

拓幻科技
TiSDK (Android)
产品文档
【V2050230105】



合肥拓幻智能科技有限公司



目录

一、	基础描述	2
1.	最新特性	2
2.	支持平台	2
二、	集成步骤	2
1.	申请试用 KEY	2
2.	检查 SDK 资源	2
3.	导入工程	2
4.	使用 TIUI（开源，可选）	3
三、	对接说明	4
1.	初始化	4
2.	渲染步骤	4

一、基础描述

1. 最新特性

TiSDK 是针对视频帧渲染的一个库组件，含有**美颜**（美白、磨皮、粉嫩、饱和、亮度），**美型**（大眼、瘦脸、额头、下巴、瘦鼻、嘴型）、**贴纸**、**滤镜**（特效滤镜、美颜滤镜）、**礼物**、**抖动**、**哈哈镜**、**绿幕抠图**等功能，提供视频帧特效的渲染解决方案。

2. 支持平台

TiSDK 核心功能是渲染相关部分，支持对 OpenGL 纹理、byte 数组等数据的渲染，兼容腾讯云、七牛云、金山云、声网、环信、网易云、即构科技、阿里云、三体云等各大云平台的直播 SDK、短视频 SDK、连麦 SDK、视频通话 SDK 等。

二、集成步骤

1. 申请试用 Key

TiSDK 需要用 **Key** 鉴权后方可使用，**Key** 需要向商务申请，分为在线鉴权的 **Key** 和离线鉴权的 **Key** 两种，对应在线鉴权和离线鉴权两种鉴权模式。

2. 检查 SDK 资源

Android 版本的 **SDK**，包含 **ti-sdk.aar** 依赖库、**jniLibs** 文件夹、**assets** 资源文件、**tiui** 依赖工程（可选）。

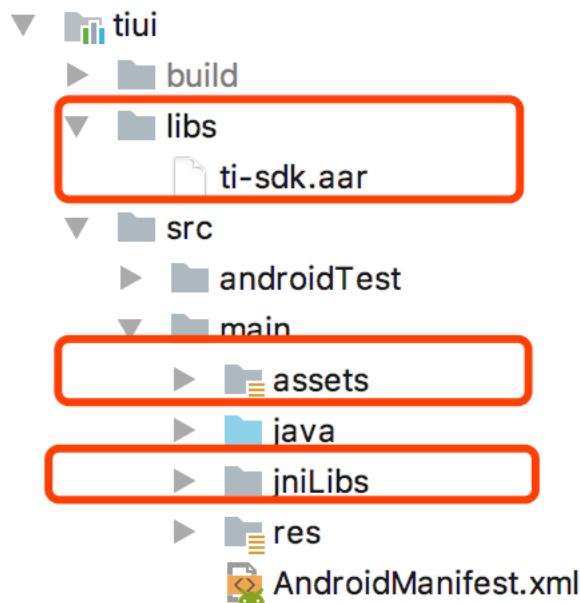
3. 导入工程

- 将 **ti-sdk.aar** 文件拷贝到 **app** 模块中的 **libs** 文件夹下，并在 **app** 模块的 **build.gradle** 文件的

dependencies 中，增加如下依赖：

```
dependencies {  
    implementation files('libs/ti-sdk.aar')  
    //老版本的gradle使用compile语句  
    //compile files('libs/ti-sdk.aar')  
}
```

- 将 **jniLibs** 文件夹中，各个 **ABI** 对应的 **libTiSDK.so** 文件，拷贝到对应目录中；
- 将 **assets** 资源文件拷贝到项目的对应目录中；



4. 使用 tiui（开源，可选）

- 依赖我们的 **tiui** 工程，使用我们提供的开源 **UI** 库，将 **tiui** 文件夹拷贝到工程根目录下，在工程根目录的 **settings.gradle** 文件中，增加如下代码：

```
include ':tiui'
```

- 在 **app** 模块中的 **build.gradle** 文件的 **dependencies** 中，增加如下代码：

```
implementation project(':tiui')  
//老版本的 gradle 使用 compile 语句  
//compile project(':tiui')
```

三、 对接说明

1. 初始化

- TiSDK 的初始化接口，需要使用 **Key** 进行鉴权，**Key** 分为在线鉴权的 **Key** 和离线鉴权的 **Key** 两种，根据需求分别向商务申请即可。TiSDK 初始化函数程序中调用一次即可生效，建议用户在 Application 创建的时候调用；如果渲染功能使用不频繁，也可以在使用的时候调用，接口如下：

```
//在线鉴权初始化方法
TiSDK.init(" key", context);
//离线鉴权初始化方法
TiSDK.initOffline(" key", context);
```

2. 渲染步骤

- 新建 **TiSDKManager** 管理器对象，所有的渲染都必须通过 **TiSDKManager** 对象调用，**TiSDKManager** 的默认构造函数，新建默认渲染参数的渲染器对象；也可以通过具体功能函数来预设 **TiSDKManager** 的渲染参数，代码如下：

```
//生成默认参数渲染器
TiSDKManager tiSDKManager = new TiSDKManager();
//构造器生成
TiSDKManager tiSDKManager = new TiSDKManagerBuilder()
    .setBeautyEnable(true)
    .setSkinWhitening(100)
    //...
    .build();
```

- 渲染 **GL_TEXTURE_EXTERNAL_OES** 纹理，调用 **TiSDKManager** 的 **renderOESTexture** 方法，返回 **GL_TEXTURE_2D** 类型的纹理 **Id**，代码如下：

```
int textureId = tiSDKManager.renderOESTexture(
    oesTexture, //oes纹理Id
```

```
        imageWidth, //图像宽度
        imageHeight, //图像高度
        rotation, //TiRotation枚举, 图像顺时针旋转的角度
        isMirror //图像是否左右镜像
    );
```

- 渲染 **GL_TEXTURE_2D** 的纹理, 调用 **TiSDKManager** 的 **renderTexture2D** 方法, 返回 **GL_TEXTURE_2D** 类型的纹理 **Id**, 代码如下:

```
int textureId = tiSDKManager.renderTexture2D(
    texture2D, //2D纹理Id
    imageWidth, //图像宽度
    imageHeight, //图像高度
    rotation, //TiRotation枚举, 图像顺时针旋转的角度
    isMirror //图像是否左右镜像
);
```

- 渲染 **byte[]** 视频帧, 调用 **TiSDKManager** 的 **renderPixels** 方法, 渲染视频帧, 代码如下:

```
tiSDKManager.renderPixels(
    data, //视频帧数组
    format, //视频帧格式, NV21 or RGBA
    imageWidth, //图像宽度
    imageHeight, //图像高度
    rotation, //TiRotation枚举, 图像顺时针旋转的角度
    isMirror //图像是否左右镜像
);
```

- 渲染结束, 要释放渲染资源, 否则会造成内存泄漏, 一般随着渲染数据回调接口的销毁释放, 或者随着当前 **Activity** 或者 **Fragment** 的生命周期结束释放。释放资源调用 **TiSDKManager** 的 **destroy** 方法, 代码如下:

```
tiSDKManager.destroy();
```

- 调用 **TiSDKManager** 里面的相关设置函数来设置渲染参数, 具体接口说明如下:

模块	功能	函数	参数说明
美颜	开关	setBeautyEnable(boolean enable)	true:美颜生效 ; false:美颜失效
美颜	美白	setSkinWhitening(int param)	美白参数, 范围[0,100], 0 为没有效果
美颜	磨皮	setSkinBlemishRemoval(int param)	磨皮参数, 范围[0,100]
美颜	粉嫩	setSkinTenderness(int param)	粉嫩参数, 范围[0,100]
美颜	饱和	setSkinSaturation(int param)	饱和参数, 范围[0,100]
美颜	亮度	setSkinBrightness(int param)	亮度参数, 范围[0,100]
美型	大眼	setEyeMagnifying(int param)	瘦脸参数, 范围[0, 100]
美型	瘦脸	setChinSlimming(int param)	下巴参数, 范围[0, 100]
美型	下巴	setJawTransforming(int param)	额头参数, 范围[0, 100]
美型	额头	setForeheadTransforming(int param)	嘴型参数, 范围[0, 100]
美型	嘴型	setMouthTransforming(int param)	瘦鼻参数, 范围[0, 100]
美型	瘦鼻	setNoseMinifying(int param)	瘦脸参数, 范围[0, 100]
贴纸	切换	setStickerName(String stickerName)	切换贴纸特效, json 文件中选择
滤镜	切换	setFilterEnum(TiFilterEnum filterEnum)	切换滤镜特效, TiFilterEnum 中选择
礼物	切换	setGiftName(String giftName)	切换礼物特效, json 文件中选择
抖动	切换	setRockEnum(TiRockEnum rockEnum)	切换抖动特效, TiRockEnum 中选择
水印	切换	setWatermarkName(String watermarkName)	切换水印特效, json 文件中选择
哈哈镜	切换	setDistortionEnum(TiDistortionEnum distortionEnum)	切换哈哈镜特效, TiDistortionEnum 中选择

■ 将贴纸、礼物和水印文件配置到自己的服务器（强烈建议）：

```

TiSDK.setStickerUrl(String url);//贴纸 zip 包根目录
TiSDK.setStickerThumbUrl(String url);//贴纸封面图根目录
TiSDK.setGiftUrl(String url);//礼物 zip 包根目录
TiSDK.setGiftThumbUrl(String url);//礼物封面图根目录
TiSDK.setWatermarkUrl(String url);//水印图片和封面根目录

```

■ 将模型文件从服务器端下载（针对包体大小有要求的）：

```

//将模型文件下载到downloadPath后, 再调用TiSDK.init(String license, Context context)进行初
//始化操作
String downloadPath = TiSDK.getModelPath();

```



■ SDK 的日志功能

```
TiSDK.setLog(boolean enable);//true:打开日志 false:关闭日志
```