



AdView Android SDK 开 发 者 手 册

目录

I.	注册并获取 SDK	3
II.	添加 SDK	4
III.	AndroidManifest.xml 文件配置.....	5
3.1	增加权限代码.....	5
3.2	增加 Activity 声明.....	5
IV.	创建横幅广告(AdBannerActivity).....	6
4.1	增加代码来增添广告(CodeLayout)	6
4.2	纯代码方式(btnCode2).....	8
4.3	XML 方式布局广告(btnXml)	8
4.4	AdViewInterface 侦听接口	9
V.	创建插屏广告.....	10
5.1	创建插屏.....	10
5.2	插屏接口调用	10
5.3	创建自定义插屏	11
VI.	创建开屏广告.....	12
6.1	创建开屏广告	12
6.2	插屏接口调用	12
6.3	自定义开屏顶部倒计时通知样式	13
VII.	AdView 后台应用设置.....	14
7.1	发布应用设置.....	14
7.2	选择广告平台	14
7.3	添加广告平台	15
7.4	配置广告平台	15
7.5	地理位置优化.....	16
VIII.	添加自定义广告平台.....	17
8.1	参考代码.....	17
8.1.1	自定义函数实现.....	17
8.1.2	回调函数处理(部分).....	18
IX.	指定应用渠道.....	19
X.	常见问题处理.....	20
10.1	如果应用想混淆 (ProGuard) 怎么办?	20
10.2	如果有问题, 和谁联系?	25
10.3	如何使用 sample	25
10.3.1	加载 sample 工程, 方式一	25
10.3.2	加载 sample 工程, 方式二	27

I. 注册并获取 SDK

1. 访问 AdView 网站 <http://www.adview.cn> 注册 Adview 帐号。
2. 登录后在“我的产品”页面选择“发布应用”。
3. 应用平台选择“Android”,根据提示完成相关信息,您将获取唯一的 SDK key。
4. 点击“应用管理”页面,配置自己的应用。

yyz_3 SDK-KEY: SDK20151516030130pj9h5ywnfyqacuu

☐ 横幅广告
 ☐ 全屏/插屏广告
 ☐ 开屏广告
 跳转到

地域优化 ☒ ON
 广告补余 ☒ ON

国内用户配置 + 添加广告平台

广告平台	评分	设置	开关	投放量	优先级	删除平台
竞价广告	7.69	设置	<input checked="" type="checkbox"/>	<input type="text" value="100"/> %		
广点通	6.91	未设置	<input type="checkbox"/>	<input type="text" value="--"/> %		删除
直投广告	4.55	设置	<input type="checkbox"/>	<input type="text" value="--"/> %		
交换广告	2.00	设置	<input type="checkbox"/>	<input type="text" value="--"/> %		

投放总量: **100 %**

国外用户配置 + 添加广告平台

广告平台	评分	设置	开关	投放量	优先级	删除平台
竞价广告	7.69	设置	<input type="checkbox"/>	<input type="text" value="50"/> %		

投放总量: **0 %**

[保存](#)

5. 首页-> Android SDK 下载, 或者应用管理-> Android SDK 下载, 能获取 AdView SDK 包, 包括文档和 sample;

Android 横幅广告SDK开源地址(一体包中包含横幅广告SDK)

AdBannerSDK_Android SDK开源代码: [点击此处去下载>>>](#)

Android SDK(横幅&全/插屏广告) 版本号: 3.1.3 更新时间: 2015-01-23

[查看更新日志](#) [查看在线文档](#)

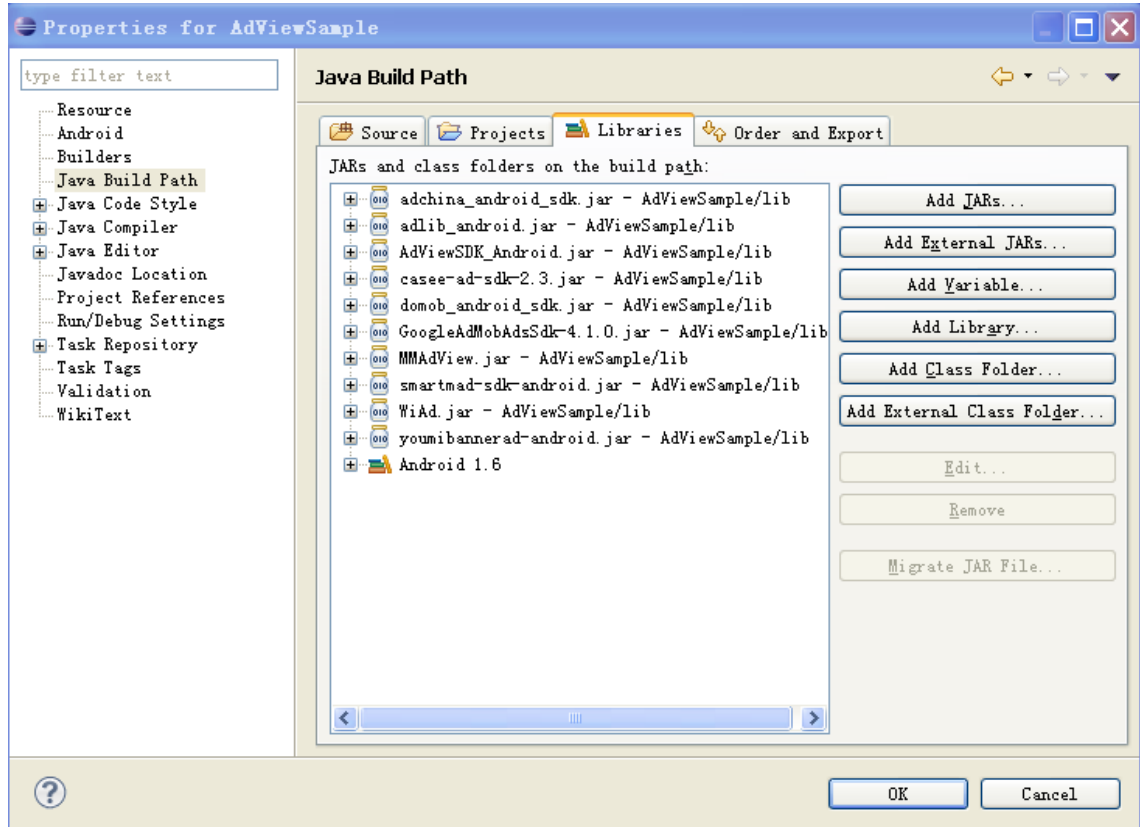
[下载](#)

[下载Android SDK历史版本](#)

[下载iOS SDK历史版本](#)

II. 添加 SDK

6. 将从网站获取 AdView SDK 开发包解压，libs 文件夹里包含所有广告平台的 SDK。(README.pdf 中有每个 jar 对应的广告平台的说明)
7. 将 AdViewSDK_Android.jar 放入到您的应用工程中。
8. 将应用程序可能用到的所有广告商的 SDK 用同样的方式添加进去。(使用 1.9.2 版以后的版本时，只能使用 **adview** 提供的 jar，使用其他途径取得的会导致广告不能展示)



注意：我们提供的SDK可以让您自由选择您喜欢的广告公司，但我们还是建议您添加我们提供的所有广告公司SDK,因为您可以在后台更自由的选择和配置。

9. 如果您还没有注册这些广告平台，请先到这些广告平台注册，添加App，记下对应的App Key，以便在AdView中绑定和设置。
10. 如果你是一个新手，对广告平台不怎么了解，也不知道选择哪家广告平台，或者不知道哪家广告平台收入稳定，没关系，**Adview**建议你使用竞价看看效果。

特别提醒：集成友盟(Umeng)SDK的时候需要把umeng_res目录下的资源文件放到工程的res

集成微云(wiyun)SDK的时候需要把wiyun_res目录下的资源文件放到工程的res,并且添加相应权限。

使用新版 Admob(google_play_service)需要更新ADT至最新版本。

III. AndroidManifest.xml 文件配置

3.1 增加权限代码

需要添加的必备权限（完整代码可参考Sample中AndroidManifest）

```
<!-- AdView SDK mandatory or important permissions -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

备注：

INTERNET：允许访问网络，这是必须的

ACCESS_NETWORK_STATE：允许访问网络的状态，这是必须的

READ_PHONE_STATE：允许访问手机的各种状态，这是必须的

ACCESS_COARSE_LOCATION：允许一个程序访问CellID或WiFi热点来获取粗略的位置

ACCESS_FINE_LOCATION：允许一个程序访问精确位置(如GPS)

ACCESS_WIFI_STATE：允许程序访问Wi-Fi网络状态信息

WRITE_EXTERNAL_STORAGE：允许程序访问外部存储设备，可以缓存广告

READ_EXTERNAL_STORAGE：允许程序访问外部存储设备

3.2 增加 Activity 声明

1. 一些广告平台需要声明 Activity，才能正常工作，在 application 标签中包含；没有在下面列出来的，表示那些广告平台不需要声明 Activity；

其中竞价广告须添加配置：

```
<service android:name="com.kyview.DownloadService" />
<activity android:name="com.kyview.AdviewWebView" />
```

2. 使用竞价/补余时需要添加 **android-support-v4.jar**

3. 还有一些广告配置比较特殊，需要特别说明，比如 Baidu 和 Greystripe, Umeng 的广告，添加的时候一定要注意看说明；Millennial 平台需要将整个 google-play-services-lib 的项目导入；具体注册内容请参照 Sample 中 AndroidManifest 文件。

注：如果不聚合触控、有米、指点等广告请务必将该广播接收器删除！

IV. 创建横幅广告(AdBannerActivity)

4.1 增加代码来增添广告(CodeLayout)

```
layout = (LinearLayout) findViewById(R.id.adLayout);  
if (layout == null)  
    return;  
adStream = new AdViewStream(this, "SDK20111812070129bb9oj4n571faaka");  
adStream.setAdViewInterface(this);  
layout.addView(adStream);  
layout.invalidate();
```

注：(AdViewTargeting 为全局作用，只需要调用一次即可)

1. AdViewTargeting.setRunMode(RunMode.TEST)功能为您提供调试状态，当为 RunMode.TEST 时您的所有的广告都和广告公司之间处与测试状态，当完成后，一定要去掉这句话，因为系统默认为非测试状态。

2. AdViewTargeting.setUpdateMode(UpdateMode.EVERYTIME)功能为每次都从服务器获取最新的设置，方便您的调试，当调试结束后，一定要去掉这句话，因为系统已经为您优化成最佳时间，同时不影响应用本身的性能。

3. 通过adStream.setAdViewInterface用户可以监听广告显示(onDisplayAd)、用户点击事件(onClickAd)。

4. AdViewTargeting.setBannerSwitcherMode (BannerSwitcher.CANCLOSED)功能为开启可关闭广告功能，方便您的程序停止广告展示，默认为不可关闭，通过adStream.setAdViewInterface用户可以监听广告将要被关闭的事件 (onCloseAd)。

5. AdViewTargeting.setHtml5Switcher(this,Html5Switcher.SUPPORT)，设置后将只接收html5类型的广告，默认值为Html5Switcher.NONSUPPORT，即html5和非html5广告都接收。

6. 增加部分平台设置尺寸接口：

平台	常量值	实例
InMobi	INMOBI_AD_UNIT_728x90 INMOBI_AD_UNIT_468x60 INMOBI_AD_UNIT_320x50	AdViewTargeting.setAdInMobiSize(AdInMobiSize.INMOBI_AD_UNIT_320x50);
AdMob	BANNER, LARGE_BANNER MEDIUM_RECTANGLE FULL_BANNER LEADERBOARD SMART_BANNER	AdViewTargeting.setAdMobSize(AdMobSize.BANNER);
O2omobi (百灵欧拓)	AD_SMART_BANNER AD_SIZE_728x90 AD_SIZE_468x60 AD_SIZE_320x50	AdViewTargeting.setAdO2omobiSize(AdO2omobiSize.AD_SMART_BANNER);
Youmi (有米)	SIZE_320x50 SIZE_468x60 SIZE_728x90 FIT_SCREEN	AdViewTargeting.setAdYoumiSize(AdYoumiSize.FIT_SCREEN);
SamrtAd (亿动智道)	AUTO_AD_MEASURE TABLET_AD_MEASURE_728X90 TABLET_AD_MEASURE_468X60	AdViewTargeting.setAdSmartAdSize(AdSmartAdSize.AUTO_AD_MEASURE);
Gdt (广点通)	BANNER BANNER_SMALL SMART_BANNER	AdViewTargeting.setAdGdtSize(AdGdtSize.BANNER);
Adview 竞价、补余	BANNER_AUTO_FILL BANNER_SMART BANNER_320X50 BANNER_480X75 BANNER_728X90	AdViewTargeting.setAdSize(AdSize.BANNER_SMART);

4.2 纯代码方式(btnCode2)

```

if (adStream != null)
    adStream.setClosed(true);
if (null != adStream) {
    ViewGroup rootView = (ViewGroup)findViewById(android.R.id.content);
    for (int i = 0; i < rootView.getChildCount(); i++) {
        if (rootView.getChildAt(i) == adStream) {
            rootView.removeView(adStream);
        }
    }
}
if (layout != null)
    layout.removeAllViews();
adStream = new AdViewStream(this,
    "SDK20111812070129bb9oj4n571faaka");
adStream.setAdViewInterface(this);
FrameLayout.LayoutParams params = new FrameLayout.LayoutParams(
    FrameLayout.LayoutParams.WRAP_CONTENT,
    FrameLayout.LayoutParams.WRAP_CONTENT);
params.gravity = Gravity.BOTTOM;
this.addContentView(adStream, params);

```

4.3 XML 方式布局广告(btnXml)

开发者可以根据广告在界面中的显示位置，在需要添加广告的 Activity 对应的 XML 布局文件中增加 AdViewStream 布局，示例代码如下：

```

if (adStream != null)
    adStream.setClosed(true);
adStream = (AdViewStream) findViewById(R.id.adview_ayout);
adStream.setAdViewInterface(this);

<com.kyview.AdViewStream
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/adview_ayout" />

```

在 AndroidManifest.xml 中增加如下 meta-data 配置 AdView SDK Key:

```

<meta-data
    android:name="ADVIEW_SDK_KEY"
    android:value=" SDK20111812070129bb9oj4n571faaka"/>

```


4.4 AdViewInterface 侦听接口

```
public interface AdViewInterface {  
    /**  
     * 当广告被点击时调用该函数  
     *  
     */  
    public void onClickAd();  
  
    /**  
     * 当广告被显示时调用该函数。  
     *  
     */  
    public void onDisplayAd();  
  
    /**  
     * 当广告被关闭时调用该函数。  
     *  
     */  
    public void onClosedAd();  
}
```

注：当接口被调用时广告会停止请求，需要处理是否继续请求广告。
具体注意事项请参阅Sample。

V. 创建插屏广告

5.1 创建插屏

```
adInstlManager = new AdInstlManager(this, "SDK20111812070129bb9oj4n571faaka");
adInstlManager.setAdInstlInterface(this);

//只请求广告, 适用于预加载
adInstlManager.requestAd();
//需要请求广告成功后调用该方法, 与requestAd()配合使用
adInstlManager.showInstal();
//请求与展示广告, 单独使用
adInstlManager.requestAndshow();
```

1. 插屏关闭按钮: 在onCreate中添加代码
AdViewTargeting.setInstlSwitcherMode (InstlSwitcher.CANCLOSED);可参照demo中AdInstlActivity中的代码。
2. 设置插屏展示模式: dialog/ popupwindow。在onCreate中添加代码:
AdViewTargeting.setInstlDisplayMode(AdViewTargeting.InstlDisplayMode.DIALOG_MODE); 默认dialog模式, 或者写成AdViewTargeting.InstlDisplayMode.POPUPWINDOW_MODE, 表示popupwindow模式, 可设置窗体外可点击等。

5.2 插屏接口调用

```
public interface AdInstlInterface {
    /**
     * 点击事件通知
     */
    public void onClickAd();

    /**
     * 展示事件通知
     */
    public void onDisplayAd();

    /**
     * 消失事件通知
     */
    public void onAdDismiss();

    /**
     * 请求成功通知
     */
    public void onReceivedAd(int type, View view);
```

```
/**
 * 请求失败通知
 */
public void onReceivedAdFailed(String errorCode);
}
```

5.3 创建自定义插屏

```
adInstlManager = new AdInstlManager(this, "SDK20111812070129bb9oj4n571faaka");
adInstlManager.setAdViewInterface(this);

//设置为用户托管模式，设置之后必须要手动调用展示
AdViewTargeting.setInstlControlMode(InstlControlMode.USERCONTROL);

//只请求广告，适用于预加载
adInstlManager.requestAd();

//广告需要被展示时需要调用，返回不为null（View）则表示有广告返回，反之则没有得到广告
//可将返回的view放置在自定义的容器中展示，如dialog
adInstlMgr.getContentView();

//当成功展示时需要调用展示汇报方法（必须）
adInstlMgr.impressionAdReport();

//当广告被点击时需要调用点击事件处理方法，否则无响应（必须）
adInstlMgr.clickAdReport();

//当展示框，如dialog 被关闭的时候，请务必调用释放方法，一次请求只能对应一次展示
adInstlMgr.destroy();
```

VI. 创建开屏广告

6.1 创建开屏广告

```
//初始化开屏广告 ,需要传入布局用来加载开屏页面
AdSpreadManager adSpreadManager = new AdSpreadManager(this,
    "SDK201310111003303e4rx5msd7cn1pa", (RelativeLayout)
    findViewById(R.id.spreadLayout));
//设置开屏回调接口
adSpreadManager.setAdSpreadInterface(this);
//设置开屏下方LOGO, 必须调用该方法
adSpreadManager.setLogo(R.drawable.spread_Logo);
//设置开屏背景颜色, 可不设置
adSpreadManager.setBackgroundColor(Color.WHITE);
//请求开屏广告,广告结果需要在AdSpreadInterface接口中处理
adSpreadManager.requestAd();
```

6.2 插屏接口调用

```
public interface AdSpreadInterface {
    /**
     * 当广告被显示时调用该函数.
     */
    public void onAdDisplayAd();

    /**
     * 当广告请求成功时调用该函数.
     */
    public void onAdReceived(View v);

    /**
     * 当广告请求失败时调用该函数.
     */
    public void onAdReceiveFailed(String errorCode);

    /**
     * 当广告被关闭时调用该函数.
     */
    public void onAdClosedAd();

    /**
     * 展示时间结束可以关闭时调用.
     */
}
```

```

public void onAdSpreadPrepareClosed();

/**
 * 用户手动取消关闭 后回调.
 */
public void onAdClosedByUser();

/**
 * 自定义回调
 */
public void onAdNotifyCustomCallback(ViewGroup view,int ruleTime,int
delayTime);
}

```

6.3 自定义开屏顶部倒计时通知样式

```

// 设置后顶部会显示跳过按钮
adSpreadManager.setSpreadNotifyType(this, AdSpreadManager.NOTIFY_TEXT);

// 默认值, 不显示任何通知
public final static int NOTIFY_NULL = 0;
// 设置后顶部显示倒计时
public final static int NOTIFY_NUM = 1;
// 设置后顶部显示为跳过按钮, 在规定展示时间之后才会出现
public final static int NOTIFY_TEXT = 2;
// 设置后将调用 onAdNotifyCustomCallback(ViewGroup view,int ruleTime,int
delayTime) 接口, 可在其中自定义通知样式
public final static int NOTIFY_CUSTOM = 3;

```

开屏广告请保证广告曝光时间充足, 否则将影响收入。
可参照 demo 中 SpreadScreenActivity 中的代码。

VII. AdView 后台应用设置

7.1 发布应用设置

添加应用信息

应用信息

应用名称：

test

通过

应用平台：

Android

应用类型：

冒险模拟

冒险

通过

下载地址：

广告文字设置

背景颜色：

000000

文字颜色：

ffffff

刷新时间：

15秒

地域优化：

ON

下载提示框：

ON

推广及受众描述

保存

注：3.1.7 版本开始支持开发者在后台控制广告下载提示框的开启和关闭

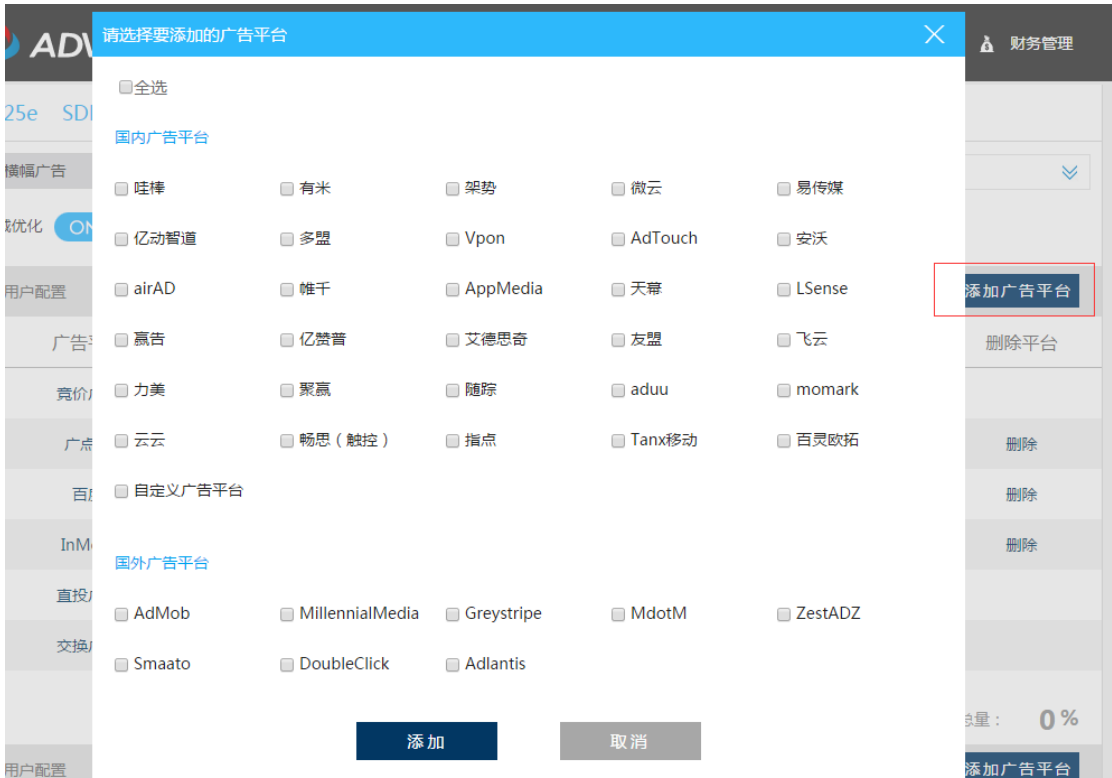
7.2 选择广告平台

yyz1 SDK-KEY:SDK20141430021046ue831q28ozje350

横幅广告	全屏/插屏广告	开屏广告	跳转到			
地域优化	ON	广告补余	OFF			
国内用户配置		+ 添加广告平台				
广告平台	评分	设置	开关	投放量	优先级	删除平台
竞价广告	7.69	设置	ON	100%		
百灵欧拓	5.33	设置	OFF	100%	+	删除
畅思 (触控)	2.00	设置	OFF	100%	+	删除
广点通	6.91	设置	OFF	100%	+	删除
帷千	7.16	设置	OFF	100%	+	删除
InMobi	5.69	设置	OFF	100%	+	删除

Copyright © 2009–2015 AdView. All Rights Reserved.

7.3 添加广告平台



开发者根据需要选择想要的平台。

7.4 配置广告平台



- 1、APP ID设置：开发者应当先去相关的广告平台注册，获取对应的APP信息。根据各广告平台接口不同，一般只需要输入APP ID，有的还需要输入其他信息。
- 2、开关：只有开关打开的那家平台的广告才会在应用中显示。开发者可以控制广告平台状态切换不同的广告平台。
- 3、投放量：只有开关打开的投放量才有意义，投放量为对广告平台的请求比例。切换广告平台应同时调整投放量，所有广告状态为on的应用投放量

总额应为100，否则无法正确保存。

特别提醒：横幅广告、全屏/插屏广告、开屏广告，每页下面均有一个“保存”键，每修改完一种广告形式，都要点击该页的“保存”键，否则所做的修改无效。

7.5 地理位置优化

横幅广告

全屏/插屏广告

开屏广告

跳转到

地域优化 ☒

广告补余 ☒

国内用户配置

+ 添加广告平台

广告平台	评分	设置	开关	投放量	优先级	删除平台
竞价广告	7.69	设置	<input checked="" type="checkbox"/>	<input type="text" value=""/>		
广点通	6.91	未设置	<input type="checkbox"/>	<input type="text" value="--"/>	<input type="text" value="↑↓"/>	删除
百度	7.42	未设置	<input type="checkbox"/>	<input type="text" value="--"/>	<input type="text" value="↑↓"/>	删除
直投广告	4.55	设置	<input type="checkbox"/>	<input type="text" value="--"/>	<input type="text" value="↑↓"/>	
交换广告	2.00	设置	<input type="checkbox"/>	<input type="text" value="--"/>	<input type="text" value="↑↓"/>	

投放总量：0%

国外用户配置

+ 添加广告平台

广告平台	评分	设置	开关	投放量	优先级	删除平台
AdMob	6.74	未设置	<input type="checkbox"/>	<input type="text" value="--"/>	<input type="text" value="↑↓"/>	删除

投放总量：0%

1、开启地域优化的功能是指定在国内的手机显示国内配置的广告，在国外的手机显示国外配置的广告；最大限度的满足不同的需求。

当地域优化关闭的时候，就不区分国内国外了，设置为开关打开的都可以显示。

VIII. 添加自定义广告平台

开发者有时想添加一个不是聚合里面的广告平台; Adview 提供了这种需求的实现;

添加广告平台中有个“自定义广告平台”;



开发者需要填写App ID1, 这是个函数名字, 需要客户端实现这个函数; 这个函数的功能是调用广告平台的接口; App ID2, 随便填点东西, 否则最终不能保存。

8.1 参考代码

Sample中提供的例子是amazon广告的实现;

8.1.1 自定义函数实现

```
// 可以访问https://developer.amazon.com/sdk/mobileads.html
public void amazon_proc() {
    // TODO Auto-generated method stub
    Log.d("AdViewSample", "Into azmazon");
    // 测试模式
    AdRegistration.enableLogging(this, true);
    AdRegistration.enableTesting(this, true);
    AdRegistration.setAppKey(this, "sample-app-v1_pub-2");

    // 创建amazon的adview实例
    adView = new AdLayout(this, AdSize.AD_SIZE_320x50);
    // 指定侦听接口
    adView.setListener(this);

    // 清除当前的view
    adStream.removeAllViews();

    RelativeLayout.LayoutParams layoutParams;
```

```

        layoutParams = new RelativeLayout.LayoutParams(
            RelativeLayout.LayoutParams.WRAP_CONTENT,
            RelativeLayout.LayoutParams.WRAP_CONTENT);

        // 添加amazon的view到当前的adlayout中
        adStream.addView(adView, layoutParams);

        AdTargetingOptions adOptions = new AdTargetingOptions();
        adView.loadAd(adOptions);
    }

```

8.1.2 回调函数处理(部分)

一般广告平台都会提供广告请求成功，请求失败的接口，通知调用者；

```

@Override
public void onAdFailedToLoad(AdLayout view, AdError error) {
    Log.w("AdViewSample",
        "Ad failed to load. Code: " + error.getResponseCode()
        + ", Message: " + error.getResponseMessage());

    // 失败之后开始请求下一个广告
    adStream.rotateThreadedPri(0);
}

@Override
public void onAdLoaded(AdLayout view, AdProperties adProperties) {
    Log.d("AdViewSample", adProperties.getAdType().toString()
        + " Ad loaded successfully.");

    // 取消侦听接口
    view.setListener(null);

    // 广告请求成功之后，启动定时器，到时后请求下一个广告
    adStream.reportImpression();
    adStream.adViewManager.resetRollover();
    adStream.rotateThreadedDelayed();
}

```

IX. 指定应用渠道

开发者在Manifest文件中加

```
<meta-data
    android:name="AdView_CHANNEL" android:value="GFAN"></meta-data>
```

可以配置渠道；（上传机锋时必须标明否则审核不过）；之前的AdViewTargeting.setChannel接口已经作废；

目前adview支持的渠道如下：

EOE (优亿市场)
GOOGLEMARKET (谷歌电子市场)
APPCHINA (应用汇)
HIAPK (安卓市场)
GFAN (机锋)
GOAPK (安智)
NDUOA (N多网)
91Store (手机91)
LIQUCN (历趣)
WAPTW (天网)
ANDROIDCN (安卓中国)
GGDWON (G友网)
ANDROIDAI (安卓之家)
STARANDROID (安卓星空)
ANDROIDD (安致)
YINGYONGSO (应用搜)
IMOBILE (手机之家)
SOUAPP (搜应用)
MUMAYI (木蚂蚁)
MOBIOMNI (欧米)
PAOJIAO (泡椒网)
AIBALA (爱扒拉市场)
COOLAPK (酷安网)
ANFONE (安丰)
APKOK (乐致网)
360MARKET (360市场)

如果不配置，或配置其他的值，一律作为“OTHER”处理；
在移动广告观察(<http://t.adview.cn/>)中有各个市场的链接；

X. 常见问题处理

10.1 如果应用想混淆（ProGuard）怎么办？

AdView是动态调用，不必对其进行混淆，广告公司代码基本已经独立混淆过，如果应用自己代码需要混淆，可以在文件proguard.cfg开头如下增加，详情可参考sample中的代码（下面的code可以去sample中copy）：

```
#The below is used for AdView SDK settings,only instead for your app
-dontwarn

# for google play service
-libraryjars '/libs/android-support-v4.jar'

-keep public class com.kyview.** {*; }
-keep public class com.kuaiyou.** {*; }
-keepclassmembers class * {public *; }
-keep public class * {public *; }
-keep class com.five.adwoad.** {*; }
-keep public class com.wooboo.** {*; }
-keep public class cn.aduu.android.** {*; }
-keep public class com.wqmobile.** {*; }
-keep class com.baidu.mobads.** {
    public protected *;
}
-keep public class com.google.android.gms.ads.** {
    public *;
}
-keep public class com.google.ads.** {
    public *;
}
-keep public class com.millennialmedia.android.* {
    <init>(...);
    public void *(...);
    public com.millennialmedia.android.MMJSResponse *(...);
}

-keep class com.qq.e.** {
    public protected *;
}
-keep class com.tencent.gdt.** {
    public protected *;
}
```

```

-dontoptimize
-dontwarn
-keep class com.mobisage.android.** {*; }
-keep interface com.mobisage.android.** {*; }
-keep class com.msagecore.** {*; }
-keep interface com.msagecore.** {*; }

#使用 触控 广告时使用
-keepattributes Exceptions
-keepattribute Signature
-keepattribute Deprecated
-keepattributes *Annotation*
#-dontwarn com.chance.**
#-dontwarn com.android.volley.NetworkDispatcher
-flattenpackagehierarchy com.chance.v4

-keep class * extends com.chance.ads.Ad {
public *;
}
-keep class com.chance.ads.AdActivity {
public *;
}
-keep class com.chance.recommend.RecommendActivity {
public *;
}
-keep class com.chance.ads.AdRequest {
public *;
}
-keep class com.chance.ads.MoreGameButton {
public *;
}
-keep class com.chance.ads.OfferWallButton {
public *;
}
-keep class com.chance.ads.RecommendButton {
public *;
}
-keep class com.chance.ads.OfferWallData {
public *;
}
-keep class com.chance.ads.OfferWallAdInfo {
public *;
}

```

```
-keep class com.chance.ads.OfferWallAdDetail {
public *;
}
-keep class com.chance.response.TaskInfo {
public *;
}
-keep class com.chance.exception.PBException {
public *;
}
-keep class com.chance.listener.AdListener {
public *;
}
-keep class com.chance.listener.PointsChangeListener {
public *;
}
-keep class com.chance.listener.QueryPointsListener {
public *;
}
-keep class com.chance.listener.GetAdDetailListener {
public *;
}
-keep class com.chance.listener.GetAdListListener {
public *;
}
-keep class * extends android.app.Service {
public *;
}
-keep class com.chance.report.ReportData {
public *;
}
-keep class com.chance.engine.DownloadData {
public *;
}
-keep class com.chance.engine.DownloadSubData {
public *;
}
-keep class com.chance.data.DownloadCompleteInfo {
public *;
}
-keep class com.chance.data.AppInfo {
public *;
}
-keep class com.chance.util.PBLog {
public *;
```

```

}
-keep class com.chance.recommend.** {*; }
-keep class com.chukong.android.crypto.** {*; }
-keep class com.chance.d.** {*; }

#触控 混淆结束

-keep class com.suizong.mobile.** {*; }
-keep class com.go2map.mapapi.** {*; }

-keep public class cn.Immob.sdk.** {*; }
-keep public class cn.Immob.sdk.controller.** {*; }

-keep class net.youmi.android.** {*; }

-keepclassmembernames cn.smartmad.ads.android
-keepclassmembernames I
-keep class cn.smartmad.ads.android.* {*; }
-keep class I.* {*; }

-keep public class MobWin.*
-keep public class MobWin.cnst.*
-keep class com.tencent.lbsapi.*
-keep class com.tencent.lbsapi.core.*
-keep class LBSAPIProtocol.*
-keep class com.tencent.lbsapi.core.QLBSJNI {
*;
}

-keepclassmembernames com.adchina.android.ads
-keepclassmembernames com.adchina.android.ads.controllers
-keepclassmembernames com.adchina.android.ads.views
-keepclassmembernames com.adchina.android.ads.animations
-keep class com.adchina.android.ads.*{*; }
-keep class com.adchina.android.ads.controllers.*{*; }
-keep class com.adchina.android.ads.views.*{*; }
-keep class com.adchina.android.ads.animations.*{*; }

-optimizationpasses 5
-dontusemixedcaseclassnames
-dontskipnonpubliclibraryclasses
-dontpreverify

```

```

-verbose

-optimizations !code/simplification/arithmetic,!field/*,!class/merging/*

-keep public class * extends android.app.Activity
-keep public class * extends android.app.Application
-keep public class * extends android.app.Service
-keep public class * extends android.content.BroadcastReceiver
-keep public class * extends android.content.ContentProvider
-keep public class * extends android.app.backup.BackupAgentHelper
-keep public class * extends android.preference.Preference
-keep public class com.android.vending.licensing.ILicensingService

-keepclasseswithmembers class * {
    native <methods>;
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet);
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet, int);
}

-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}

-keep class * implements android.os.Parcelable {
    public static final android.os.Parcelable$Creator *;
}
-keep class com.mediax.** {*; }
-keep class org.adver.score.**{*; }

-keep class com.easou.ecom.mads.** {
    public protected *;
}
-keep class com.adroi.sdk.** {
    public protected *;
}
-keep class com.mopanspot.sdk.** {
    *;
}

```



```

}
-keep class com.imopan.plugin.spot.** {
    *;
}
-keep class com.jd.**{
    *;
}
-keep class cn.pro.ad.sdk.*

```

目前Adview SDK混淆支持proguard4.6以上的版本，开发者可以去proguard官方网站<http://sourceforge.net/projects/proguard/files/proguard/>下载4.6或以上版本；如果要升级的话，用下载的版本直接替换掉“android-sdk-windows\tools\proguard”就可以了；

10.2 如果有问题，和谁联系？

用户可以登录AdView，在首页底部，有服务邮箱，客服热线，企业QQ客服：

合作伙伴



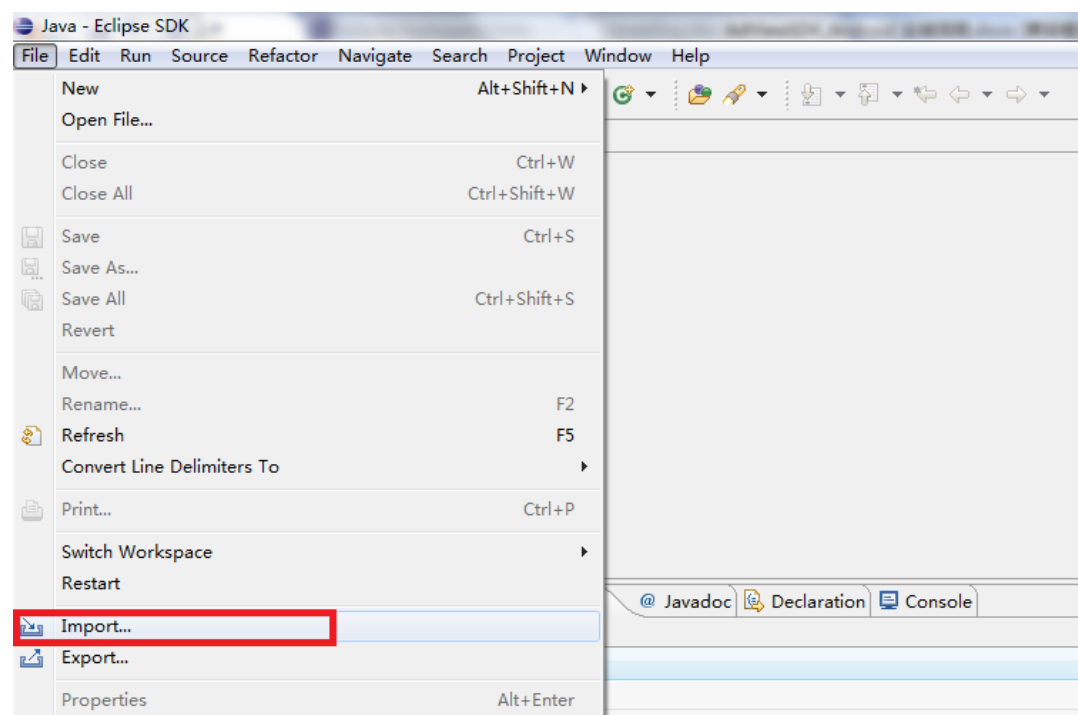
7X24服务热线：4000-1314-00 客服邮箱：service@kuaiyou.com 企业客服：4000-1314-00

产品合作	开发者中心	帮助中心	关注AdView微信	关注AdView微博
解决方案	Android SDK下载	AdView使用流程		
商务合作	iOS SDK下载	在线帮助		
合作伙伴	开源SDK下载	联系我们		
移动广告观察	在线文档			

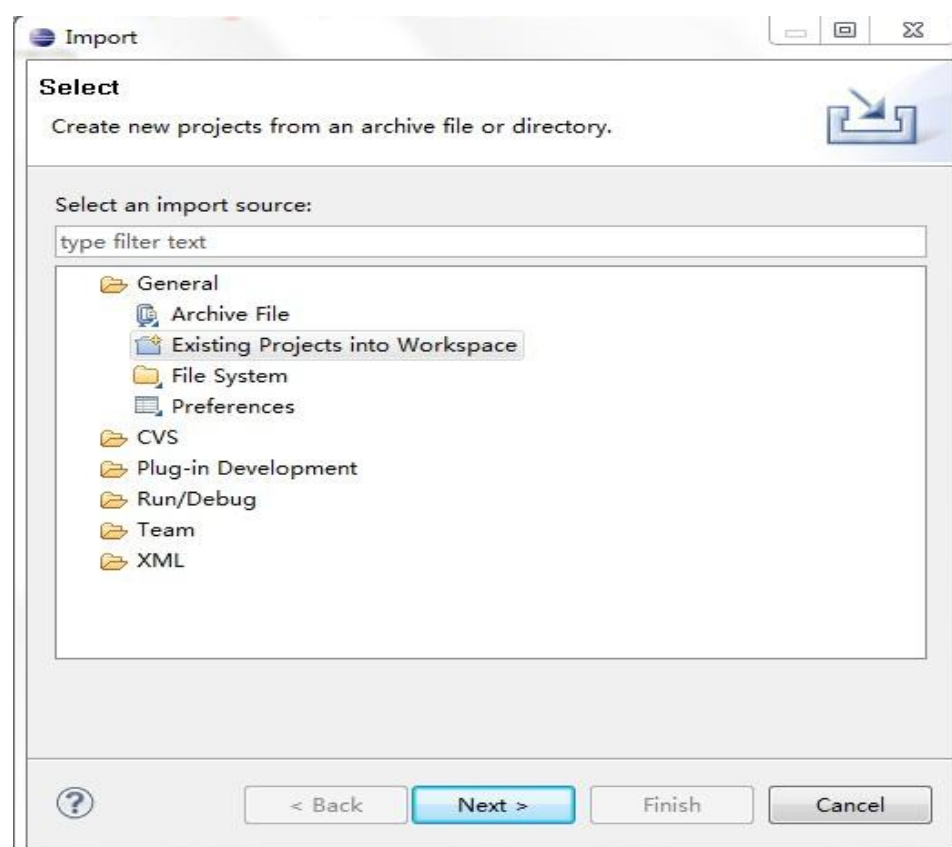
10.3 如何使用 sample

10.3.1 加载 sample 工程，方式一

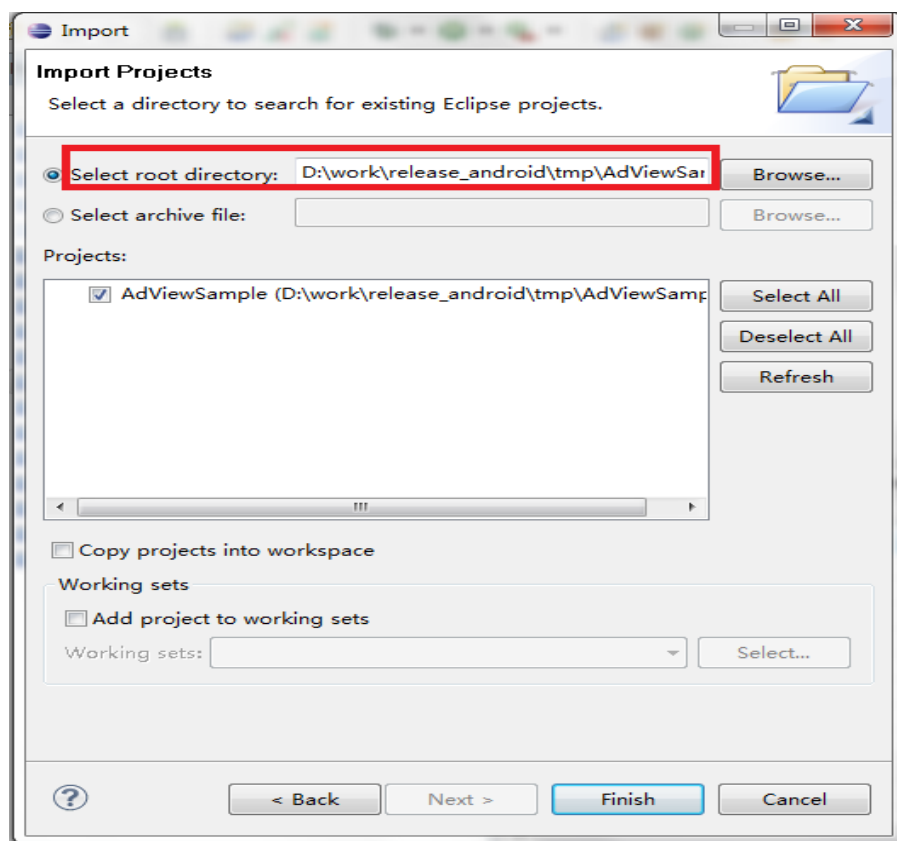
步骤一：



步骤二：



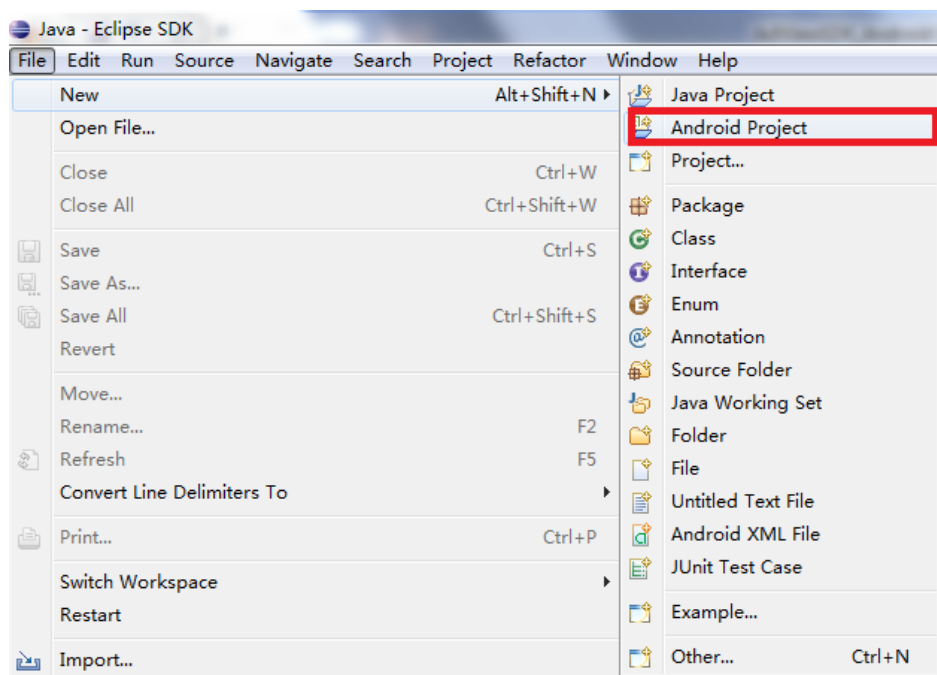
步骤三：



路径选择 sdk 包中的 sample 目录；

10.3.2 加载 sample 工程，方式二

步骤一：



步骤二:

路径选择 sdk 包中的 sample 目录;

默认 target 是 android1.6;

