

小鹅通店铺 SDK Android 接入说明

一、前置条件

- 1、注册使用小鹅店铺，取得店铺 appID
- 2、购买小鹅店铺 sdk 接入服务，取得 sdkAppId

二、引入 SDK

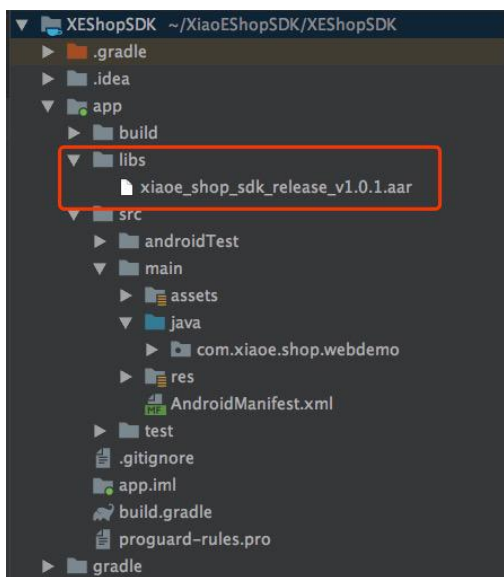
Android SDK 暂时只提供 aar 的方式进行下载，eclipse 开发等待后续升级

1、SDK 文件说明：

SDK 内包含 aar 文件、sdk 使用 demo、sdk 接入文档、sdk 更新说明

三、接入指引

1、将“xiaoe_shop_sdk_release_vx.x.x.aar”文件拷贝进待接入 sdk 的 moudle 下的“libs”目录下，并加入 gradle 依赖。若“libs”文件夹不存在，需新建该文件夹



2、在项目或 moudle 的入口位置（建议在自定义的 Application 的 onCreate()）调用 init 方法：

```
XiaoEWeb.init(this,"appId","clientId",WebViewType.X5);
```

参数说明:

this: Context 上下文参数

appId: 店铺 id

clientId: 小鹅店铺 sdk appId

type: 使用的 webview 类型 (sdk 支持原生和腾讯 x5 两种内核, 从性能方面考虑, 建议使用腾讯 X5 内核)

3、权限声明:AndroidManifest.xml 里加入权限声明、使用硬件加速以及屏幕旋转配置

权限声明:

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

开启硬件加速和配置屏幕旋转:

使用小鹅店铺 sdk, 加载小鹅店铺的 webview 所在 activity 必须开启硬件加速以及屏幕旋转配置, 如下:

```
<activity android:name=".WebActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"
    android:hardwareAccelerated="true"/>
```

4、最小化调用

```
xiaoEWeb = XiaoEWeb.loadDefault(this,webLayout,url);
```

参数说明:

this: Activity

webLayout: webview 的承载容器布局 (只能为 layout)

url: 加载的页面链接 (一般为店铺首页 url)

5、注册店铺页面回调

sdk 针对小鹅店铺提供如下的页面回调事件:

```
public class JsInteractType{
    public static final int SHARE_ACTION = 1;//分享
    public static final int LOGIN_ACTION = 2;//登录
    public static final int LOGOUT_ACTION = 3;//取消登录
}
```

注册方式：通过注册 JsBridgeListener 实现

```
public interface JsBridgeListener{
    void onJsInteract(intactionType, JsCallbackResponse response);
}
```

在监听器内通过 `actionType` 区分不同的回调事件，具体实现方式参考 demo 实现。小鹅的店铺，必须需要 **token** 鉴权才能进入店铺商品详情页等业务操作或个人信息页面。(获取 **token** 的方式见 **API 文档**) 获取到 **token** 后需调用

```
public void sync(XEToken xeToken){}
XEToken 说明：（key 与 value 必须业务后台从小鹅 sdk 后台获取到的 key、value 一致）
public class XEToken
{
    private String tokenKey;
    private String tokenValue;
}
```

将 **token** 设置进 **sdk**，在适当的时机需清空 **token** 信息（如：退出登录）

```
public void syncNot()
```

6、生命周期管理

为了避免内存泄漏，需要在 `activity` 或 `fragment` 的对应生命周期函数内调用 `webview` 对应的生命周期函数：

```
xiaoEWeb.webLifeCycle().onResume();
xiaoEWeb.webLifeCycle().onPause();
xiaoEWebwebLifeCycle().onDestroy();
```

7、页面预加载

为了提供对某些页面的加载速度，**sdk** 提供了预加载实现方案（暂时只支持一级页面的预加载）

```
XiaoEWeb.preloadingUrl(context,url);
```

四、高阶用法

1、自定义显示：自定义显示需使用调用链的方式进行调用：

```
XiaoEWeb.with(this)
    .setWebParent()
    .useCustomUI()
    .useCustomTopIndicator()
    .setWebSetting()
    .setUrlLoader()
    .setWebUIController()
    .setWebViewClient()
    .setWebChromeClient()
    .buildWeb()
    .loadUrl()
```

黑色部分：必须原样调用

绿色部分：设置 **webview** 的父容器，可传父容器、**LayoutParams**、**webview** 在父容器内是第几个子 **view**，三个参数中 **webview** 的父容器为必填字段，其他两个字段为可选字段，自由组合

红色部分：设置加载中、加载错误的 **view**、以及加载错误时点击重新加载的 **view id**
所有参数都可以自由组合，若不需要自定义可直接使用 **useDefaultUI()** 方法

蓝色部分：可设置加载中进度样式、**webview** 的 **WebSetting**、链接加载器、**webview** 的 **UIController**、**WebViewClient**、**WebChromeClient**，所有 **set** 方法可以调用一个或多个，如不需要自定义，可直接调用 **useDefaultTopIndicator()** 使用默认的加载进度指示器或调用 **colseTopIndicator()** 关闭加载指示器

2、功能性方法

clearWebViewCache(): 清除 **webview** 加载缓存

loginCancel(): 退出登录

share(): 获取分享内容（暂只支持分享店铺首页）