

ADMobile Android Sdk——接入文档

V2.4.7.03252

目录

ADMobile Android Sdk——接入文档 V2.4.7.03252

1. 概述
2. 支持的广告类型
3. 修订历史(仅展示最近三个版本)
4. Demo及SDK下载链接
5. SDK接入流程
 - 5.1 添加SDK到工程中
 - 5.1.1 远程依赖集成方式
 - 5.1.2 本地AAR集成方式
 - 5.1.3 注意事项
 - 5.2 OAID支持
 - 5.3 权限申请
 - 5.4 示例工程Demo导入说明
6. 接入代码
 - 6.1 SDK初始化
 - 6.2 开屏广告示例
 - 6.3 Banner横幅广告示例
 - 6.4 信息流原生模板广告示例
 - 6.5 自渲染2.0广告示例
 - 6.5 插屏广告示例
 - 6.7 激励视频广告示例
 - 6.8 DrawVod广告示例
 - 6.9 全屏视频广告示例
 - 6.10 原生自渲染广告示例（已过时）
7. 适配问题
 - 7.1 Android7.0适配
 - 7.2 Android8.0、9.0适配
8. 混淆配置
9. 常见问题和说明
 - 9.1 下载弹框样式修改
 - 9.2 SDK版本号相关信息
 - 9.3 Q&A和错误信息对照表
- 10.商务合作

1. 概述

尊敬的开发者朋友，欢迎您使用艾狄墨搏集合SDK。通过本文档，您可以快速完成广告SDK的集成。

如果您的支持库是AndroidX而非Support请点击跳转至[Q&A文档并参考Q&A文档2.4进行适配](#)

2. 支持的广告类型

类型	简介	适用场景
开屏广告	开屏广告以APP启动作为曝光时机的模板广告，需要将开屏广告视图添加到承载的广告容器中，提供5s可感知广告展示	APP启动界面常会使用开屏广告
Banner广告	Banner广告是横向贯穿整个可视页面宽高比为640:100的模板广告，需要将Banner广告视图添加到承载的广告容器中	任意界面的固定位置，不建议放在RecyclerView、List这种滚动布局中当item
信息流模板广告	信息流模板广告是有多种样式（上图下文、左图右文、双图双文、三小图等）的模板广告，需要将返回的广告视图添加到承载的广告容器中，支持样式微调	信息流列表，轮播控件，固定位置都是较为适合
自渲染2.0广告	自渲染2.0能够返回标题，描述，图片链接等信息，用于用户的自定义样式展示。在有视频的情况下，还会直接返回视频控件，用户只需把视频控件按规则添加到指定位置即可，无须关心视频的播放、缓存等问题。	信息流列表，轮播控件，固定位置都是较为适合
激励视频广告	将短视频融入到APP场景当中，用户观看短视频广告后可以给予一些应用内奖励	常出现在游戏的复活、任务等位置，或者网服类APP的一些增值服务场景
插屏广告	插屏广告是移动广告的一种常见形式，在应用流程中弹出，当应用展示插屏广告时，用户可以选择点击广告，访问其目标网址，也可以将其关闭并返回应用	在应用执行流程的自然停顿点，适合投放这类广告
全屏视频广告	全屏视频广告的效果为播放全屏的视频，视频播放一定时间后出现可关闭或者跳过按钮，可提前结束广告播放无需全程观看完	在应用执行流程的自然停顿点，适合投放这类广告

3. 修订历史(仅展示最近三个版本)

版本号	修订说明	修订时间
2.4.5.01182	1、ADMobGenSdk优化； 2、升级第三方SDK； 3、其他已知问题优化；	2020.02.06
2.4.6.03022	1、ADMobGenSdk优化； 2、DrawVod类型广告重新支持； 3、新增远程依赖集成方式（可选择是否包含第三方SDK两种）； 4、第三方SDK升级； 5、升级移动安全联盟(MSA)的OAID支持库，当前最新版本为msa_mdid_1.0.13.aar（上一版本为miit_mdid_1.0.10.aar），由于该SDK过去几个版本存有较大的兼容性问题，强烈建议使用当前广告SDK使用的版本；此外msa_mdid_1.0.13.aar暂时无法兼容Inmobi平台，等Inmobi平台更新SDK后我们会再做跟进； 6、广点通信息流广告关闭按钮不再接管； 7、其他已知问题优化；	2020.03.09
2.4.7.03252	1、ADMobGenSdk优化； 2、升级第三方SDK； 3、广点通强制要求banner2.0广告 4、广点通兼容androidx； 5、inmobi 仅支持OAID最新版本，当前最新版本为msa_mdid_1.0.13.aar(上一版本为miit_mdid_1.0.10.aar，不再支持)； 6、其他已知道问题优化；	2020.03.25

PS1：更新SDK时，为防止不必要问题，请将使用到的SDK均替换成最新版本；

PS2：视频广告需要满足资源路径与ApplicationId一致，即AndroidManifest.xml中的package与build.gradle中的ApplicationId一致；

PS3：建议在适当的时机释放广告资源，例如onDestroy生命周期或者根据业务需求在合适的时机释放，释放后的广告不能再次使用；

4. Demo及SDK下载链接

[ADMobGenDemo头条商务版链接](#)

PS 1: 我们的头条SDK是商务版版本，如果开发者的头条账号是自注册版，也可使用该账号下载的open_ad_sdk.aar结合我们的头条适配器（本地的toutiao-release-x.x.x.x.aar或远程依赖com.admobile.ad:adapter_toutiao_without:x.x.x.x），商务版和自注册版区分请咨询商务人员；

PS 2: 由于广点通新版本开屏广告有所变动，我们广点通适配器不再支持低于4.170.1040版本的广点通sdk。如果开发者需要自己集成广点通SDK，则务必选择不低于4.170.1040版本的广点通SDK；

5. SDK接入流程

5.1 添加SDK到工程中

接入环境：**Android Studio**。

接入方式：2.4.6.03022版本起，新增加远程依赖的接入方式，建议使用该接入方式，后续版本本地aar接入方式有可能会不再维护，目前还会继续维护本地aar接入的方式；远程依赖方式默认会将第三方SDK一起集成，如果想自行集成第三方的SDK（我们开发是基于第三方SDK某个特定版本，如果自行集成第三方SDK需要自行承担版本不一致或第三方配置等问题带来的风险），可使用without的依赖（目前头条、广点通、百度支持），以广点通为例如下。

```
// 广点通平台依赖，默认已将广点通的SDK一起集成
implementation 'com.admobile.ad:adapter_gdt:2.3.4.1'
```

```
// 广点通平台依赖，没有将广点通的SDK集成，需要用户自行集成广点通SDK
// 适配器仅支持4.170.1040及以上版本的广点通SDK
implementation 'com.admobile.ad:adapter_gdt_without:2.3.4.1'
```

5.1.1 远程依赖集成方式

首先需要在项目build.gradle文件中引入相关配置，如下：

```
allprojects {
    repositories {
        ...
        // 添加以下仓库地址
        maven { url "http://101.37.191.20:9091/repository/maven-releases/" }
    }
}
```

然后再将广告所需要的依赖集成进去，其中com.admobile.ad:admobgensdk和com.admobile.ad:adapter_admobile以及msa_mdid是必须的，其他的可根据接入平台情况进行选择接入（需要接入哪些平台，请联系对接商务人员）。

```
dependencies {
    // admobgensdk和admobile以及oaid (msa_mdid) 是必须的
    implementation 'com.admobile.ad:admobgensdk:2.4.7.03252'
    implementation 'com.admobile.ad:adapter_admobile:2.4.7.1'
    implementation 'com.admobile:oaid:1.0.13'

    // 广点通平台需要的
    implementation 'com.admobile.ad:adapter_gdt:2.3.4.1'
    // 如自行集成广点通SDK，仅支持4.170.1040以上版本的广点通SDK
```

```
//      implementation 'com.admob.ad:adapter_gdt_without:2.3.4.1'

// 头条广告平台所需要的
implementation 'com.admob.ad:adapter_toutiao:2.4.7.1'
//      implementation 'com.admob.ad:adapter_toutiao_without:2.4.7.1'

// 百度广告平台所需要的
implementation 'com.admob.ad:adapter_baidu:1.2.3.2'
//      implementation 'com.admob.ad:adapter_baidu_without:1.2.3.2'

// inmobi平台所需要的（Inmobi暂不支持without，还需要V4和RecyclerView）
// picasso 依赖库添加失败会导致Inmobi广告无法请求
implementation 'com.admob.ad:adapter_inmobi:1.2.3.1'
implementation 'com.squareup.picasso:picasso:2.5.2'

// mobvsita平台所需要的（Mobvsita暂不支持without）
// 需要将Android Support v4包升级到26.0.0或以上。
implementation 'com.admob.ad:adapter_mobvsita:1.2.3.1'
}
```

5.1.2 本地AAR集成方式

Demo中libs文件目录下有所需要的依赖包，其中`admobgensdk-release.aar`和`admob-release`以及`msa_mdi`是必须的，其他依赖包请根据接入平台情况进行选择接入（需要接入哪些平台，请联系对接商务人员）。将所需要的aar包拷贝到libs目录下，并且在build.gradle中进行相应的配置。

```
android {
    ....
    //1. 添加依赖库目录
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
}

dependencies {
    ....
    //2. 添加广告依赖包

    // 必须的
    compile(name: 'admobgensdk-release-x.x.x.x', ext: 'aar')
    compile(name: 'admob-release-x.x.x.x', ext: 'aar')
    compile(name: 'msa_mdid_1.0.13', ext: 'aar')

    // 广点通平台需要的
    compile(name: 'gdt-release-x.x.x.x', ext: 'aar')
```

```

// 头条广告平台所需要的
compile(name: 'toutiao-release-x.x.x.x', ext: 'aar')
compile(name: 'open_ad_sdk-x.x.x.x', ext: 'aar')

// 百度广告平台所需要的
implementation(name: 'baidu-release-x.x.x.x', ext: 'aar')

// inmobi平台所需要的（还需要V4和RecyclerView）
// inmobi平台所需要picasso依赖库,不添加会导致广告无法请求
implementation(name: 'inmobi-release-x.x.x.x', ext: 'aar')
implementation 'com.squareup.picasso:picasso:2.5.2'

// mobvsita平台所需要的
implementation(name: 'mobvsita-release-x.x.x.x', ext: 'aar')
}

```

5.1.3 注意事项

- 如果项目中已经集成选择的广告平台的SDK，需要移除避免冲突；
- 需要将Android Support v4包升级到26.0.0或以上；

5.2 OAID支持

Android10之后IMEI等数据无法获取，这对广告投放将生一定影响，所以移动安全联盟(MSA)提出OAID来代替IMEI参与广告投放决策，OAID的支持会在一定程度上影响广告收益；

SDK中已经进行了OAID的封装，只需以下几步即可完成OAID的支持；

1. 导入安全联盟的OAID支持库 **msa_mdid_1.0.13.aar**（之前是miit_mdid_1.0.10.aar，强烈使用和广告SDK相对应的版本，因为OAID的SDK最近几个版本变化较大，不同版本之间存在兼容性问题）并依赖；
2. 将Demo中assets文件下的supplierconfig.json文件复制到自己的assets目录下并按照supplierconfig.json文件中的说明进行相关配置，**supplierconfig.json文件名不可修改**；目前supplierconfig.json文件中**仅需填写vivo应用市场的appid**，vivo 市场的Appid 具体位置如下：

[vivo开放平台](#) ——> 管理中心 ——> 应用于游戏 ——> 应用列表选择相对应的应用 ——> 选择App-ID



PS：需要更多帮助可参考目录下《移动智能终端补充设备标识体系统一调用SDK开发者说明文档》；

3. 添加以下混淆配置：

```
-keep class com.bun.miitmdid.core.** {*;}
```

PS：Inmobi平台目前仅兼容最新版本的OAID库（msa_mdid_1.0.13.aar），不再对旧版本OAID库兼容；

5.3 权限申请

使用SDK时可能需要以下权限，为了保证使用广告的正确，请在6.0及以上的手机中使用SDK前及时申请

```
<!-- 广告必须的权限，AAR中已经内置 -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
<uses-permission
android:name="android.permission.REQUEST_INSTALL_PACKAGES" />

<!-- 影响广告填充，强烈建议的权限 -->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<!-- 为了提高广告收益，建议设置的权限 -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
/>

<!-- 如果有视频相关的广告播放请务必添加-->
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

5.4 示例工程Demo导入说明

示例工程Demo请使用AndroidStudio来打开。

6. 接入代码

6.1 SDK初始化

在Application中进行SDK的初始化(详情请参考Demo)

```
private static final String APP_ID = "2482522";

private static final String[] PLATFORMS = {
    ADMobGenAdPlaforms.PLATFORM_ADMOB
    , ADMobGenAdPlaforms.PLATFORM_GDT
    , ADMobGenAdPlaforms.PLATFORM_TOUTIAO
    , ADMobGenAdPlaforms.PLATFORM_BAIDU
    , ADMobGenAdPlaforms.PLATFORM_INMOBI
    , ADMobGenAdPlaforms.PLATFORM_MOBVSITA
};

@Override
public void onCreate() {
    super.onCreate();
    ADMobGenSDK.instance().initSdk(this, new ADMobGenSdkConfig.Builder()
        // 修改为自己的appId
        .appId(APP_ID)
        // PLATFORMS只需要选择所需的广告平台(其中PLATFORM_ADMOB是必须的)
        .platforms(PLATFORMS)
        .build());
}
```

PS：AppId和需要接入哪些渠道的广告平台请联系对接的商务人员，初始化必须在主进程和主线程完成，SDK暂不支持多进程。

6.2 开屏广告示例

开屏广告加载使用方式如下，具体使用请参考Demo，开屏广告高度默认为屏幕高度的75%,可自定义高度比例,但不能低于0.75，可通过ADMobGenSDK.instance().getScreenHeight(activity)获取屏幕精确高度；

ADMobGenSplashView

- cn.admob.admobgensdk.ad.splash.ADMobGenSplashView

方法名	方法介绍
ADMobGenSplashView(Activity activity, double rate, int adIndex)	构造方法； activity：级别的上下文； rate：开屏广告高度占屏幕高度的比例 (0.75~1或-1， -1撑满广告容器)； adIndex：广告位序号
ADMobGenSplashView(Activity activity, double rate)	构造方法，广告位序号默认为0； activity：级别的上下文； rate：开屏广告高度占屏幕高度的比例 (0.75~1或-1， -1撑满广告容器)
ADMobGenSplashView(Activity activity)	构造方法，rate默认是0.75，广告位序号默认为0； activity：级别的上下文
setListener(ADMobGenSplashAdListener listener)	设置开屏广告监听； listener：开屏广告监听
setImmersive(boolean immersive)	设置是否适配沉浸式； immersive：是否适配沉浸式(true的话跳过按钮的上边距会加上状态栏高度)，不设置默认false
setTimeout(long timeout)	设置超时时间（总超时时长可能会略高于这个时间），不建议低于5000ms； timeout：超时时长，毫秒级别，不设置默认为5000ms
loadAd()	加载开屏广告
destroy()	释放开屏广告，建议在界面销毁的时候调用

ADMobGenSplashAdListener

开屏广告回调接口

- cn.admob.admobgensdk.ad.listener.ADMobGenSplashAdListener

方法名	方法介绍
onADReceiv()	开屏广告获取成功
onADFailed(String error)	开屏广告获取失败；error：错误信息
onADClick()	开屏广告被点击
onAdClose()	开屏广告关闭回调，不可单一在此回调进行跳转，跳转逻辑需要以下回调方法onADFailed、onADReceiv、onADClick、onAdClose和生命周期onResume、onPause配合实现，具体逻辑请参考Demo或示例
onADExposure	开屏广告曝光回调，但不一定是有效曝光，如网络问题导致曝光上报失败等

[开屏广告示例，rate为0.75~1](#)

[开屏广告示例，rate为-1](#)

6.3 Banner横幅广告示例

Banner广告为640:100，高度建议自适应，Banner广告建议放在固定位置的容器中，不建议放在ViewPager、RecyclerView、ListView或其他非固定位置的控件中。

ADMobGenBannerView

- cn.admob.admobgensdk.ad.banner.ADMobGenBannerView

方法名	方法介绍
ADMobGenBannerView(Activity activity, int adIndex)	构造方法； activity : activity级别的上下文； adIndex : 广告位序号
ADMobGenBannerView(Activity activity)	构造方法，广告位序号默认为0； activity : activity级别的上下文
setRefreshTime(int refreshTime)	设置刷新间隔（30s~120s，其中百度广告是30s刷新间隔且不可改变），不设置默认不刷新（不适用百度），头条Banner已经升级为模板广告，如果需要开启自动刷新，需要后台开启并设置刷新时间
setRemoveTouTiaoBannerInXiaoMiUp8(boolean removeTouTiaoBannerInXiaoMiUp8)	测试发现安卓版本在8.0及以上部分小米手机，当APP长时间处于后台时，Bugly有几率收集到WebView相关的Native Crash信息（暂时无法手动复现），目前测试发现有可能是头条的Banner广告导致的，后续将持续跟进此问题，设置为true（默认为false）将不在安卓8.0及以上的小米手机中获取头条的Banner广告
setGdt2(boolean gdt2)	设置广点通的Banner模式为2.0，目前默认为false，后续可能废弃并默认为true
setListener(ADMobGenBannerAdListener listener)	设置Banner广告监听； listener : Banner广告监听
loadAd()	加载Banner广告
destroy()	释放广告资源，建议在Banner广告被关闭或界面销毁时调用

ADMobGenBannerAdListener

Banner广告监听回调接口

- cn.admob.admobgensdk.ad.listener.ADMobGenBannerAdListener

方法名	方法介绍
onADReceiv()	Banner广告获取成功
onADFailed(String error)	Banner广告获取失败；error：错误信息
onADClick()	Banner广告被点击
onAdClose()	Banner广告关闭回调
onADExposure()	Banner广告曝光回调，但不一定是有效曝光，如网络问题导致曝光上报失败等

[Banner广告示例](#)

6.4 信息流原生模板广告示例

信息流原生模板广告，有丰富的广告样式（上图下文、上文下图、纯图、左图右文、左文右图、双图双文、三小图等），同时支持视频并且支持样式微调整

ADMobGenInformation

- `cn.admob.admobgensdk.ad.information.ADMobGenInformation`

方法名	方法介绍
ADMobGenInformation(Activity activity, int type, int adIndex)	构造方法； activity : activity级别的上下文； type : 信息流模板广告样式类型； adIndex : 广告位序号
ADMobGenInformation(Activity activity, int type)	构造方法，广告位序号默认为0； activity : activity级别的上下文； type : 信息流模板广告样式类型；
ADMobGenInformation(Activity activity)	构造方法，广告样式默认为 ADMobGenAdType.TYPE_INFORMATION， 广告位序号默认为0； activity : activity级别的上下文
setListener(SimpleADMobGenInformationAdListener listener)	设置信息流模板广告监听
getInformationAdStyle()	获取信息流模板广告样式，用于微调整信息流模板样式（广点通模板广告跳转需要给我们样式UI后台调整），具体调整规则请参考 QA文档
setShowClose(boolean showClose)	设置是否展示信息流模板广告的关闭按钮
setExposureDelay(long exposureDelay)	设置广告曝光校验最小间隔时间(0~200)，默认为200ms，在RecyclerView或ListView这种列表中不建议设置更小值，在一些特定场景（如Dialog或者固定位置可根据要求设置更小值）
setTTNativeExpressParam(TTNativeExpressParam ttNativeExpressParam)	设置头条平台的广告为模板广告（之前头条可创建自渲染广告由我们二次封装成样式统一的模板广告，2019年12月之后头条关闭了自渲染广告位的创建，均只能创建头条的模板广告，所以这个时间后新建的信息流广告位需要设置这个属性，之前创建的广告位可忽略，没有接入头条信息流模板广告可忽略； ttNativeExpressParam : 头条模板广告宽高参数）
loadAd()	加载信息流模板广告
destroy()	释放广告资源，建议在界面销毁时调用

InformationAdType

信息流模板广告样式类型

属性	属性介绍
NORMAL	上图下文，图片比例16:9
BOTTOM_IMAGE	上文下图，图片比例16:9
LEFT_IMAGE	左图右文，图片比例3:2
RIGHT_IMAGE	左文右图，图片比例3:2
VERTICALPICFLOW	竖图纯图，图片比例2:3
ONLY_IMAGE	横图纯图，图片比例16:9
THREE_PIC_FLOW	三小图, 图片不够时重复展示
THREE_PIC_FLOW_AUTO_FIT	三小图, 图片不够时自动展示几张
TWO_PIC_TWO_TEXT_FLOW	双图双文，图片比例16:9

SimpleADMobGenInformationAdListener

信息流模板广告监听回调

- cn.admob.admobgensdk.ad.listener.SimpleADMobGenInformationAdListener

方法	方法介绍
onADReceiv(IADMobGenInformation adMobGenInformation)	广告获取成功回调；adMobGenInformation：信息流模板广告对象
onADExposure(IADMobGenInformation adMobGenInformation)	广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败； adMobGenInformation：信息流模板广告对象
onADClick(IADMobGenInformation adMobGenInformation)	信息流模板广告被点击回调； adMobGenInformation：被点击的广告对象
onADFailed(String error)	信息流模板广告获取失败；error：错误信息
onADClose(IADMobGenInformation iadMobGenInformation)	广告关闭按钮被点击回调； iadMobGenInformation：被点击关闭按钮的广告对象
onADRenderFailed(IADMobGenInformation iadMobGenInformation)	信息流模板广告渲染失败； iadMobGenInformation：渲染失败的广告对象

IADMobGenInformation

信息流模板广告对象

- cn.admob.admobgensdk.ad.information.IADMobGenInformation

方法	方法介绍
getInformationAdView()	获取信息流模板广告视图
render()	开始渲染信息流模板广告
destroy()	释放信息流模板广告资源
isDestroy()	信息流模板广告是否已被释放，被释放的信息流模板广告不可在操作
isVideo()	判断当前信息流模板广告对象是否是视频广告
setADMobGenVideoListener(ADMobGenVideoListener videoListener)	设置视频监听，监听回调因各个平台会有所差异（有些回调方法可能不会执行）

TTNativeExpressParam

头条信息流模板广告参数

- cn.admob.admobgensdk.entity.TTNativeExpressParam

方法	方法介绍
TTNativeExpressParam(int width)	构造方法；width：模板广告期望的宽度，小于等于0将会取屏幕宽度
setWidth(int width)	设置模板广告期望的宽度；width：模板广告期望的宽度，小于等于0将会取屏幕宽度
setHeight(int height)	设置模板广告期望的高度；height：模板广告期望的高度，小于等于0将会自适应
setWithoutLowVersion(boolean withoutLowVersion)	目前测试发现在5.1及以下版本的手机中头条的模板广告有可能存在崩溃现象，设置这个为true将不再这些版本中加载头条信息流模板广告

信息流模板广告示例

PS：不建议一次获取多条广告，这样会导致广告资源的浪费（获取了很多广告但是有很大一部分没有被展示曝光，这些没有曝光的广告是没有收益的，建议可以像DEMO一样即取即用），一次获取多条广告也可能影响后续广告填充率

6.5 自渲染2.0广告示例

自渲染2.0能够返回标题，描述，图片链接等信息，目前支持**广点通**、**百度**、**Mobvsita**、**艾狄墨搏**平台，用于用户的自定义样式展示。在有视频的情况下，还会直接返回视频控件，用户只需把视频控件按规则添加到指定位置即可，无须关心视频的播放、缓存等问题。

ADMobGenNativeUnified

- cn.admob.admobgensdk.ad.nativead.ADMobGenNativeUnified

方法名	方法介绍
ADMobGenNativeUnified(Activity activity, int adIndex)	构造方法；activity：activity级别的上下文；adIndex：广告位序号；
ADMobGenNativeUnified(Activity activity)	构造方法，广告位序号默认为0；activity：activity级别的上下文
setListener(ADMobGenNativeUnifiedListener listener)	设置自渲染2.0广告监听
loadAd()	加载一条自渲染2.0广告
loadAd(int count)	加载count条(count大于0小于等于3)自渲染2.0广告，加载成功返回的广告数量小于等于count
resume()	恢复广告状态（有广点通平台必须调用该方法），建议在生命周期onResume中调用
destroy()	释放广告资源，建议在界面销毁时调用

ADMobGenNativeUnifiedListener

自渲染2.0广告监听回调

- cn.admob.admobgensdk.ad.listener.ADMobGenNativeUnifiedListener

方法	方法介绍
onADReceiv(List nativeUnifiedAdList)	广告获取成功回调；nativeUnifiedAdList：自渲染2.0广告集合
onADExposure(IADMobGenNativeUnifiedAd nativeUnifiedAd)	广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败；nativeUnifiedAd：自渲染2.0广告对象
onADClick(IADMobGenNativeUnifiedAd nativeUnifiedAd)	自渲染2.0广告被点击回调；nativeUnifiedAd：被点击的广告对象
onADFailed(String error)	自渲染2.0广告获取失败；error：错误信息

IADMobGenNativeUnifiedAd

自渲染2.0广告对象

- cn.admob.admobgensdk.entity.IADMobGenNativeUnifiedAd

方法	方法介绍
getTitle()	获取广告标题
getDescription()	获取广告描述
getPlatform()	获取广告来源平台
getIcon()	获取小图标
getImageList()	获取图片
getMediaView()	获取视频等媒体视图，如果这个视图不为空需要展示，具体的请参考Demo
isVideo()	判断是否是视频广告
registerVideoLister(ADMobGenNativeUnifiedVideoListener videoListener)	设置视频监听
registerViewForInteraction(ADMobGenNativeUnifiedContainer nativeUnifiedContainer)	设置承载广告的容器，一定要调用，影响收益； nativeUnifiedContainer：承载广告的容器，广告内容需要在该容器中，具体请参考Demo

[自渲染2.0广告示例](#)

6.5 插屏广告示例

插屏广告是移动广告的一种常见形式，在应用流程中弹出，当应用展示插屏广告时，用户可以选择点击广告，访问其目标网址，也可以将其关闭并返回应用，目前支持广点通、头条、Mobvsita、Inmobi、艾狄墨搏

ADMobGenInterstitial

- cn.admob.admobgensdk.ad.interstitial.ADMobGenInterstitial

方法	方法介绍
ADMobGenInterstitial(Activity activity, int adIndex)	构造函数； activity : activity级别的上下文； adIndex : 广告位序号
ADMobGenInterstitial(Activity activity)	构造方法，广告位序号默认为0； activity : activity级别的上下文
setListener(ADMobGenInterstitialAdListener listener)	设置插屏广告监听
loadAd()	加载插屏广告
destroy()	释放插屏广告资源，建议界面销毁的时候调用

ADMobGenInterstitialAdListener

插屏广告监听回调接口

- cn.admob.admobgensdk.ad.listener.ADMobGenInterstitialAdListener

方法	方法介绍
onADReceive(IADMobGenInterstitial iadMobGenInterstitial)	插屏广告获取成功回调，可在此回调之后展示插屏广告； iadMobGenInterstitial : 插屏广告对象
onADExposure(IADMobGenInterstitial iadMobGenInterstitial)	插屏广告广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败； iadMobGenInterstitial : 插屏广告对象
onADClick(IADMobGenInterstitial iadMobGenInterstitial)	插屏广告被点击回调； iadMobGenInterstitial : 插屏广告对象
onADClose(IADMobGenInterstitial iadMobGenInterstitial)	插屏广告被关闭回调； iadMobGenInterstitial : 插屏广告对象
onADFailed(String error)	插屏广告获取失败回调； error : 失败信息

IADMobGenInterstitial

插屏广告对象

- cn.admob.admobgensdk.entity.IADMobGenInterstitial

方法	方法介绍
hasShown()	返回插屏广告是否已经展示过，展示过的插屏不能再次展示
hasExpired()	获取插屏广告是否已经过期，过期的插屏广告不能展示
show(Activity activity)	展示插屏广告；activity : activity级别的上下文

[插屏广告示例](#)

6.7 激励视频广告示例

将短视频融入到APP场景当中，用户观看短视频广告后可以给予一些应用内奖励

ADMobGenRewardVod

- cn.admob.admobgensdk.ad.rewardvod.ADMobGenRewardVod

方法	方法介绍
ADMobGenRewardVod(Activity activity, int adIndex)	构造函数；activity : activity级别的上下文； adIndex : 广告位序号
ADMobGenRewardVod(Activity activity)	构造方法，广告位序号默认为0； activity : activity级别的上下文
setListener(SimpleADMobGenRewardVodAdListener listener)	设置激励视频广告监听
loadAd()	加载激励视频广告
destroy()	释放激励视频广告资源，建议界面销毁的时候调用

SimpleADMobGenRewardVodAdListener

激励视频广告监听回调接口

- cn.admob.admobgensdk.ad.listener.SimpleADMobGenRewardVodAdListener

方法	方法介绍
onADReceiv(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告获取成功回调，除Mobvsita平台外的激励视频可以在此回调之后展示激励视频； iadMobGenRewardVod : 激励视频广告对象
onVideoCached(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告视频缓存完毕回调，可以在此回调之后展示激励视频；iadMobGenRewardVod : 激励视频广告对象
onADExposure(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败； iadMobGenRewardVod : 激励视频广告对象
onADClick(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告被点击回调； iadMobGenRewardVod : 激励视频广告对象
onReward(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告奖励回调； iadMobGenRewardVod : 激励视频广告对象
onVideoComplete(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告视频播放完成回调； iadMobGenRewardVod : 激励视频广告对象
onADClose(IADMobGenRewardVod iadMobGenRewardVod)	激励视频广告被关闭回调； iadMobGenRewardVod : 激励视频广告对象
onADFailed(String error)	激励视频广告获取失败回调；error : 失败信息

IADMobGenRewardVod

激励视频广告对象

- cn.admob.admobgensdk.entity.IADMobGenRewardVod

方法	方法介绍
hasShown()	返回激励视频广告是否已经展示过，展示过的激励视频广告不能再次展示
hasExpired()	获取激励视频广告是否已经过期，过期的激励视频广告不能展示
showRewardVideo(Activity activity)	展示激励视频广告；activity : activity级别的上下文

[激励视频广告示例](#)

6.8 DrawVod广告示例

类似抖音小视频一样的沉浸式视频广告类型，目前仅支持头条

ADMobGenDrawVod

- cn.admob.admobgensdk.ad.drawvod.ADMobGenDrawVod

方法	方法介绍
ADMobGenDrawVod(Activity activity, int adIndex)	构造函数； activity : activity级别的上下文； adIndex : 广告位序号
ADMobGenDrawVod(Activity activity)	构造方法，广告位序号默认为0； activity : activity级别的上下文
setListener(ADMobGenDrawVodListener listener)	设置沉浸式视频广告监听
loadAd(int count)	加载count（1~3）条沉浸式视频广告
loadAd()	加载一条沉浸式视频广告
destroy()	释放沉浸式视频广告资源，建议界面销毁的时候调用

ADMobGenDrawVodListener

沉浸式视频广告监听回调接口

- cn.admob.admobgensdk.ad.listener.ADMobGenDrawVodListener

方法	方法介绍
onADReceiv(List drawVodList)	获取沉浸式视频广告成功回调； drawVodList : 沉浸式视频广告对象列表，列表长度小于等于load个数
onADExposure(IADMobGenDrawVod drawVod)	沉浸式视频广告广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败； drawVod : 沉浸式视频广告对象
onADClick(IADMobGenDrawVod drawVod)	沉浸式视频广告被点击回调； drawVod : 沉浸式视频广告对象
onADFailed(String error)	沉浸式视频广告获取失败回调； error : 失败信息

IADMobGenDrawVod

沉浸式视频广告对象

- cn.admob.admobgensdk.entity.IADMobGenDrawVod

方法	方法介绍
getView()	获取沉浸式视频广告视图，isOnly为true的话仅展示视频
getPlatform()	获取沉浸式视频广告来源平台
render()	渲染沉浸式视频广告
destroy()	释放资源
setADMobGenVideoListener(ADMobGenDrawVodVideoListener videoListener)	设置沉浸式视频播放监听；videoListener：沉浸式视频播放监听

[沉浸式视频广告示例](#)

6.9 全屏视频广告示例

ADMobGenFullScreenVod

- cn.admob.admobgensdk.ad.fullscreenvod.ADMobGenFullScreenVod

方法	方法介绍
ADMobGenFullScreenVod(Activity activity, int adIndex);	构造函数；activity：activity级别的上下文； adIndex：广告位序号
ADMobGenFullScreenVod(Activity activity)	构造方法，广告位序号默认为0； activity：activity级别的上下文
setListener(SimpleADMobGenFullScreenVodAdListener listener)	设置全屏视频广告监听
loadAd()	加载全屏视频广告
destroy()	释放全屏视频广告资源，建议界面销毁的时候调用

SimpleADMobGenFullScreenVodAdListener

全屏视频广告监听回调接口

- cn.admob.admobgensdk.ad.listener.SimpleADMobGenFullScreenVodAdListener

方法	方法介绍
onADReceiv(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告获取成功回调，可以在此回调之后展示全屏视频视频； iadMobGenFullScreenVod : 全屏视频广告对象
onVideoCached(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告视频缓存完毕回调，可以在此回调之后展示全屏视频； iadMobGenFullScreenVod : 全屏视频广告对象
onADExposure(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败； iadMobGenFullScreenVod : 全屏视频广告对象
onADClick(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告被点击回调； iadMobGenFullScreenVod : 全屏视频广告对象
onVideoComplete(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告播放完成回调； iadMobGenFullScreenVod : 全屏视频广告对象
onADClose(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告被关闭回调； iadMobGenFullScreenVod : 全屏视频广告对象
onSkipVideo(IADMobGenFullScreenVod iadMobGenFullScreenVod)	全屏视频广告跳过回调； iadMobGenFullScreenVod : 全屏视频广告对象
onADFailed(String error)	全屏视频广告获取失败回调； error : 失败信息

IADMobGenFullScreenVod

全屏视频广告对象

- cn.admob.admobgensdk.ad.fullscreenvod.IADMobGenFullScreenVod

方法	方法介绍
hasShown()	返回全屏视频广告是否已经展示过，展示过全屏视频广告不能再次展示
hasExpired()	获取全屏视频广告是否已经过期，过期的全屏视频广告不能展示
showFullScreenVod(Activity activity)	展示全屏视频广告； activity : activity级别的上下文

6.10 原生自渲染广告示例（已过时）

目前仅支持头条和Admobile，需要商务协商，详情请参考Demo

```
// adMobGenNative = new ADMobGenNative(this);
// 第二个参数是广告位序号（默认为0，用于支持单样式多广告位，无需要可以填0或者使用其他构造方法）
adMobGenNative = new ADMobGenNative(this, MyApplication.adIndex);
adMobGenNative.setListener(new ADMobGenNativeListener() {
    @Override
    public void onADReceiv(List<IADMobGenNativeAd> nativeAdList) {
        Log.e(TAG, "广告获取成功 ::::: ");
        finishLoad(nativeAdList);
    }

    @Override
    public void onADClick(IADMobGenNativeAd iadMobGenNativeAd) {
        Log.e(TAG, "广告被点击 ::::: ");
    }

    @Override
    public void onADExposure(IADMobGenNativeAd iadMobGenNativeAd) {
        Log.e(TAG, "广告展示曝光回调，但不一定是曝光成功了，比如一些网络问题导致上报失败 ::::: ");
    }

    @Override
    public void onADFailed(String error) {
        Log.e(TAG, "广告数据获取失败时回调 ::::: " + error);
        finishLoad(null);
    }
});
```

PS：具体的请参考Demo，onADClick和onADExposure只有IADMobGenNativeAd 调用 registerViewForInteraction之后才会回调（registerViewForInteraction一定需要调用，这将会影响收益）。

7. 适配问题

7.1 Android7.0适配

适配Anroid7.0以及以上，请在AndroidManifest中添加如下代码：

[如果您的支持库是AndroidX而非Support请点击跳转至Q&A文档并参考Q&A文档2.4进行适配](#)

```
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="${applicationId}.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths" />
</provider>
```

在res / xml目录下(如果xml目录不存在需要手动创建)，新建XML文件file_paths，在该文件中替换如下代码：

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <external-path name="external_path" path="." />
    <external-files-path name="external_files_path" path="." />
</paths>
```

PS：为了适配下载和安装相关功能，在工程中引用的包 com.android.support:support-v4:26.0.0请使用26.0.0及以上版本。

7.2 Android8.0、9.0适配

- 需要在 AndroidManifest.xml 增加权限声明
android.permission.REQUEST_INSTALL_PACKAGES，详情见前面添加权限声明部分。
- 需要在 AndroidManifest.xml 添加依赖声明 uses-library
android:name="org.apache.http.legacy" android:required="false"，且 application 标签中添加 android:usesCleartextTraffic="true"，适配网络http请求，否则 SDK 无法工作，接入代码示例如下：

```

<application
    android:name=".MyApplication"
    ...
    android:usesCleartextTraffic="true">

    <uses-library
        android:name="org.apache.http.legacy"
        android:required="false" />

    ...
</application>

```

8. 混淆配置

艾狄墨搏sdk内部已经混淆，因此在混淆之前添加以下配置

```

-ignorewarnings
# v4、v7
-keep class android.support.v4.**{public *;}
-keep class android.support.v7.**{public *;}

## 资源文件混淆配置
-keep class **.R$* { *; }
-keep public class **.R$*{
    public static final int *;
}
-keepclassmembers class **.R$* {
    public static <fields>;
}

# AdMobGenSdk混淆
-dontwarn cn.admob.admobgensdk.**
-keep class cn.admob.admobgensdk.**{*;}
-keep interface cn.admob.admobgensdk.**{*;}
-keep class com.android.**{*;}
-keep class com.ciba.**{ *; }
-keep interface com.ciba.**{ *; }
-dontwarn org.apache.commons.**
-keep class org.apache.**{*;}

# OAID混淆
-keep class com.bun.miitmdid.core.** {*;}

# AdMob广告SDK混淆
-keep class admsdk.library.**{*;}

```

```

# 广点通广告SDK混淆
-keep class com.qq.e.** {public protected *;}
-keep class MTT.ThirdAppInfoNew {*;}
-keep class com.tencent.** {*;}

# 百度广告SDK混淆
-keepclassmembers class * extends android.app.Activity {public void *
(android.view.View);}
-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}
-keep class com.baidu.mobads.**{*;}

## 头条广告混淆
-keep class com.bytedance.sdk.openadsdk.** { *; }
-keep class com.androidquery.callback.** {*;}
-keep public interface com.bytedance.sdk.openadsdk.downloadnew.** {*;}
-keep class com.ss.sys.ces.* {*;}

## imobi广告SDK混淆
-keepattributes SourceFile,LineNumberTable
-keep class com.inmobi.** { *; }
-dontwarn com.inmobi.**
-keep public class com.google.android.gms.**
-dontwarn com.google.android.gms.**
-dontwarn com.squareup.picasso.**
-keep class com.google.android.gms.ads.identifier.AdvertisingIdClient{public
*;}
-keep class
com.google.android.gms.ads.identifier.AdvertisingIdClient$Info{public *;}
-keep class com.squareup.picasso.** {*;}
-dontwarn com.squareup.picasso.**
-dontwarn com.squareup.okhttp.**
-keep class com.moat.** {*;}
-dontwarn com.moat.**
-keep class com.integralads.avid.library.** {*;}

## mobvsita广告SDK混淆
-keepattributes Signature
-keepattributes *Annotation*
-keep class com.mintegral.** {*; }
-keep interface com.mintegral.** {*; }
-dontwarn com.mintegral.**
-keep class com.alphab.** {*; }
-keep interface com.alphab.** {*; }

```

PS：如果混淆了资源文件，请注意白名单（WhiteList.txt）内的资源不能混淆

9. 常见问题和说明

9.1 下载弹框样式修改

头条、Inmobi、Mobvsita和Admob的下载弹框默认样式，可通过重写这个style修改样式

```
<style name="Theme.Dialog.TTDownload"
parent="@android:style/Theme.DeviceDefault.Light.Dialog">
    <item name="android:windowMinWidthMajor">65%</item>
    <item name="android:windowMinWidthMinor">95%</item>
</style>
```

9.2 SDK版本号相关信息

2.3.8及之后版本SDK可以通过AD_VERSION的log来查看集成的各个平台SDK的版本号

9.3 Q&A和错误信息对照表

[Q&A和错误信息对照表\(点击此链接跳转\)](#)

10. 商务合作

tomato@admobile.top