

## Android SDK 集成

### 1、客户端 SDK 接入需要的准备工作

- 1) 进入 cps 联盟-推广管理-app 管理, 创建应用;
- 2) 待审核成功后, 操作列中即可下载唤起 sdk;
- 3) 下载客户端 SDK。

下载 SDK 需要 certMD5 (即 Android 签名 CERT.RSA 文件中的 md5 值) 和 certSHA1 (即 Android 签名 CERT.RSA 文件中的 SHA1 值), 这些信息将生成安全图片。

#### 安全图片:

「安全图片」是京东联盟 SDK 安全机制的一部分, 安全图片位于下载的 SDK 包中, 为了使 SDK 正常工作, 需要开发者确保正确的使用安全图片。需要注意以下几点:

- 安全图片的文件名为「safe.jpg」, \*\*请勿修改\*\*。
- 安全图片位于 res/drawable(1.0.0 版本在 raw)目录中, 如果通过模块依赖的方式集成 SDK, 不需要移动该文件。
- 安全图片与开发者编译构建 APK 所用的 keystore 对应, 如果修改了 keystore (比如从 debug 切换为正式的 keystore), 则需要替换成对应的安全图片。

### 2、客户端 SDK 加入工程

针对不同的 IDE 有不同的添加方式。以 AndroidStudio 为例进行说明。

首先, 将下载的联盟 sdk 解压。

然后, 通过 Import Module 的方式引入联盟 sdk 模块, 如下图。

最后, 添加联盟 sdk 依赖, 也就是主工程依赖刚刚添加的模块即可。**(注意: 不可将联盟 sdk 中资源、lib 库、文件拷贝到工程的方式添加, 这样可能产生编译或者跟单等问题)。**

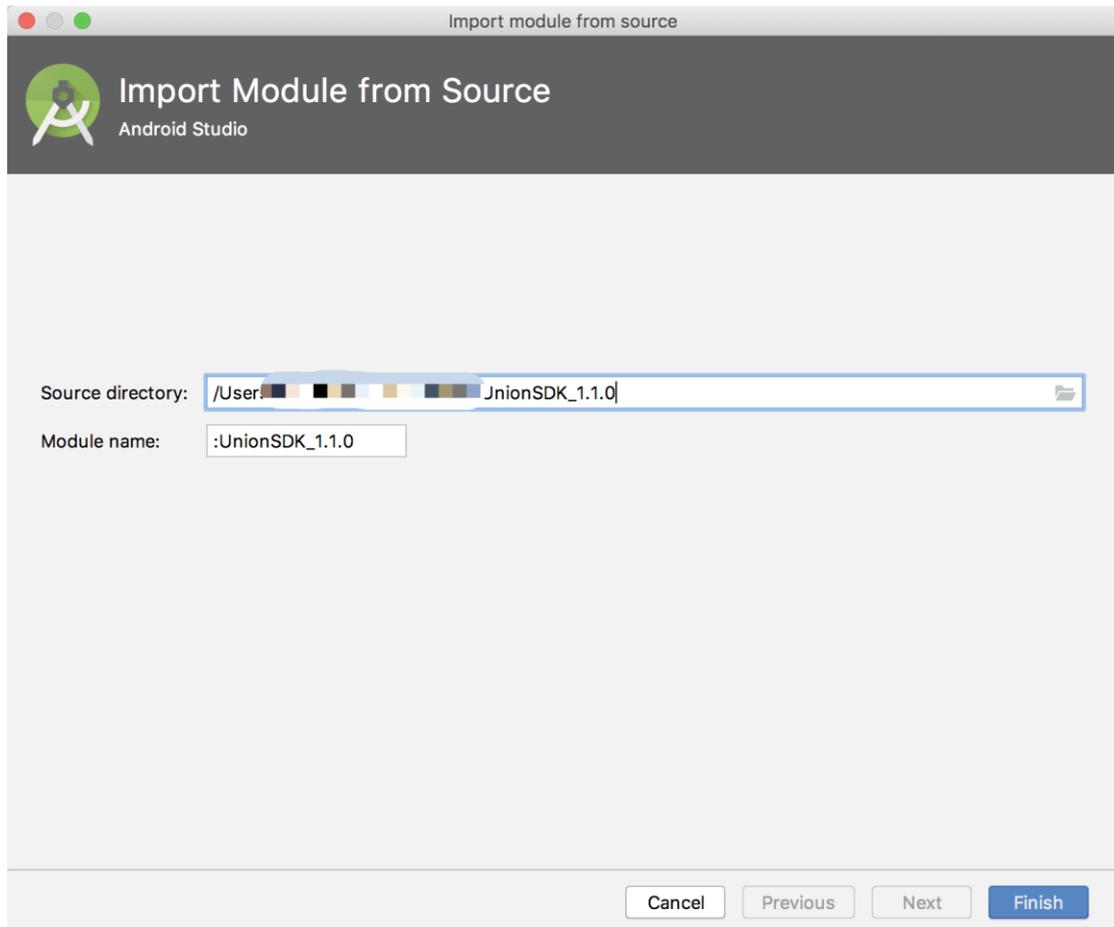


图 1 联盟 sdk 引入工程

### 3、代码

#### 配置权限

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

#### 配置混淆

```
-keep class com.kepler.jd.**{ public <fields>; public <methods>; public *; }
```

## 1) SDK 初始化

使用 KeplerApiManager.asyncInitSdk 方法来初始化 SDK, 建议将 KeplerApiManager.asyncInitSdk 方法的调用放在 Application 的 onCreate 中,

在应用启动时就初始化。(SDK 中的所有方法都要在主线程中调用)

上传 APK 下载 SDK, 这里需要校验, 您的证书(shh), 还有包名 (package name), 签名, 请保证这些信息和上传 APK 一直否则校验不能通过

```
KeplerApiManager.asyncInitSdk(Application.this, appKey, keySecret,
new AsyncInitListener() {
    @Override
    public void onSuccess() {
        // TODO Auto-generated method stub
        Log.e("Kepler", "Kepler asyncInitSdk onSuccess ");
    }
    @Override
    public void onFailure() {
        // TODO Auto-generated method stub
        Log.e("Kepler",
"Kepler asyncInitSdk 授权失败, 请检查 lib 工程资源引用; 包名,签名证书是否和注册一致");
    }
});
```

## 2) 打开京东主站任意 url

```
private KeplerAttachParameter mKeplerAttachParameter = new KeplerAttachParameter();
```

```
OpenAppAction mOpenAppAction = new OpenAppAction() {
    @Override
    public void onStatus(final int status, final String url) {
        mHandler.post(new Runnable() {
            @Override
```

```

        public void run() {
            if (status == OpenAppAction.OpenAppAction_start) { //开始状态未必一定执
行,
                dialogShow();
            }else {
                mKelperTask = null;
                dialogDiss();
            }
            if(status == OpenAppAction.OpenAppAction_result_NoJDAPP) {
                //未安装京东
            }else if(status == OpenAppAction.OpenAppAction_result_BlackUrl){
                //不在白名单
            }else if(status == OpenAppAction.OpenAppAction_result_ErrorScheme){
                //协议错误
            }else if(status == OpenAppAction.OpenAppAction_result_APP){
                //呼京东成功
            }else if(status == OpenAppAction.OpenAppAction_result_NetError){
                //网络异常
            }
        }
    });
}
};

```

// 通过 url 呼京东主站

// url 通过 url 呼京东主站的地址

// mKeplerAttachParameter 存储第三方传入参数

// mOpenAppAction 呼京东主站回调

KeplerApiManager.getWebViewService().openAppWebViewPage(mContext,

url,

mKeplerAttachParameter,

mOpenAppAction);

错误码	描述	解决方法
0	呼京东成功	—
1	正在启动	—
2	协议错误	—
3	未安装京东	安装京东 app
4	不在白名单	使用白名单的 url
5	安全参数 id 为空	检查联盟 sdk 中 libs 下依赖库文件的完整性（所有 jar 包，以及是否包含 4 种 CPU 类型 arm64-v8a, armeabi, x86, armeabi-v7a 的 so 文件，所有文件均不可删除
-1100	网络异常	设置网络连接

### 3) 参数

KeplerAttachParameter 作为入口参数，承载以下字段功能：

setCustomerInfo(String kepler\_customer\_Info)

// 该参数为第三方应用自定义参数，可以为页面，频道标示，实现统计，打点等需求，在订单维度会存储此字段。

// 传参长度，使用 URL encode 之后长度必须小于 256 字节，否则会抛出异常

// 严禁传入中文及以下特殊字符 (= # % & + ? < {})

setPositionId(int positionId)

// 该参数为计费参数

putKeplerAttachParameter(String key, String value)

// 扩展功能，暂时无扩展能力

详细使用方法请查询 demo.

注：如需要通过 SDK 唤起京东 APP 首页，请使用如下 url 转链后，再调用 SDK 唤起：

[https://m.jd.com/?isopen=1&ad\\_od=1&allowJDApp=1](https://m.jd.com/?isopen=1&ad_od=1&allowJDApp=1)