

AdsMOGO iOS SDK 信息流自定义平台使用说明

基于 AdsMOGO_iOS_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 View 中，即在您加载 AdsMogo 广告窗体的 View 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。

平台名称	APPID-1	APPID-2	投放量	优先级
LSense	-	未设置	- %	+
AirAd	-	未设置	- %	+
...
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

广告平台	收入分析	状态	投放量	优先级
自动优化	-	<input type="checkbox"/>	- %	+
自定义平台一	-	<input checked="" type="checkbox"/>	100 %	+

四、自定义横幅接口介绍和使用方法

首先创建一个类作为 Adapter

引入 AdMoGoNativeAdapterCustomEvent.h

继承 AdMoGoNativeAdapterCustomEvent

1, 需要在实现文件中重新实现三个方法：

```

+ (void)load;
+ (AdMoGoNativeAdNetworkType)networkType ;
+ (void)registerClass:(id)clazz;
- (void)loadAd:(int)adcount;
    
```

其中

a: **+(void)load** 是 NSObject 中的方法,在此重新实现是为了将自定义的 Adapter 注册进 Mogo 的 SDK 中,具体方法如下:

```
/**
 *注册adapter
 *将自定义的adapter注册入Mogo
 */
+(void)load{
    [super registerClass:self];
}
```

b: **+ (AdMoGoNativeAdNetworkType)networkType** 方法用来标记当前的 Adapter 对应的是哪个自定义平台,mogo 暂时支持最多 3 个自定义平台,他们的定义分别为:

```
// AdMoGoNativeAdNetworkTypeCustomEventPlatform_1
// AdMoGoNativeAdNetworkTypeCustomEventPlatform_2
// AdMoGoNativeAdNetworkTypeCustomEventPlatform_3
```

将其对应作为返回值即可,具体方法如下:

```
+(AdMoGoNativeAdNetworkType)networkType{
return    AdMoGoNativeAdNetworkTypeCustomEventPlatform_1;
}
```

c: **-(void)loadAd:(int)adcount** 这个方法用于初始化信息流广告,并开始请求。一般的应该在此方法中初始化用户所需要添加平台信息流广告对象,设置请求信息流广告个数(个数可选),设置信息流广告回调对象等。

下面是初始化 inmobi 的示例:

```
//通知Mogo请求广告开始并且获得广告尺寸类型
```

```
-(void)loadAd:(int)adcount{

    [InMobi setLogLevel:IMLogLevelVerbose];

    id key = [self.ration objectForKey:@"k"];

    NSString *publisherAppId = nil;

    if ([key isKindOfClass:[NSString class]]) {

        publisherAppId = key;
    }
}
```

```
}else if ([key isKindOfClass:[NSDictionary class]]){

    NSDictionary *keydict = (NSDictionary *)key;

    publisherAppId = [keydict objectForKey:APPID_Native_F];

}else{

    [self adMoGoNativeCustom:self didFail:-1];

    return;

}

[InMobi setLogLevel:IMLogLevelVerbose];

[InMobi initialize:publisherAppId];

self.native = [[IMNative alloc] initWithAppId:publisherAppId];

self.native.delegate = self;

[self.native loadAd];

}
```

2. 广告请求成功封装信息流数据，广告请求失败接口介绍

a : -(void)adMoGoNativeCustom:(AdMoGoNativeAdNetworkAdapter *)adapter
didReceiveAd:(NSArray*)adList

通知 MogoSDK 广告请求成功并且将 adList（封装后的平台信息流广告对象集合）返回 MogoSDK。

b: -(void)adMoGoNativeCustom:(AdMoGoNativeAdNetworkAdapter *)adapter
didFail:(int)errorCode ;

通知 MogoSDK 广告请求失败,继续轮换其他的平台。

c: 信息流广告对象封装

在自定义信息流平台 SDK 请求成功的消息中，获取平台返回的信息流对象，平台返回的信息流对象 属性请参考各平台定义。开发者需要将平台的信息流对象，统一封装为芒果的信息流对象 AdsMogoNativeAdInfo。下面是 inmobi 的封装示例：

```
// 消息 -(void)nativeAdDidFinishLoading:(IMNative*)native 是inmobi信息流请求成功消息
-(void)nativeAdDidFinishLoading:(IMNative*)native
{
    NSError *error;
    NSData *data = [native.content dataUsingEncoding:NSUTF8StringEncoding];
    NSDictionary *imobiDict = [NSJSONSerialization JSONObjectWithData:data options:NSJSONReadingMutableLeaves error:&error];
    NSMutableArray *mutableArray = [[NSMutableArray alloc] init];
    AdsMogoNativeAdInfo *adsMogoNativeInfo = [[AdsMogoNativeAdInfo alloc] init];

    // AdsMoGoNativeMoGoTitle 定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"title"] forKey:AdsMoGoNativeMoGoTitle];

    // AdsMoGoNativeMoGoIconUrl定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"icon"] objectForKey:@"url"] forKey:AdsMoGoNativeMoGoIconUrl];

    // AdsMoGoNativeMoGoIconWidth定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"icon"] objectForKey:@"width"] forKey:AdsMoGoNativeMoGoIconWidth];

    // AdsMoGoNativeMoGoIconHeight定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"icon"] objectForKey:@"height"] forKey:AdsMoGoNativeMoGoIconHeight];

    // AdsMoGoNativeMoGoDesc定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"description"] forKey:AdsMoGoNativeMoGoDesc];

    // AdsMoGoNativeMoGoImageUrl定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"screenshots"] objectForKey:@"url"] forKey:AdsMoGoNativeMoGoImageUrl];

    // AdsMoGoNativeMoGoImageWidth定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"screenshots"] objectForKey:@"width"] forKey:AdsMoGoNativeMoGoImageWidth];

    // AdsMoGoNativeMoGoImageHeight定义在AdMoGoNativeUserKey.h
    [adsMogoNativeInfo setValue:[imobiDict objectForKey:@"screenshots"] objectForKey:@"height"] forKey:AdsMoGoNativeMoGoImageHeight];

    [adsMogoNativeInfo setValue:imobiDict forKey:AdsMoGoNativeMoGoPdata];
    [adsMogoNativeInfo setValue:[self getMogoJsonByDic:adsMogoNativeInfo] forKey:AdsMoGoNativeMoGoJsonStr];
    [mutableArray addObject:adsMogoNativeInfo];
    [adsMogoNativeInfo release];
    [self adMoGoNativeCustom:self didReceiveAd:mutableArray];
}
```

注意以上例子中AdsMoGoNativeMoGoTitle、AdsMoGoNativeMoGoIconUrl等定义在AdMoGoNativeUserKey.h具体解释请参照AdMoGoNativeUserKey.h中注释。

3. 展示广告，点击广告接口

a : `-(void)attachAdView:(UIView*)view nativeData:(AdsMogoNativeAdInfo*)info;`

开发需要在这个方法中调用`[super attachAdView:view nativeData:info];`并且根据平台的要调用平台展示广告的相应消息。下面是inmobi的平台示例：

```
-(void)attachAdView:(UIView*)view
```

```
nativeData:(AdsMogoNativeAdInfo*)info{

    [super attachAdView:view nativeData:info];

    // self.native 是inmobi的信息流对象，inmobi 要求在展示广告调用inmobisdk中attachToView方法

    [self.native attachToView:view];

}
```

b: `-(void)clickAd:(AdsMogoNativeAdInfo*)info`

开发需要在这个方法中调用[super clickAd:info];并且根据平台的要求调用平台点击广告的相关消息。下面是inmobi的平台示例：

```
-(void)clickAd:(AdsMogoNativeAdInfo*)info{

    [super clickAd:info];

    NSDictionary *dict = [info valueForKey:AdsMoGoNativeMoGoPdata];

    NSString *url = [dict valueForKey:@"landingURL"];

    NSURL* URL = [NSURL URLWithString:url];

    [[UIApplication sharedApplication] openURL:URL];

    [self.native handleClick:nil];

}
```