

# 点聚应用互推平台 SDK 开发者接入手册 V1.0

## 目录

1	概述.....	3
2	开发前准备.....	3
2.1	申请开发者账号.....	3
2.2	创建应用.....	3
3	配置 AndroidManifest.xml.....	6
3.1	配置 appId 和 appKey.....	6
3.2	权限声明.....	7
3.3	广播配置.....	7
3.4	全屏广告及推广墙配置.....	7
3.5	广告条配置.....	8
3.6	开发环境搭建.....	8
4	点聚应用互推平台 API .....	9
4.1	初始化 SDK 及关闭 SDK.....	9
4.2	广告条 API .....	10
4.3	广告墙 API .....	11
4.4	全屏广告 API .....	11
5	平台错误代码.....	12
6	注意事项.....	12
6.1	代码混淆.....	12
6.2	关于安卓 V4 开发包 .....	13

## 版本历史

版本/状态	作者	参与者	起止日期	备注
V1.00/审核	赖震宇		2013-01-18	

版本说明：版本从 1.00 开始，每次功能增加版本号增加 0.01，大版本号增加 1；  
每次版本增加的内容使用黄底黑字标出，便于浏览。

# 1 概述

本文档主要介绍应用如何接入点聚应用互推平台和SDK中相关API调用。点聚应用互推平台以广告条、全屏广告、推广墙的方式展示被推广APP的应用信息。

点聚应用互推平台支持Android SDK1.6及以上版本。下载完点聚应用互推平台SDK的zip包，解压后文件结构如下：

demo	点聚应用互推平台 SDK 的 DEMO 工程
doc	点聚应用互推平台 SDK 相关的文档
libs	dianju.jar (版本号以官网发布的具体版本为准)
readme.txt	点聚应用互推平台 SDK 包的描述文件

## 2 开发前准备

### 2.1 申请开发者账号

如果要在应用中植入点聚 SDK，开发者需要事先在点聚主页(<http://tt.91.com/>)申请开发者账号。



图 1

### 2.2 创建应用

进入开发者应用管理界面，点击“新增应用”添加应用



图 2

填写应用信息(以 SDK\_Demo 为例)

开发者中心

1、基本信息

系统平台：

\* 应用名称：

点聚Demo

\* 应用分类：

工具

\* 应用图标：

上传图标

应用版权信息：

V1.0

应用描述：

点聚Demo

新增

返回

图 3

应用创建完成后，可以在应用列表中查看所添加的应用的 **appId** 和 **appKey**

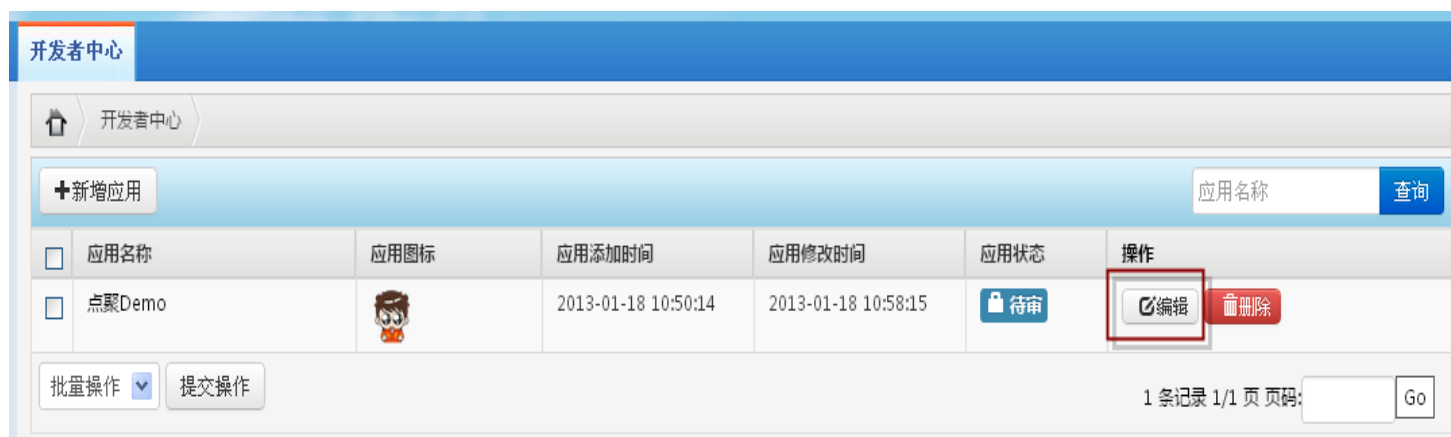


图 4

点击“编辑”后，依次按步骤完成应用添加操作



图 5

点击“下一步”，在步骤 2 中点击“查看应用密码”（嵌入 SDK 时将使用到）



图 6

注：上传 apk 包后，需要等待平台管理员审核后，才能进行广告发布操作！

## 3 配置 AndroidManifest.xml

### 3.1 配置 appId 和 appKey

在 AndroidManifest.xml 中 application 节点下配置先前得到的应用密码：

```
<!-- 配置 appId 和 appKey(必须) -->
```

```
<meta-data
```

```
    android:name="dianju-app-id"
```

```
    android:value="258" />
```

```
<meta-data
```

```
    android:name="dianju-app-key"
```

```
    android:value="745f3f6135efbfe2de1a732a3c44783d" />
```

这里需要修改成自己应用的 appId 和 appKey

## 3.2 权限声明

AndroidManifest.xml 中 application 节点内需要加入如下权限：

```
<!-- 配置所需要的权限(必须) -->

<uses-permission android:name="android.permission.READ_PHONE_STATE" />

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<uses-permission android:name="android.permission.INTERNET" />
```

## 3.3 广播配置

```
<!-- 配置 Receiver(必须) -->

<receiver android:name="com.bodong.dianju.sdk.receiver.AppStateReceiver" >

    <intent-filter>

        <action android:name="com.bodong.dianju.sdk.action.STATE_CHANGE" />

        <action android:name="android.intent.action.PACKAGE_ADDED" />

        <action android:name="android.intent.action.PACKAGE_REPLACED" />

        <action android:name="android.intent.action.PACKAGE_REMOVED" />

        <data android:scheme="package" />

    </intent-filter>

</receiver>
```

## 3.4 全屏广告及推广墙配置

```
<!-- 配置全屏 Activity(不需要全屏广告，则可不用配置) -->

<activity

    android:name="com.bodong.dianju.sdk.DianJuFullscreen"

    android:configChanges="orientation|keyboard|keyboardHidden"

    android:theme="@android:style/Theme.Dialog" />
```

```
<!-- 配置广告墙 Activity(不需要广告墙，则可不用配置) -->

<activity

    android:name="com.bodong.dianju.sdk.DianJuActivity"

    android:configChanges="orientation|keyboard|keyboardHidden" />
```

## 3.5 广告条配置

```
<!-- 这里若设置成 android:visibility="gone"，则广告条不会显示 -->

<com.bodong.dianju.sdk.DianJuBanner

    android:id="@+id/banner_id"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentBottom="true"

    android:layout_centerInParent="true"

    android:visibility="visible" >

</com.bodong.dianju.sdk.DianJuBanner>
```

如果要使用广告条，将以上代码复制到需要显示广告条的页面布局文件中即可

## 3.6 开发环境搭建

基于点聚应用互推平台SDK进行应用开发时需要加入并引用dianju(版本号).jar。以Eclipse为例演示一下搭建过程。

(SDK需要使用 android-support-v4.jar 开发包，如果Eclipse没有自动在libs目录下加入该开发包，请在SDK的zip包中找到该文件并复制到工程libs目录下！)

首先需要在接入应用中添加dianju(版本号).jar。具体步骤如下：

- 1、右键单击接入应用工程，在弹出菜单中选择New -> Folder新建libs文件夹。



2、将位于SDK zip包中libs目录下的dianju(版本号).jar拷贝到该文件夹。

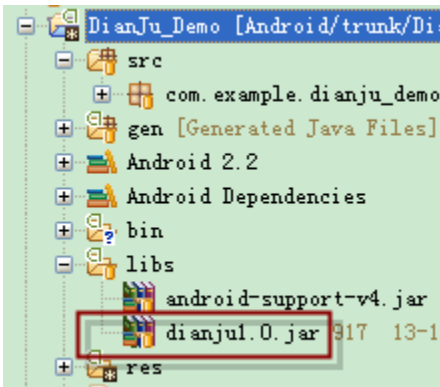


图 7

右键单击工程，选择Build Path中的Configure Build Path...，选中Libraries这个tab，并通过Add Jars...导入工程libs目录下的dianju(版本号).jar文件

## 4 点聚应用互推平台 API

### 4.1 初始化 SDK 及关闭 SDK

初始化 SDK
<pre>public static void init(Context context)</pre>
初始化点聚 SDK， <b>每个应用只需要初始化一次</b>  调用所有平台提供的 API 前，请确保对 SDK 进行了初始化！
使用示例：  <pre>@Override  public void onCreate(Bundle savedInstanceState) {     super.onCreate(savedInstanceState);      //这里需要先调用平台的init方法！推荐写在onCreate的第一句。      DianJuPlatform.init(this);      setContentView(R.layout.activity_main);  }</pre>

关闭 SDK
<pre>public static void destroy(Context context)</pre>
<p>使用示例：</p> <pre>@Override     protected void onDestroy() {         //需要在退出程序时调用平台的destroy方法关闭SDK         DianJuPlatform.destroy(this);         super.onDestroy();     }</pre>

## 4.2 广告条 API

首先将广告条植入相关页面(参考 5.5)

显示广告条
<pre>public void display ()</pre>
<p>使用示例：</p> <pre>DianJuBanner banner=(DianJuBanner)findViewById(R.id.banner_id); banner.display();</pre> <p>注：广告条默认是显示状态的，不需要特别去调用该 API</p>

关闭广告条
<pre>public void close ()</pre>
<p>使用示例：</p> <pre>DianJuBanner banner=(DianJuBanner)findViewById(R.id.banner_id); banner.close();</pre>

### 4.3 广告墙 API

打开广告墙
<pre>public static void openOfferWall(Context context)</pre>

### 4.4 全屏广告 API

准备全屏广告
<pre>public static void prepareFullscreenAd(Context context, PrepareFullAdListener listener)</pre>
<p>当全屏广告准备完毕后，会回调 PrepareFullAdListener 的 onSuccess 方法</p> <p>如果广告数据准备失败，会回调 PrepareFullAdListener 的 onError 方法</p> <p>为了提升用户体验，建议在使用全屏广告时，先调用该 API 准备好广告数据</p>

打开全屏广告
<pre>public static void openFullscreenAd(Context context) throws PreparingException, UnPreparedException</pre>
<p>当全屏广告准备完毕后，调用该函数可以显示全屏广告。</p> <p>如果未调用 prepareFullscreenAd 函数而直接调用该函数，函数将抛出 UnPreparedException 错误</p> <p>如果调用了 prepareFullscreenAd 函数但广告数据未准备完毕时就调用该函数，函数将抛出 PreparingException 错误！</p>

准备并打开全屏广告
<pre>public static void prepareAndOpenFullscreenAd(Context context, OpenFullAdListener listener) throws PreparingException</pre>
<p>准备并打开全屏广告，调用该函数后，广告数据准备完毕后将直接显示全屏广告，无需再调用 openFullscreenAd 函数！</p>

关闭全屏广告窗口
<pre>public static void closeFullscreenAd(Context context)</pre>
注：全屏广告窗口上已经提供了关闭按钮给用户关闭，该函数仅提供开发者更方便控制全屏广告使用。

设置全屏广告窗口能否响应返回键
<pre>public static void setFullscreenCloseableOnBackPressed(boolean isCloseOnBackPressed)</pre>
设置当用户按下返回键时，是否关闭全屏广告窗口(默认是不响应返回键，用户只能通过点击全屏广告窗口右上方的关闭按钮来关闭该窗口)

## 5 平台错误代码

101	无网络连接
102	服务器应答错误码
103	数据解析错误
104	网络通信异常
105	无广告数据
106	未知异常

## 6 注意事项

### 6.1 代码混淆

如果需要使用 `proguard` 进行代码混淆的话，需要在 `proguard` 配置中加上以下参数：

```
-dontwarn android.support.**  
-dontwarn com.bodong.dianju.sdk.**  
-keep class android.support.**{*;}  
-keep class com.bodong.dianju.sdk.**{*;}
```

## 6.2 关于安卓 V4 开发包

如果捕获到以下错误，请检查工程的 `libs` 目录下，是否有 `android-support-v4.jar` 支持包

AndroidRuntime	FATAL EXCEPTION: main
AndroidRuntime	java.lang.VerifyError: com.bodong.dianju.sdk.other.cj
AndroidRuntime	at com.bodong.dianju.sdk.DianJuActivity.a(Unknown Source)
AndroidRuntime	at com.bodong.dianju.sdk.DianJuActivity.a(Unknown Source)
AndroidRuntime	at com.bodong.dianju.sdk.DianJuActivity.onCreate(Unknown Source)
AndroidRuntime	at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1047)
AndroidRuntime	at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:1615)
AndroidRuntime	at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:1667)
AndroidRuntime	at android.app.ActivityThread.access\$1500(ActivityThread.java:117)
AndroidRuntime	at android.app.ActivityThread\$H.handleMessage(ActivityThread.java:935)
AndroidRuntime	at android.os.Handler.dispatchMessage(Handler.java:99)
AndroidRuntime	at android.os.Looper.loop(Looper.java:130)
AndroidRuntime	at android.app.ActivityThread.main(ActivityThread.java:3687)
AndroidRuntime	at java.lang.reflect.Method.invokeNative(Native Method)
AndroidRuntime	at java.lang.reflect.Method.invoke(Method.java:507)
AndroidRuntime	at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:867)
AndroidRuntime	at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:625)
AndroidRuntime	at dalvik.system.NativeStart.main(Native Method)