# 概述

口袋工厂广告 **SDK**

#### 尊敬的开发者朋友，欢迎使用口袋工厂广告 SDK。本文档旨在帮助 Android 应用开发者在程序中快速植入口袋工厂平台提供的广告。您只需要进行简单的配置，就可以快速接入本平台提供的广告。源于 SDK 具体的使用方法，请仔细阅读下面的文档。

1. 支持广告类型

|  |  |  |
| --- | --- | --- |
| 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， 提供 5s 的可感知广告展示。 | app 启动时，常会使用开屏广告。 |
| 插屏广告 | 插屏广告是移动广告的一种常见形式，在应用开流程中弹出，当应用展示插页式广告时，用户可以选择点按广告，访问其目标网址，也可以将其关闭，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。 |
| Banner 广告 | 位于 app 顶部、中部、底部任意一处， 横向贯穿整个 app 页面 | 任意界面的固定位置。 |
| 信息流广告 | 信息流广告是有（上文下图，左图右文，三小图）等多种样式的模板广告 | 信息流列表，轮播控件， 固定位置都比较适合。 |
| 激励视频广告 | 将短视频融入到 app 场景当中，成为app“任务”之一。用户观看短视频广告后可以得到一些应用内奖励 | 常出现在游戏的复活、登录等位置，或者网服类app 的一些增值服务场景。 |
| 全屏视频广告 | 插屏使用场景都可以 | 在应用执行流程的自然停顿点，适合投放这类广  告。 |

1. **SDK** 集成
   1. 导入实例工厂

#### 如果您是初次使用口袋工厂广告 SDK，我们建议您利用 Demo 工厂来了解广告 SDK 的使用规范。使用编译工具 AndroidStudio 通过 import project 的方式导入实例工厂，再弹出的先导中按照 AndroidStudio 的默认选择和提示，一直按 next 即可导入工程。

* 1. 集成广告 **SDK**

#### 将 pocket\_ad\_sdk\_1.8.0.aar、csj\_ad\_sdk\_0622.aar、ks\_ad\_sdk\_0608.aar、ks\_ad\_sdk\_0608.aar、miit\_mdid\_1.0.10.aar 的 arr 包拷贝到 libs 目录下，并且再build.gradle 中进行相应的配置。

android {

....

//1. 添加依赖库⽬录

repositories {

maven { url 'https://jitpack.io' } flatDir {

dirs 'libs'

}

}

}

dependencies {

....

//2. 添加⼴告依赖包

implementation(name: 'csj\_ad\_sdk\_211014', ext: 'aar')

implementation (name: 'gdt\_ad\_sdk\_210928', ext:'aar') implementation (name: 'ks\_ad\_sdk\_211014', ext:'aar')

implementation(name: 'pocket\_ad\_sdk\_2.0.0', ext:'aar') implementation 'com.squareup.okhttp3:okhttp:4.9.2'

implementation 'com.google.code.gson:gson:2.8.8'

implementation 'androidx.legacy:legacy-support-core-ui:1.0.0'

}

* 1. 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：

<!--必要权限-->

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<uses-permission android:name="android.permission.GET\_TASKS"/>

<!--可选权限-->

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

<uses-permission android:name="android.permission.REQUEST\_INSTALL\_PACKAGES"/>

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<!--demo 场景用到的权限，不是必须的-->

<uses-permission android:name="android.permission.RECEIVE\_USER\_PRESENT" />

<uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />

<uses-permission android:name="android.permission.EXPAND\_STATUS\_BAR" />

<!--快手 SDK 内自定义的权限，与下载相关，aar 中已经申请，不用可以不加-->

<permission

android:name="${applicationId}.permission.KW\_SDK\_BROADCAST" android:protectionLevel="signature" />

<uses-permission android:name="${applicationId}.permission.KW\_SDK\_BROADCAST" />

<application>

<activity

android:name="com.qq.e.ads.PortraitADActivity"

android:screenOrientation="portrait"

android:configChanges="keyboard|keyboardHidden|orientation|screenSize"

tools:replace="android:screenOrientation"/>

<activity

android:name="com.qq.e.ads.LandscapeADActivity"

android:screenOrientation="landscape"

android:configChanges="keyboard|keyboardHidden|orientation|screenSize"

tools:replace="android:screenOrientation"/>

<provider

android:name="com.qq.e.comm.GDTFileProvider"

android:authorities="${applicationId}.gdt.fileprovider"

android:exported="false"

android:grantUriPermissions="true">

<meta-data

android:name="android.support.FILE\_PROVIDER\_PATHS"

android:resource="@xml/gdt\_file\_path" />

</provider>

<provider

android:name="com.bytedance.sdk.openadsdk.TTFileProvider"

android:authorities="${applicationId}.TTFileProvider"

android:exported="false"

android:grantUriPermissions="true">

<meta-data

android:name="android.support.FILE\_PROVIDER\_PATHS"

android:resource="@xml/csj\_file\_paths" />

</provider>

<provider

android:name="com.bytedance.sdk.openadsdk.multipro.TTMultiProvider"

android:authorities="${applicationId}.TTMultiProvider"

android:exported="false" />

</application>

## 混淆配置

-keep class com.zh.pocket.\*\*{\*;}

# 接入代码

* 1. **SDK** 初始化

#### 为保证 SDK 功能的正常使用，必须在 Application 中继续 SDK 的初始化(详情参考 Demo)

接入代码：

##### @Override

##### public void onCreate() {

super.onCreate();

// 参数 1 为 channel：标识渠道（可随意填写），不要出现特殊字符即可；参数 2 为

APPID(在开发者后台获取，也可联系客服获取)

PocketSdk.initSDK(context, "xiaomi", "1");

}

* 1. 开屏广告示例

(具体使⽤请参考 Demo，可⾃定义⾼度⽐例,但不能低于 0.75。如果从后台回到app 需要回到启动页展示广告，开发者不需要实现，已经在 sdk 完成。在计时完成或者跳过时调用 ActivityFrontBackProcessor.toFront(getIntent()) 判断是否后台到前台，true：是、false：否。详见 demo 工程的 SplashADActivity)

接入代码：

SplashAD splashAD = new SplashAD(Activity, “广告位id”);

splashAD.show(ViewGroup);

**API**：

com.zh.pocket.ads.splash.SplashAD

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| SplashAD(Activity activity, String codeId) | 构造函数，广告相关回调通过 SplashADListener 通知开发者。参数：Activity(展示广告的 Activity)、广告位id |
| setSplashADListener(SplashADListener listener) | 设置广告监听 |
| show(ViewGroup container) | 调用展示广告。参数：ViewGroup(广告容器) |
| destroy() | 释放资源 |

com.zh.pocket.ads.splash.SplashADListener

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| onFailed(ADError error) | 广告加载失败，error 对象包含了错误码和错误信息 |

|  |  |
| --- | --- |
| onADExposure() | 广告曝光时调用 |
| onADClicked() | 广告被点击时调用 |
| onADDismissed() | 广告关闭时调用，用户关闭或者展示时间到 |
| onADTick(long millisUntilFinished) | 广告展示倒计时回调，参数单位为毫秒 |

* 1. 插屏广告示例

#### 接入代码：

##### 在回调函数 onAdLoader 展示广告

InterstitialAD interstitialAD = new InterstitialAD(Activity, “广告位 id”);

interstitialAD.setInterstitialADListener(new InterstitialADListener()); interstitialAD.show();

**API**：

### com.zh.pocket.ads.interstitial.InterstitialAD

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| InterstitialAD(Activity activity, String id) | 构造函数，广告相关回调通过InterstitialADListener 通知开发者。参数：Activity(展示广告的 Activity)、id(广告位 id) |
| setInterstitialADListener(InterstitialADListener listener) | 设置广告监听 |
| Load() | 加载广告 |
| showAD() | 展示广告 |

|  |  |
| --- | --- |
| close() | 广告广告 |
| destroy() | 释放资源 |

com.zh.pocket.ads.interstitial.InterstitialADListener

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| onFailed(ADError error) | 广告加载失败，error 对象包含了错误码和错误信息 |
| onADExposure() | 广告曝光时调用 |
| onADClicked() | 广告被点击时调用 |
| onADClosed() | 广告关闭时调用 |
| onSuccess() | 广告加载成功 |
| onADLoaded() | 广告加载完毕时回调 |

* 1. **banner** 广告示例

#### 接入代码：

##### 在回调函数 onAdLoader 展示广告

BannerAD bannerAD = new BannerAD(Activity, “广告位 id”);

bannerAD.setBannerADListener(new BannerADListener()); bannerAD.loadAD(ViewGroup);

**API**：

### com.zh.pocket.ads.banner.BannerAD

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| BannerAD(Activity activity, string id) | 构造函数，广告相关回调通过BannerADListener 通知开发者。参数： Activity(展示广告的 Activity)、id(广告位id) |
| loadAD(ViewGroup container) | 调用加载展示广告。container(广告容器) |
| setBannerADListener(BannerADListener banner) | 设置监听 |
| destroy() | 释放资源 |

com.zh.pocket.ads.banner.BannerADListener

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| onFailed(ADError error) | 广告加载失败，error 对象包含了错误码和错误信息 |
| onADExposure() | 广告曝光时调用 |
| onADClicked() | 广告被点击时调用 |
| onADClosed() | 广告关闭时调用 |
| onSuccess() | 广告加载成功 |

* 1. 信息流广告示例

#### 接入代码：

##### 在回调函数 onAdLoader 展示广告

NativeAD nativeAD = new NativeAD(Activity, ViewGroup, “广告位 id”);

nativeAD.setNativeADListener(new NativeADListener()); nativeAD.loadAD();

**API**：

### com.zh.pocket.ads.nativ.NativeAD

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| NativeAD(Activity activity, ViewGroup container, String id) | 构造函数，广告相关回调通过BannerADListener 通知开发者。参数：Activity(展示广告的 Activity)、ViewGourp(广告容器)、id(广告位 id) |
| setInterstitialADListener(InterstitialADListener listener) | 设置广告监听 |
| LoadAD() | 加载广告 |
| showAD() | 调用加载展示广告 |
| destroy() | 释放资源 |

com.zh.pocket.ads.nativ.NativeADListener

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| onFailed(ADError error) | 广告加载失败，error 对象包含了错误码和错误信息 |
| onADExposure() | 广告曝光时调用 |

|  |  |
| --- | --- |
| onADClicked() | 广告被点击时调用 |
| onADClosed() | 广告关闭时调用 |
| onADLoaded() | 广告加载成功 |

#### 信息流广告展示：

* 1. 激励视频广告示例

接入代码：

##### 在回调函数 onAdLoader 展示广告

RewardVideoAD rewardVideoAD = new RewardVideoAD(Activity, “广告位 id”);

rewardVideoAD.setRewardVideoADListener(new RewardVideoADListener()) rewardVideoAD.loadAD();

**API**：

### com.zh.pocket.ads.video.RewardVideoAD

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| RewardVideoAD(Activity activity, String id) | 构造函数，广告相关回调通过RewardVideoADListener 通知开发者。参数： Activity(展示广告的 Activity)、String(广告位 id) |
| loadAD() | 调用加载广告 |
| showAD() | 展示广告 |
| destroy() | 释放资源 |

com.zh.pocket.ads.video.RewardVideoADListener

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| onFailed(ADError error) | 广告加载失败，error 对象包含了错误码和错误信息 |
| onSuccess() | 广告加载成功 |
| onADLoad() | 广告加载成功，可在此回调后进行广告展示，此时广告过期时间确定，可通过RewardVideoAD.getExpireTimestamp()获取 |
| onVideoCached() | 视频素材缓存成功，可在此回调后进行广告展示 |
| onADShow() | 激励视频广告页面展示，此后  RewardVideoAD.hasShown()返回 true |
| onADExposure() | 广告曝光时调用 |
| onReward() | 广告激励发放 |
| onADClicked() | 广告被点击时调用 |
| onADClosed() | 广告关闭时调用 |
| onVideoComplete() | 广告视频素材播放完毕 |
| onSkippedVideo() | 广告被跳过 |
| onADLoaded() | 广告预加载成功 |

* 1. 全屏视频广告示例

#### 接入代码：

##### 在回调函数 onAdLoader 展示广告

FullscreenVideoAD fullscreenVideoAD=new FullscreenVideoAD(Activity, “广告位 id”);

fullscreenVideoAD.setFullscreenVideoADListener(new FullscreenVideoADListener()) fullscreenVideoAD.loadAD();

**API**：

### com.zh.pocket.ads.video.FullscreenVideoAD

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| FullscreenVideoAD(Activity activity, String id) | 构造函数，广告相关回调通过FullscreenVideoADListener 通知开发者。参数： Activity(展示广告的 Activity)、id(广告位 id) |
| loadAD() | 加载广告 |
| showAD() | 展示广告 |
| destroy() | 释放资源 |

com.zh.pocket.ads.video.FullscreenVideoADListener

|  |  |
| --- | --- |
| 方法名 | 方法介绍 |
| onFailed(ADError error) | 广告加载失败，error 对象包含了错误码和错误信息 |
| onSuccess() | 广告加载成功 |
| onADLoaded() | 广告加载成功，可在此回调后进行广告展示，此时广告过期时间确定，可通过  RewardVideoAD.getExpireTimestamp()获取 |

|  |  |
| --- | --- |
|  |  |
| onVideoCached() | 视频素材缓存成功，可在此回调后进行广告展示 |
| onADShow() | 激励视频广告页面展示，此后  RewardVideoAD.hasShown()返回 true |
| onADExposure() | 广告曝光时调用 |
| onADClicked() | 广告被点击时调用 |
| onADClosed() | 广告关闭时调用 |
| onVideoComplete() | 广告视频素材播放完毕 |
| onSkippedVideo() | 广告被跳过 |
| onPreload() | 广告预加载加载成功 |