

AdsMOGO Android SDK 信息流自定义平台使用说明

基于 AdsMOGO_Android_SDK 使用说明，请先阅读此文档

一、什么是自定义广告平台

当您使用的广告平台在 AdsMogo 中不支持时，可以使用“自定义广告平台”，这是一个很好的解决办法。您可以使用“自定义平台”去启动任何代码。当您需要使用那些广告平台时，您需要在“自定义平台”中添加这个平台名称并填写相应的 key 或 id，开启它。在程序中通过一个 Adapter（适配器）来工作。

注：自定义平台只适用于芒果不支持的广告平台，如果自定义芒果支持的平台有可能对芒果自有的系统造成影响，因此，请不要再自定义芒果已经支持的平台。

二、工作流程

首先，您需要在 App 配置页面添加一个“自定义广告平台”，填写相应平台 key。随后，您就可以开启使用该自定义平台。当这个“自定义平台”被轮换到时，就会被调用。在您的 Activity 中，即在您加载 AdsMogo 广告窗体的 Activity 中，定义返回您写好的 Adapter，AdsMogo 广告系统就可以使用这个 Adapter。您可以在这个方法中完成任何功能，当然也包括在这个方法里去执行别的广告平台，来将 AdsMogo 正在展示的 banner 内容换成其他平台的广告，或者任何其他广告。

三、后台配置

- 1，登录后台网站 <http://my.adsmogo.com>；
- 2，进入某一 App 的广告平台配置页面，点击“自定义广告平台”；
- 3，填写平台名称及该平台 key；
- 4，开启该平台并设置投放量后，点击保存。



请注意，每个广告平台列表中，投放总数必须为100%或者全部关闭，否则无法进行下一步操作。



四、接口和使用方法（详细使用请参考 demo 中代码）

1，使用芒果 SDK 自定义平台功能，首先必须实现 AdsMogoNativeListener 回调中的 getCustomEventPlatformAdapterClass()方法：

实现 AdsMogoNativeListener 回调中的 getCustomEventPlatformAdapterClass()方法，并在该方法中 return 继承于 AdsMogoNativeCustomEventPlatformAdapter 类的自定义 adapter 类的 class 对象。芒果 SDK 会在需要请求自定义平台时，会调用该类对象的请求广告方法，来达到请求广告。

```
@Override
    public Class<? extends AdsMogoNativeCustomEventPlatformAdapter>
getCustomEventPlatformAdapterClass(

        com.adsmogo.natives.adapters.AdsMogoCustomEventPlatformEnum
enumIndex) {
    if(AdsMogoCustomEventPlatformEnum.AdsMogoCustomEventPlatform_1
        .equals(enumIndex)) {
        return CustomAdapter1.class;
    }else
    if(AdsMogoCustomEventPlatformEnum.AdsMogoCustomEventPlatform_2
        .equals(enumIndex)){
        return CustomAdapter2.class;
    }
    return null;
}
```

2. 编写自定义 adapter 类：

编写自定义 adapter 类，继承于 AdsMogoNativeCustomEventPlatformAdapter 父类，并且必须实现 startRequestNativeAd(int reqAD)这个父类方法，在

startRequestNativeAd 方法中编写初始化、请求单一平台 sdk 代码，当获取到广告信息时需要通过 addNativeAdInfo()方法把广告信息添加到芒果数据列表中，当把所有广告信息都添加到芒果数据列表后需要通知芒果 SDK 广告请求成功：

notifyAdsmogoAdRequestAdSuccess，广告请求失败时需要通知芒果 SDK 广告请求失败：notifyAdsmogoAdRequestAdFail。

```
/**
 * reqAD 需要请求广告条数
 */

@Override
```

```
public void startRequestNativeAd(int reqAD) {  
  
    // TODO Auto-generated method stub  
  
    gdtnativead = new GDTNativeAd(activity, getAppID_1(), getAppID_2(),  
        new GDTNativeAdListener() {  
  
        @Override  
  
        public void onGDTNativeAdLoaded(  
            List<GDTNativeAdDataRef> refs) {  
  
            for (final GDTNativeAdDataRef ref : refs) {  
  
                HashMap<String, Object> content = new  
HashMap<String, Object>();  
  
                try {  
  
                    //具体key-Value 描述可以参考原生广告信息流植  
入文档  
  
                    content.put(AdsMogoNativeKey.TITLE, ref.getTitle()); //标题  
                    content.put(AdsMogoNativeKey.DESCRPTION, ref.getDesc()); //描述  
                    content.put(AdsMogoNativeKey.LINK, ""); //点击后的连接  
                    content.put(AdsMogoNativeKey.LATYPE, "10"); //点击后的类型  
                    content.put(AdsMogoNativeKey.RATING, ""); //评级  
                    content.put(AdsMogoNativeKey.ICON_URL, ref.getIconUrl()); //icon地址
```

```
content.put(AdsMogoNativeKey.ICON_WIDTH, "0");//icon宽度
content.put(AdsMogoNativeKey.ICON_HEIGHT, "0");//icon高度
content.put(AdsMogoNativeKey.IMAGE_URL, ref.getImgUrl());//大图地址
content.put(AdsMogoNativeKey.IMAGE_WIDTH, "0");//大图宽
content.put(AdsMogoNativeKey.IMAGE_HEIGHT, "0");//大图高
content.put(AdsMogoNativeKey.RATION_NAME, "自定义平台广点通");//平台
```

名称

```
    } catch (Exception e) {
        L.e(AdsMogoUtil.ADMOGO, "gdt fail error:"
            + e.getMessage());
    }
```

```
//添加广告信息到芒果信息列表同时传入广告被点击和
```

展示监听器

```
        addNativeAdInfo(content, new
AdsMogoNativeAdapterListener() {
    @Override
    public void onClickAd() {
        // TODO Auto-generated method stub
        //当前广告被点击后执行该方法
        ref.onClicked();
    }
}
```

```
@Override

    public void onAttachAdView(ViewGroup view)
{
    // TODO Auto-generated method stub

    //当前广告被展示执行该方法

    ref.onExposed(view);

}

});

}

L.d(AdsMogoUtil.ADMOGO, "gdt request success");

//通知芒果广告请求成功

notifyAdsmogoAdRequestAdSuccess();

}

@Override

    public void onGDTNativeAdFail(int errorCode) {

        L.e(AdsMogoUtil.ADMOGO,

            "gdt request fail ,errorCode is"

                + errorCode);

        // 通知芒果广告请求失败

        notifyAdsmogoAdRequestAdFail();

    }

});
```

```
gdtnativead.loadAd(reqAD);  
}
```

3.自定义广告平台提供可以调用的方法

```
/**  
 * 提供给开发者调用,用来通知芒果SDK广告请求展示失败  
 */  
protected void notifyAdsmogoAdRequestAdFail()  
  
/**  
 * 提供给开发者调用,用来通知芒果SDK广告展示成功  
 */  
protected void notifyAdsmogoAdRequestAdSuccess()  
  
/**  
 * 返回在服务器端配置的广告平台的名称  
 * @return 广告平台的名称,错误则返回空  
 */  
protected String getAdsPlatformName()  
  
/**  
 * 返回在服务器端配置的APPID-1  
 */  
public String getAPPID_1()  
  
/**  
 * 返回在服务器端配置的APPID-2  
 */  
public String getAPPID_2() {
```