

AdsMOGO Android SDK 自定义广告平台使用说明

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。



配置自定义广告平台一

平台名称

APPID-1

APPID-2

保存

平台对应ID

自定义广告平台 [?]

未设置

投放总数为： 100%

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。

保存 取消

已开启补余 创建广告 删除 跳转到→

横幅广告 全屏广告 积分墙

地域优化 ☐ OFF 点击优化 ☒ ON 关闭广告 ☒ ON 一键配置

广告平台	评分 <small>与评论得现金</small>	状态	投放量	优先级
自定义平台1		<input checked="" type="checkbox"/> ON	- %	<input checked="" type="checkbox"/> 优先
有米	-分	<input type="checkbox"/> OFF	- %	+
微云	-分	<input type="checkbox"/> OFF	- %	+

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 `AdsMogoInterstitialListener` 回调中的 `getCustomEventPlatformAdapterClass()` 方法：

实现 `AdsMogoInterstitialListener` 回调中的 `getCustomEventPlatformAdapterClass()` 方法，并在该方法中 `return` 继承于 `AdsMogoInterstitialCustomEventPlatformAdapter` 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
AdsMogoInterstitialCustomEventPlatformAdapter>
getCustomEventPlatformAdapterClass(
        com.adsmogo.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 AdsMogoInterstitialCustomEventPlatformAdapter 父类，并且必须实现 startRequestInterstitialAd ()、startShowInterstitialAd()和 onFinishClearCache()这三个父类方法 在 startRequestInterstitialAd 方法中编写初始化、请求单一平台广告代码。**广告预加载成功（请求成功）时通知芒果 SDK 广告预计在成功：notifyAdsmogoAdRequestAdSuccess。广告展示成功是需要通知芒果 SDK 广告请求成功：notifyAdsmogoAdShowSuccess。**广告请求失败或展示失败时需要通知芒果 SDK 广告请求失败：notifyAdsmogoAdRequestAdFail。全插屏广告被点击时通知芒果 SDK 广告被点击：notifyAdsmogoAdClicked。全插屏广告关闭时通知芒果 SDK 广告已经关闭：notifyAdsmogoAdCloseed。在 startShowInterstitialAd 方法中需要编写展示单一平台插屏广告的代码。在 onFinishClearCache 方法中编写销毁单一平台对象的代码。

```
/**
 * （开发者必须实现）开始请求单一平台广告方法
 */
@Override
public void startRequestInterstitialAd() {
    // 获取当前的Activity对象，如果Activity对象为空，则通知芒果请求广告失败并终止请求
    Activity activity = getCurActivity ();
    if (activity == null) {
        // 通知芒果请求广告失败
        notifyAdsmogoAdRequestAdFail();
        // 终止请求
        return;
    }
    // 实例化单一平台广告SDK
    adInterstitial = new AdInterstitial(activity, getAppID_1());
    // 设置请求广告回调，全插屏需要有请求广告成功回调、请求广告失败回调、广告准备完成回调、广告关闭回调
    adInterstitial.setAdInterstitialListener(new AdInterstitialListener() {

        @Override
        public void onReceivedItstAd() {
            // 广告加载完毕通知芒果SDK广告准备完成

            AdchinaInterstitialCustomAdapter.this.notifyAdsmogoAdRequestAdSuccess();
        }

        @Override
        public void onFailedToReceiveItstAd() {
            // 请求广告失败时通知芒果SDK广告请求失败
            AdchinaInterstitialCustomAdapter.this
                .notifyAdsmogoAdRequestAdFail();
        }

        @Override
        public void onDisplayItst() {
            // 全插屏广告显示时通知芒果SDK广告请求成功
            AdchinaInterstitialCustomAdapter.this
                .notifyAdsmogoAdShowSuccess();
        }

        @Override
```

```
public void onCloseItst() {
    // 全插屏广告关闭时时通知芒果SDK广告已经关闭（芒果SDK会在广告关闭后开始倒计时请求下一条广告，如果为调用过notifyAdsmogoAdCloseed可能造成广告不会再次请求）

    AdchinaInterstitialCustomAdapter.this.notifyAdsmogoAdCloseed();
}

@Override
public void onClickItst() {
    // 全插屏广告点击时通知芒果SDK广告被点击

    AdchinaInterstitialCustomAdapter.this.notifyAdsmogoAdClicked();
}
});
//开始请求自定义单一平台全插屏广告
adInterstitial.start();
}

/**
 * （开发者必须实现）广告轮换结束，在此需要销毁自定义单一平台广告对象。
 */
@Override
public void onFinishClearCache() {
    // 广告轮换结束，在此需要销毁自定义单一平台广告对象
    if (adInterstitial != null) {
        adInterstitial.stop();
        adInterstitial = null;
    }
}

/**
 * （开发者必须实现）显示全插屏广告： 在需要显示全插屏广告的时候，芒果SDK会调用方法，开发者需要在该方法中将全插屏广告显示出来
 */
@Override
public void startShowInterstitialAd() {
    // 获取当前的Activity对象，如果Activity对象为空或已经Finish，则通知芒果请求广告失败并终止展示
    Activity activity = getCurActivity ();
    if (activity == null) {
        notifyAdsmogoAdRequestAdFail();
        return;
    }
}
```

```
}  
if (activity.isFinishing()) {  
    notifyAdsmogoAdRequestAdFail();  
    return;  
}  
if (adInterstitial == null) {  
    notifyAdsmogoAdRequestAdFail();  
    return;  
}  
// 显示自定义单一平台全插屏广告  
adInterstitial.showItst();  
}
```

3.自定义广告平台提供可以调用的方法

```
/**  
 * 提供给开发者调用的 通知芒果SDK广告准备完成即广告请求成功  
 */  
protected void notifyAdsmogoAdRequestAdSuccess()  
  
/**  
 * 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 notifyAdsmogoAdShowSuccess ()  
  
/**  
 * 记录广告点击计数  
 */  
protected void notifyAdsmogoAdClicked()  
  
/**  
 * 提供给开发者调用的 通知芒果SDK广告关闭  
 */  
protected void notifyAdsmogoAdCloseed()
```

```
/**
 * 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()

/**
 * 获取当前activity对象
 */
protected Activity getCurrActivity ()

/**
 * 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() {
```