

# Android端opensdk美颜集成说明文档

---

## 一、Android studio配置工程

## 二、SDK调用步骤, 功能实现

### 2.1 SDK初始化

接口描述:

初始化接口:

参数说明:

返回值:

具体代码示例如下:

### 2.2 美颜API使用

#### 2.2.1 美颜算法实例对象

算法实例化接口:

参数说明:

返回值:

具体代码示例如下:

#### 2.2.2 单个能力license鉴权

接口描述:

license证书验签接口:

获取证书路径:

参数说明:

返回值:

具体代码示例如下:

#### 2.2.3 传入美颜能力的文件

接口描述:

传入文件接口:

参数说明:

返回值:

具体代码示例如下:

#### 2.2.4 设置美颜各能力的强度

接口描述:

传入接口:

参数说明:

返回值:

具体代码示例如下:

### 2.2.5 美颜设置接口

接口描述:

传入接口:

参数说明:

返回值:

具体代码示例如下:

### 2.2.6 算法销毁

接口描述:

接口示例:

参数说明:

返回值:

具体代码示例如下:

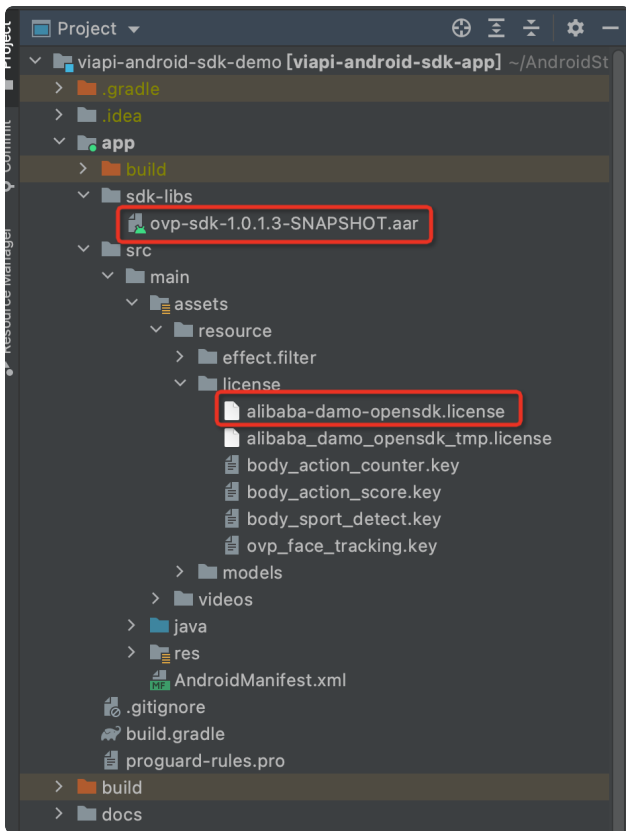
## 三、支持的系统和硬件版本

## 四、注意事项

附 离线鉴权错误码定义

# 一、Android studio配置工程

1、获取相关资源压缩包（由官网线上购买申请或阿里云相关人员提供下载链接）后，解压压缩包，可看到如下资源文件，demo示例工程、支持相关能力的aar及支持相关能力的license文件。如下图：



注意：alibaba-damo-opensdk.license为正式证书（官网下载获取的都是正式证书），\_tmp结尾的为临时证书一般线下提供，临时证书不能改名，正式license可以改名字，但是不能与tmp license重名。两个证书只需要调用其中一个进行鉴权就可以。

## 二、SDK调用步骤，功能实现

### 2.1 SDK初始化

#### 接口描述：

算法API使用前先调用SDK初始化接口，初始化之后，各功能才可以正常使用，否则会引起鉴权等异常，初始化建议放在app进程启动时Application onCreate中进行。

#### 初始化接口：

```
Plain Text | 复制代码  
1 VIAPICreateApi.getInstance().getVIAPISdkCore().init(Context context,boolean isDebug);
```

### 参数说明:

Context context 应用上下文。

boolean isDebug SDK调试开关。

### 返回值 :

int类型, 返回0为初始化成功, 其它返回为初始化失败。

### 具体代码示例如下:

```
Plain Text | 复制代码
1 private void initSDK() {
2     int status = VIAPICreateApi.getInstance().getVIAPISdkCore().init(this, false);
3     if (status != 0) {
4         Toast.makeText(this, VIAPIStatusCode.getErrorMsg(status), Toast.LENGTH_LONG).show();
5     } else {
6         Toast.makeText(this, "初始化成功!", Toast.LENGTH_LONG).show();
7     }
8 }
```

## 2.2 美颜API使用

### 2.2.1 美颜算法实例对象

OVFaceBeauty: 在需要用到美颜算法的地方, 创建OVFaceBeauty美颜实例, OVFaceBeauty是美颜API接口对象, 通过此对象可以完成camera实时视频流美颜能力的使用。

### 算法实例化接口:

```
Plain Text | 复制代码
1 //美颜
2 OVFaceBeauty mFacebeauty = new OVFaceBeauty();
```

### 参数说明:

无

返回值:

无

具体代码示例如下:

```
Plain Text | 复制代码  
1 private final OVFaceBeauty mFacebeauty;  
2 mFacebeauty = new OVFaceBeauty();
```

## 2.2.2 单个能力license鉴权

接口描述:

license证书验签接口, 验签通过后才能成功调用算法。

license证书验签接口:

```
Plain Text | 复制代码  
1 mFacebeauty.nativeCheckLicense(String licensePath);
```

获取证书路径:

```
Plain Text | 复制代码  
1 String licensePath = AssetsProvider.getDebugLicenseFilePath();
```

参数说明:

String licensePath 传入全局证书license路径或自定义的单个能力license路径。

注: 如果所有能力使用同一个全局证书默认传入全局证书路径即可, 如果接入方有针对此能力的单独证书, 则需要传入单独的证书文件的绝对路径

返回值:

int类型，返回0为验签成功，其它返回为验签失败。

具体代码示例如下：

```
Plain Text | 复制代码  
1 int status = mFacebeauty.nativeCheckLicense(licensePath);
```

### 2.2.3 传入美颜能力的文件

接口描述：

传入美颜所需的资源文件。

传入文件接口：

```
Plain Text | 复制代码  
1 mFacebeauty.nativeFaceBeautyCreate(String lutResourcePath);
```

参数说明：

String lutResourcePath 传入美颜算法所需的资源文件，传资源文件上层文件夹的路径。

返回值：

int类型，返回0为成功，其它返回为失败。

具体代码示例如下：

```
Plain Text | 复制代码  
1 int status = mFacebeauty.nativeFaceBeautyCreate(lutResource);
```

### 2.2.4 设置美颜各能力的强度

接口描述：

设置美颜各能力效果的强度

### 传入接口：

```
▼ Plain Text | 复制代码  
1 mFacebeauty.nativeFaceBeautySetParam(int mode, float values);
```

### 参数说明：

String mode 美颜的类型，目前三个类型：OSFaceBeautyMode.OS\_BEAUTIFY\_SKIN\_WHITEN.mode（美白）、

OSFaceBeautyMode.OS\_BEAUTIFY\_SKIN\_BLUR.mode（磨皮）、

OSFaceBeautyMode.OS\_BEAUTIFY\_SKIN\_SHARPEN.mode（锐化）；

float values 美颜的强度值，取值范围为0.00–1.00；

### 返回值：

int类型，返回0为成功，其它返回为失败。

### 具体代码示例如下：

```
▼ Plain Text | 复制代码  
1 int status = mFacebeauty.nativeFaceBeautySetParam(OSFaceBeautyMode.OS_BEAUTIFY_SKIN_WHITEN.mode, MathUtil.getFormatTwoDecimalPlaces  
2 ((float) progress / 100));
```

## 2.2.5 美颜设置接口

### 接口描述：

通过此接口使美颜能力生效

### 传入接口：

```
▼ Plain Text | 复制代码  
1 mFacebeauty.nativeFaceBeautyProcessTexture(int id, int Width, int Height, int rotate);
```

### 参数说明：

int id：需要处理并绑定数据的纹理id（此id指opengl es纹理绑定id）；

int Width：传入帧数据的宽；

int Height：传入帧数据的高；

int rotate：设备旋转的角度（可参考Demo查看获取方式）；

### 返回值：

int类型，返回美颜处理之后绑定的纹理id。

### 具体代码示例如下：

```
Plain Text | 复制代码  
1 id = mFacebeauty.nativeFaceBeautyProcessTexture(id, size.getWidth(), size.g  
    etHeight(), rotate);
```

## 2.2.6 算法销毁

### 接口描述：

在不需要用到算法的地方进行算法销毁。

### 接口示例：

```
Plain Text | 复制代码  
1 mFacebeauty.nativeFaceBeautyDestroy();
```

### 参数说明：

无

### 返回值：

int类型，返回0为删除算法相关成功，其它返回为删除算法相关处理失败。

### 具体代码示例如下：



```
1 int status = mFacebeauty.nativeFaceBeautyDestroy();
```

### 三、支持的系统和硬件版本

- 1、硬件要求：要求设备上有相机模块,陀螺仪模块
- 2、CPU架构：armeabi-v7a、arm64-v8a
- 3、系统：最低支持 Android 4.0（API Level 14）需要开发者通过minSdkVersion来保证支持系统的检测

### 四、注意事项

- 1、viapi-android-sdk的 minSdkVersion为 14。
- 2、demo工程Android Studio 3.4 及以上，Open GLES 2.0 及以上。
- 3、证书分为临时证书以\_tmp结尾和正式证书。临时证书一般作为调试用，正式证书一般作为发布用。根据证书的不同代码需要做相应调整，全局搜索mIsTmpLicense字段，如果是\_tmp证书，需把该字段改为true，正式证书的话把该字段改为false。

### 附 离线鉴权错误码定义

- 2011 license没有初始化直接调用API接口。
- 2012 当前的license与调用app不是绑定关系，license用在其他app中使用。
- 2013 license无效。
- 2014 license授权时间过期。
- 2015 此license中不包含调用的算法能力（未购买此能力）。
- 2016 bundle id获取失败。
- 2017 临时license时间校验失败。