

**久旭移动联盟SDK技术配置说明**

**版本信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改描述 | 作者 |
| 2016/07/08 | V1.0 | 创建 | Max.qiu |
| 2016/07/23 | V1.01 | 修改 | Wanghl |

# 概述

本文档描述安卓apps 接入SDK的技术过程。本文档面向的读者是安卓应用开发者。

SDK开发包结构说明：ADSDK开发包中已经包含以下部份：

* demo：接入AD SDK的示例代码。
* doc：AD SDK接入指南。
* SDK：包括所有的jar包、so库、AndroidManifest.xml的设置等等

# 接入流程

## 快速开始

1. 向久旭申请APPid和APPkey，由久旭提供广告的Unintid
2. 将SDK目录下的所有文件，加入工程中的对应目录，如：[2.3环境搭建](#_社交SDK接入指南)
3. 配置AndroidManifest.xml，如：[2.4配置AndroidManifest.xml](#_配置AndroidManifest.xml)
4. 初始化AD SDK。
5. 调用广告接口，广告接口如：[2.6调](#_支付类-购买道具商品)用接口；

## 创建应用以及广告位

登录开发者管理后台 创建APP并填写详细信息：如下图



APP创建完成后，即可获取应用的App Key 和 App ID（如下图）：



添加完应用后，等待管理员的审核，审核完成后，可以创建广告位。广告位管理-SDK管理页面，创建广告位并选择广告位类型，确认后会生成APPid、APPkey和Unintid，然后就可以配置SDK。（如下图）

注：在投放移动广告的时候，需要您联系我们客服，不然广告将显示空白状态。

联系方式QQ：2851765565

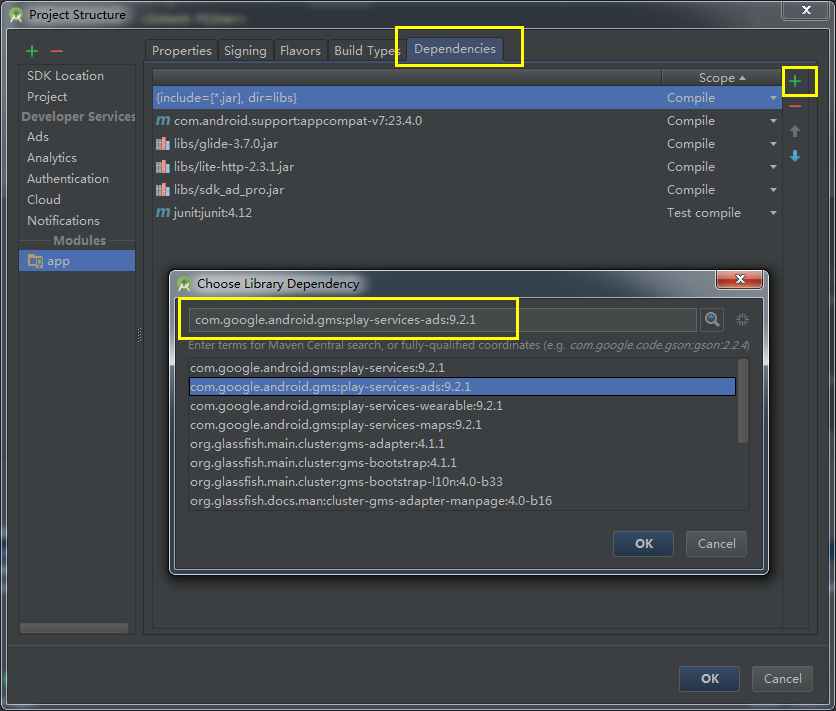
官网:app.9xu.com



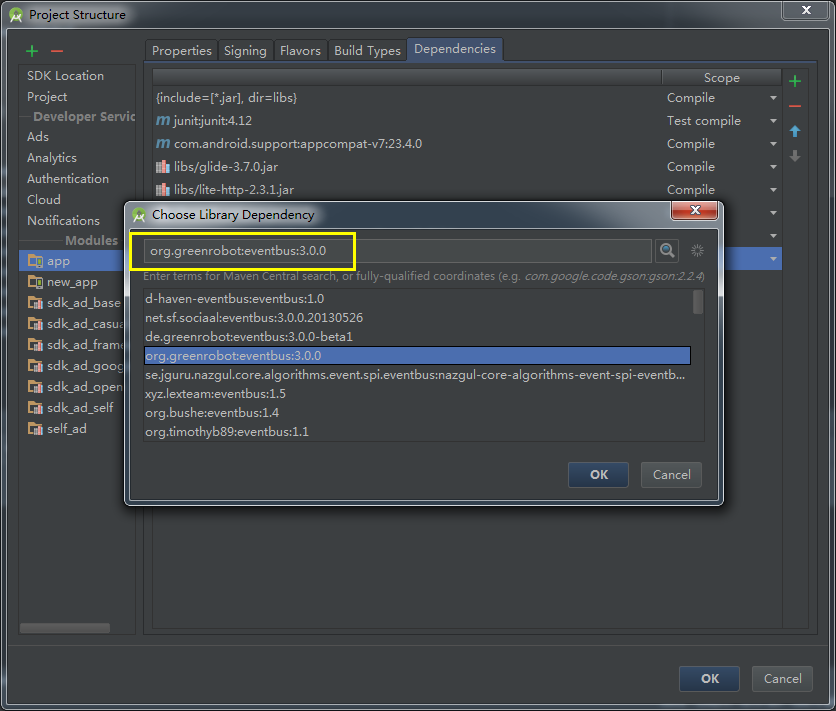
## 环境搭建（Android Studio）

* 1. 将SDK目录下的所有jar、so等复制到你的工程目录下的“libs”；

2) 需要添加google的play-service依赖，具体版本以搜索到的为主，如下图：



3) 同理，添加EventBus 3.0依赖



4) 在项目的build.gradle文件的android标签内添加以下内容，用于编译包含libs目录下的so文件

s*ourceSets{  
 main{  
 jniLibs.srcDirs = ['libs']  
 }  
 }*

## 配置AndroidManifest.xml

配置游戏工程的AndroidManifest.xml文件，包括：

* 1. 获取SDK目录下的AndroidManifest.xml.permision.txt，将其中所有权限加入xml文件
  2. 获取SDK目录下的AndroidManifest.xml.activity.txt，将其中所有内容加入application节点中

## 初始化SDK

完成androidmanifest.xml配置后，初始化SDK。

## 调用接口

### 2.6.1接口

sdk的接口调用与参数等信息可以参考sdk包中的demo程序，使用android studio导入demo工程。

demo提供了可以获取测试广告的久旭appid，appkey以及两种广告类型的unitid，开发时也可以在自己的工程中使用和验证：

**AppID：73**

**AppKey：c0325c1ac5211684269a770187c5cb68**

**横幅广告unitID：27**

**插页广告unitID：29**

需要在自定义的Application 里的onCreate调用

|  |
| --- |
| AdModOpen.*register*(context); |

然后在需要的地方进行初始化，等初始化完成后就可以进行广告的调用了。

|  |
| --- |
| AdModOpen.initialize(this, APPid, APPkey, new SdkInitListener() {  @Override  public void initFinish() {  }  @Override  public void initError() {  }  } |

### 2.6.2 广告类-横幅广告

#### 接口

|  |
| --- |
| AdModOpen.loadAd(MainActivity.this,Unintid, viewGroup,new AdListener() {  @Override  public void onAdLoaded() {  }  @Override  public void onAdFailedToLoad(int errorCode, String errorMsg) {  }  @Override  public void onAdOpened() {  }  @Override  public void onAdClosed() {  }  @Override  public void onAdLeftApplication() { } }); |

#### 描述

调用横幅广告时。只需要调用上面的方法，即可。

#### 参数说明

|  |  |  |
| --- | --- | --- |
| 参数类型 | 参数名称 | 描述 |
| Activity | this | 当前Activity界面 |
| String | Unintid | 广告的Unintid |
| LinearLayout | viewGroup | 横幅广告的显示位置，由接入sdk方自由设置位置（一般在页面的布局xml 里定义） |
| AdListener | adListener | 广告的回调接口  onAdLoaded： 当接收广告成功时调用。  onAdFailedToLoad ： 当广告请求失败时调用。  onAdOpened:当广告打开涵盖整个屏幕的重叠式广告时调用。  onAdClosed :当用户在点击广告后返回应用时调用。  onAdLeftApplication:当广告退出应用（例如，转到浏览器）时调用。 |

#### 返回值

N/A

### 2.6.3广告类-插页广告

#### 接口

对于插页式广告，建议您提前加载，并稍后在应用流程中出现自然暂停时显示。例如，我们运行一段简单的游戏，然后将其停止，并稍后利用一个按钮再次启动以展示给用户。

**以下代码段展示如何在用户开始新游戏之前预加载并显示插页式广告：**

|  |
| --- |
| AdModOpen.loadFullAd(MainActivity.this, Unintid,new AdListener() {  @Override  public void onAdLoaded() {  }  @Override  public void onAdFailedToLoad(int errorCode, String errorMsg) {  }  @Override  public void onAdOpened() {  }  @Override  public void onAdClosed() {  }  @Override  public void onAdLeftApplication() {  } }); |

**显示已经预加载好的广告**

if (!AdModOpen.showFullAd(MainActivity.this)){  
 Toast.makeText(MainActivity.this,"广告正在加载，请稍后",Toast.LENGTH\_SHORT).show();  
}

以上新代码将检查是否已加载插页广告。如果已加载，则回自动显示该广告。 如果广告尚未准备好显示，则应用自由定义需要做的事情

#### 描述

调用插页广告时。只需要调用上面的方法，即可。需要在Activity 内注册Activity！

#### 参数说明

|  |  |  |
| --- | --- | --- |
| 参数类型 | 参数名称 | 描述 |
| Activity | this | 当前Activity界面 |
| String | Unintid | 广告的Unintid |
| AdListener | adListener | 广告的回调接口  onAdLoaded：当接收广告成功时调用。  onAdFailedToLoad：当广告请求失败时调用。  onAdOpened:当广告打开涵盖整个屏幕的重叠式广告时调用。  onAdClosed :当用户在点击广告后返回应用时调用。  onAdLeftApplication:当广告退出应用（例如，转到浏览器）时调用。 |

#### 返回值

N/A