

万普平台 Android 版 SDK 开发者手册

(标准版 Ver2.2.0)

平台简介

万普世纪移动营销服务平台(以下称为“万普平台”)的 Android 版 SDK 提供了一套现成的开发包及 Demo 源代码，便于开发者在 Android 应用中方便的集成万普平台的各项功能。

本文档描述了标准版 SDK 的用途与用法，并提供了示例代码。您仅需要在现有的应用中加入少量新代码，就可以集成万普平台的各项功能，轻松获得用户量和收入的倍增。

目录

平台简介	1
基础配置	2
集成代码	2
1. 数据统计接口	2
2. 虚拟货币接口	3
3. 积分墙接口	4
4. 互动广告接口	4
5. 插屏广告接口	5
6. 功能广告接口	6
7. 自定义广告接口	6
8. 迷你广告接口	7
9. 卸载广告接口	8
10. 工具组件接口	8
附表: 常用渠道编码表	10

基础配置

1. 加入 jar 包

将 Demo 解压包中 libs 目录中的 AppOffer_2.2.0.jar 包拷贝工程指定的 libs 目录，并执行 Build Path 将 jar 包引入。

2. 添加配置信息

将以下权限添加到 AndroidManifest.xml 文件中：

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

注：如果设置 `android:targetSdkVersion`，其值须设置为 15 或 15 以下

。

3. 混淆事项

如果应用进行了混淆编译，需要在混淆的配置文件中加入以下代码，避免 SDK 被二次混淆编译，才能确保 SDK 功能正常使用。如果开发者使用了定制名的 SDK，也需要对相应的包名加入类似代码：

```
-keep public class cn.waps.** {*; }
-keep public interface cn.waps.** {*; }

//对于使用 4.0.3 以上 android-sdk 进行项目编译时产生异常的情况时,加入以下内容:
-dontwarn cn.waps.**
```

集成代码

1. 数据统计接口

该接口是所有其他接口能正常使用的基础，在每次应用启动时，必须调用该接口进行初始化操作，才能保证获得准确的统计数据。

步骤 1: 在的应用中第一个 Activity 类中(启动的第一个类), 添加下面一行代码：

```
//方式①：通过代码设置 APP_ID 和 APP_PID
AppConnect.getInstance("APP_ID","APP_PID",this);
```

APP_ID 为应用标识，该值由万普后台添加应用后自动生成，点击“应用详情”获取；

APP_PID 为分发渠道标识，使用规则请参见本文档附表《常用渠道编码表》。

除了使用以上方法外，也可以通过在 **AndroidManifest** 内添加配置来设置 **APP_ID** 和 **APP_PID**：

```
<meta-data android:name="APP_ID"android:value="应用标识"/>
<meta-data android:name="APP_PID"android:value="分发渠道标识"/>
```

//方式②：通过 **AndroidManifest** 文件读取 **APP_ID** 和 **APP_PID**

```
AppConnect.getInstance(this);
```

注意：上传到万普平台官方合作商店（机锋、N 多、木蚂蚁、优亿、联想、MM、十字猫等）的，需要采用方式②初始化，并需要严格按照《常用渠道编码表》的要求设置对应的 **APP_PID** 标识，才能快速通过商店审核；而对于其他可能会限制第三方广告 **SDK** 的应用商店，需要采用方式①初始化，并结合万普“在线参数”功能进行广告显示控制，在通过商店审核并正常发布之后再开启广告。

步骤 2:在程序退出的处理方法中, 添加下面一行代码:

```
AppConnect.getInstance(this).close();
```

2. 虚拟货币接口

在应用中合理设置虚拟货币及消费机制，可促进用户参与应用内购买或参与广告活动，增强应用粘性，大幅提升收益。如果您的应用使用了虚拟货币功能，需使用此接口与服务器同步用户的虚拟货币余额。

2.1 获取用户虚拟货币

步骤 1: 从服务器端获取用户点数/虚拟货币余额:

```
AppConnect.getInstance(this).getPoints(this);
```

步骤 2: 确保在 **this** 类中实现 **UpdatePointsNotifier** 接口，实现下面的两个回调方法，用于异步接收服务器返回的结果:

```
public void getUpdatePoints(String currencyName, intpointTotal);//获取成功
public void getUpdatePointsFailed(String error);//获取失败
```

2.2 花费用户虚拟货币

```
AppConnect.getInstance(this).spendPoints(intamount,this);
```

调用此方法的响应结果将通过 **this** 中实现的 **UpdatePointsNotifier** 接口返回

2.3 奖励用户虚拟货币

```
AppConnect.getInstance(this).awardPoints(intamount,this);
```

调用此方法的响应结果将通过 **this** 中实现的 **UpdatePointsNotifier** 接口返回

注意：所有通过万普广告获得的虚拟货币将自动给予到用户，无需调用该接口。该接口仅用于用户完成了开发者指定的其他任务的情况下额外给予用户虚拟货币。

3. 积分墙接口

积分墙（也称 Offer，广告墙）是万普平台提供的一种集中展示型广告。开发者可在应用中合适的位置加入“推荐应用”、“免费赚积分”等类似字样的功能，获取更高的广告收益。添加如下代码，即可显示万普平台推荐应用列表：

```
AppConnect.getInstance(this).showOffers(this);
```

说明：以上方法调用的为综合积分墙。SDK 还提供了 `showAppOffers(this)`、`showGameOffers(this)` 两个接口，用于直接调用推荐软件列表、推荐游戏列表的积分墙。

提示：开发者可通过万普后台的“广告设置”功能，手动设置积分墙的显示样式和广告内容。

2.0.2 版开始，加入积分墙关闭监听接口

```
//设置关闭积分墙的监听接口，必须在showOffers接口之前调用
AppConnect.getInstance(this).setOffersCloseListener(new AppListener(){

    @Override
    public void onOffersClose() {
        // TODO 关闭积分墙时的操作代码
    }

});
```

4. 互动广告接口

互动广告是一个显示在应用内固定位置高度为 50 像素广告条，将自动显示万普平台提供的广告。结合虚拟货币功能使用，可获得最佳的广告效果和用户体验。

方式①：通过布局文件添加互动广告

步骤 1: 复制如下代码到相应 Activity 的 Layout 文件中,并放置在合适的位置:

```
<LinearLayout
    android:id="@+id/AdLinearLayout"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"/>
```

步骤 2: 在调用样式文件的 Activity 类中, 添加下面的代码:

```
LinearLayout adlayout =(LinearLayout)findViewById(R.id.AdLinearLayout);
AppConnect.getInstance(this).showBannerAd(this, adlayout);
```

方式②：纯代码模式添加互动广告

```
LinearLayout adlayout = newLinearLayout(this);
adlayout.setGravity(Gravity.CENTER_HORIZONTAL);
RelativeLayout.LayoutParams layoutParams = new
RelativeLayout.LayoutParams(ViewGroup.LayoutParams.FILL_PARENT,ViewGroup.LayoutPar
ams.WRAP_CONTENT);
AppConnect.getInstance(this).showBannerAd(this, adlayout);
layoutParams.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);//此代码可设置顶端或低端
this.addView(adlayout, layoutParams);
```

5. 插屏广告接口

插屏广告是 1.8 版本 SDK 开始推出的一种新型广告形式，以全屏对话框的新式显示广告，支持透明背景和浮层特效。和传统 Banner 广告相比，显示效果更炫，转化率更高，而且具有良好的用户体验。不占用固定广告位，特别适用于应用启动、游戏通关等场景使用。

始化（预先加载）广告数据：

```
AppConnect.getInstance(this).initPopAd(this);
```

说明：该方法将通过异步方式预先加载 3-5 条插屏广告到本地缓存，如果事先未通该方法初始化广告，或者在数据未加载完成的情况下直接显示插屏广告，将重新异步加载一次数据，等待数据加载完成后才能显示出广告，会造成一定的延时。

显示插屏广告：

```
AppConnect.getInstance(this).showPopAd(this);
```

说明：该方法将根据应用当前场景，用自适应方式显示插屏广告。

获取插屏广告对话框(可选)：

```
Dialog popAdDialog = AppConnect.getInstance(this).getPopAdDialog();
```

说明：该方法仅用于先通过调用 showPodAd 方法正常显示出插屏广告后，获取到插屏广告对话框实例，实现对插屏广告的显示过程灵活控制，比如监听广告关闭事件、定时自动关闭广告等自定义效果，但不能用于取代 showPodAd 方法直接显示插屏广告。当插屏广告尚未正常显示时，该方法可能会获取到空值。

获取插屏广告内嵌布局(可选)：

```
//获取默认宽高的插屏广告布局
View popAdDialog = AppConnect.getInstance(this).getPopAdView(this);//默认 400*400
//获取可自定义宽高的插屏广告布局
View popAdDialog = AppConnect.getInstance(this).getPopAdView(this, width, height);
```

说明：该接口可用于将插屏广告灵活插入到应用内，用于实现开屏广告、退屏广告、界面内嵌入式广告等灵活的广告形式。随本文档提供的 Demo 程序中，提供了一份完整的开屏广告和退屏广告的实现样例和源码，开发者可直接引用，也可进行任意修改用于实现更多的广告形式。

判断插屏广告是否初始化完成：

```
//判断插屏广告是否已初始化完成，用于确定是否能成功调用插屏广告
booleanhasPopAd = AppConnect.getInstance(this).hasPopAd(this);
```

6. 功能广告接口

功能广告是一种将应用功能和广告产品进行有效融合的新型广告形式。开发者可以通过万普平台提供的接口方法，轻松将广告产品的部分开放功能集成到自身应用，从而实现对特定广告定向推广的同时，为用户提供更有价值的服务。和其它广告形式相比，功能广告的用户体验更好，转化率更高，特别适用于需要调用浏览器、播放器等常用第三方功能的应用。

在应用启动的 Activity 中(loading 界面除外)的 onCreate 中调用初始化接口：

```
AppConnect.getInstance(this).initFunAd (this);
```

在需要执行某项功能的位置调用以下内容(暂时提供的是浏览器打开指定页面)：

```
String uriStr = "http://www.waps.cn";
方式一：AppConnect.getInstance(this).showBrowser(this, uriStr);
```

7. 自定义广告接口

开发者可通过自定义广告广告接口获取所有可用广告的元数据，自定义广告的展现方式，便于让广告内容完全和您的应用 UI 融合，可显著提升广告效果。

初始化（预先加载）广告数据：

```
AppConnect.getInstance(this).initAdInfo();
```

说明：该接口将在首次调用时，需要预先将所有可用广告的元数据（通常 20 条左右）全部获取到手机内存，便于程序随时快速调用。广告数据首次加载时间视网络情况决定，通常需要花几秒钟时间，因此，如果应用内需要使用自定义广告功能，建议在程序启动后，尽早先调用 initAdInfo()方法预先异步加载一次元数据，便于后续调用 getAdInfo()或 getAdInfoList()方法时能立即获得数据。

随机获取一条广告的元数据：

```

AdInfo adInfo = AppConnect.getInstance(this).getAdInfo();//每次调用将自动轮换广告
String adId      = adInfo.getAdId();      //广告 id
String adName    = adInfo.getAdName();    //广告标题
String adText    = adInfo.getAdText();    //广告语文字
Bitmap adIcon    = adInfo.getAdIcon();    //广告图标(48*48 像素)
int adPoint      = adInfo.getAdPoints();  //广告积分
String description = adInfo.getDescription(); //应用描述
String version   = adInfo.getVersion();   //程序版本
String filesize  = adInfo.getFileSize();  //安装包大小
String provider  = adInfo.getProvider();  //应用提供商
String[] imageUrls = adInfo.getImageUrls(); //应用截图的 url 数组，每个应用 2 张截图
String adPackage = adInfo.getAdPackage(); //广告应用包名
String action     = adInfo.getAction();   //用于存储“安装”或“注册”的字段

```

获取所有可用广告的元数据：

```
List<AdInfo>adInfoList = AppConnect.getInstance(this).getAdInfoList();
```

自定义广告事件处理接口：

```

//当广告被点击时，显示广告详情
AppConnect.getInstance(this).clickAd(this, adInfo.getAdId());

//当用户确认要下载广告应用时，启动下载（仅适用于应用下载类广告）
AppConnect.getInstance(this).downloadAd(this, adInfo.getAdId());

```

8. 迷你广告接口

迷你广告是一种基于自定义广告的广告形式，每个广告由一个微缩图标和文字链组成，高度仅为 24 像素，文字颜色、背景颜色及透明度可自定义，特别适用于广告位空间有限的游戏类应用。和传统的 Banner 大广告条相比，迷你广告采用了流量压缩和本地缓存方式轮换广告，不仅省广告位，而且非常省流量，广告轮换频率更高，相比传统 Banner 广告条有更高的收益。

步骤 1: 复制如下代码到相应 Activity 的 Layout 文件中,并放置在合适的位置:

```

<LinearLayout android:id="@+id/miniAdLinearLayout"
    android:layout_width="wrap_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"/>

```

步骤 2: 在调用样式文件的 Activity 类中, 添加下面的代码:

```
//设置迷你广告背景颜色
AppConnect.getInstance(this).setAdBackColor(Color.argb(50, 120, 240, 120));
//设置迷你广告广告语颜色
AppConnect.getInstance(this).setAdForeColor(Color.YELLOW);
//若未设置以上两个颜色，则默认为黑底白字
LinearLayout miniLayout =(LinearLayout)findViewById(R.id.miniAdLinearLayout);
AppConnect.getInstance(this).showMiniAd(this, miniLayout, 10); //默认 10 秒切换一次广告
```

9. 卸载广告接口

卸载广告是用户卸载 APP 后所展示的广告形式，兼容各种广告，从卸载用户身上获取额外收益，从而使利润发展到最大化，用户体验绝佳。

步骤 1：将 Demo 中 libs 里的 armeabi、armeabi-v7a、mips、x86 文件夹拷贝到开发者的工程下的 libs 目录下；

步骤 2：调用以下接口进行卸载广告的初始化：

```
AppConnect.getInstance(this).initUninstallAd(this);
```

10. 工具组件接口

(1)用户反馈接口：

```
AppConnect.getInstance(this).showFeedback(this);
```

说明：在应用内增加用户反馈功能可收集用户意见反馈，便于改进用户体验；开发者可通过管理后台查看用户的反馈信息。

注：2.0.0 版本中 showFeedback 加入 Context 参数

(2)自家应用列表：

```
AppConnect.getInstance(this).showMore(this);
```

说明：自家应用列表可采用类似积分墙的形式，列出将开发者自己在万普平台发布的所有应用供用户下载，便于开发者自家应用之间相互推荐。

使用重载方法指定 APP_ID，可直接展示某个应用的下载详情页面。调用方法如下：

```
AppConnect.getInstance(this).showMore(this, app_id);
```

(3)在线配置接口

```
//String value=AppConnect.getInstance(this).getConfig(String key);//不推荐使用该方式
String value=AppConnect.getInstance(this).getConfig(String key, String defaultValue);
```


说明：开发者可以通过万普平台管理后台的“在线配置”功能设置各种参数值，便于在线修改应用的各项配置。其中参数 **key** 为在线配置的参数 ID，通过此方法可获取到对应的值。

注意：该方法将在统计器初始化时返回所有参数值，缓存到本地调用，因此需在统计器初始化完成后再调用该方法，才能保障可靠获取到在线参数值。开发者可自行定义一个缺省值 **defaultValue**，如果在第一次统计器未初始化完成时调用，将使用该默认值来替代在线参数进行操作，避免造成线程阻塞。

系统同时提供了一个同步获取在线参数的接口（该方法会阻塞进程），确保不使用本地缓存，每次都从服务器获取参数值：

```
String value = AppConnect.getInstance(this).getConfig_Sync(String key);
```

2.0.2 版开始，getConfig 增加了一个带回调接口的异步方法

```
AppConnect.getInstance(this).getConfig(key, defaultValue, new AppListener(){  
  
    @Override  
    public void onGetConfig(String value) {  
        // value为异步获取到的在线参数key对应的值  
    }  
});
```

(4) 错误报告接口：

```
AppConnect.getInstance(this).setCrashReport(false); //默认值 true 开启，设置 false 关闭
```

说明：开发者可通过代码开启错误报告功能，并通过管理后台的“错误报告”功能随时查看收集到的错误报告，以便及时进行修正。

(5) 检查更新接口：

```
AppConnect.getInstance(this).checkUpdate(this);
```

说明：该接口为手动检查程序的最新版本，提醒用户下载升级程序。

开发者在万普平台“应用详情”中勾选了“检查新版本”选项，每次应用启动时将自动检查新版本。

开发者也可以关闭自动检查新版本功能，而是用该接口在指定位置来调用检查版本更新。

附表: 常用渠道编码表

对用于每个分发渠道的安装包设置一个不同的 APP_PID（渠道标识），可便于统计各渠道的用户数。万普平台的广告将针对各种主流分发渠道进行优化，在分发包内使用正确的指定渠道标识，可获得更好的推广效果；如果未设置正确的渠道标识，可能会导致无法通过部分应用商店的审核。

上传到万普平台的版本请填写“default”，分发给其他渠道的版本填写相应值（比如“Google”）后重新打包即可；该值必须由**英文字母和数字组成，不能是纯数字，不能包含空格或中文**，下面是一些渠道 APP_PID 的官方写法:(* 标注的为万普官方合作商店，须在 AndroidManifest.xml 里添加渠道标识)

APP_PID	渠道名称
default	万普平台(默认)
google	GooglePlay 市场
91	91 手机助手
hiapk	安卓市场
gfan	机锋市场*
goapk	安智市场
appChina	掌上应用汇
mumayi	木蚂蚁市场*
eoe	优亿市场*
nduo	N 多市场*
feiliu	飞流下载
crossmo	十字猫商店*
huawei	华为智汇云

APP_PID	渠道名称
QQ	腾讯应用
3G	3G 安卓市场
360	360 应用
baidu	百度应用
sohu	搜狐应用
163	网易应用*
UC	UC 应用商店
dangle	当乐游戏中心
samsung	三星乐园
mmw	移动 MM 商店
xiaomi	小米市场
lenovo	联想乐商店*
nearme	NearMe 商店

请通过正常方式推广您的产品，不得通过 QQ 群、论坛、朋友圈等方式诱导用户大量点击广告，造成严重刷量者将封号处理，请查看官网公告  <http://www.waps.cn/news.jsp?id=1433&type=new>

万普平台: <http://www.waps.cn> 技术支持: 小万 QQ: 909155540 电话: 4006-178-378