  
Binary XML file #14:Error inflating class  
com.tencent.rtmp.ui.TXCloudVideoView
```

可以按照以下流程来排查问题：

- 确认是否已经将 SDK 中的 jar 包和 so 库放在 jniLibs 目录下。
- 如果您使用 aar 集成方式的完整版本，在工程目录下的 build.gradle 的 defaultConfig 里面确认下是否将 x64 架构的 so 库过滤掉。因为完整版本中连麦功能所使用的声学组件库暂时不支持 x64 架构的手机。

```
defaultConfig {  
    applicationId "com.tencent.liteav.demo"  
    minSdkVersion rootProject.ext.minSdkVersion  
    targetSdkVersion rootProject.ext.targetSdkVersion  
    versionCode 1  
    versionName "2.0"  
  
    ndk {  
        abiFilters "armeabi", "armeabi-v7a"  
    }  
}
```

- 检查下混淆规则，确认已将 SDK 的相关包名加入了不混淆名单。

```
-keep class com.tencent.** { *; }
```

联系我们

关注公众号“腾讯云视频”，给公众号发关键字“技术支持”，会有专人联系。