

# DU Ad Platform\_SDK for Android 接入手册（触发式广告）

---

DUAd\_SDK\_Trigger v1.0

百度在线网络技术（北京）有限公司

# 目 录

- 1. 获取身份 .....1
- 2. 加载与配置.....2
  - 2.1 加载 SDK 文件 .....2
  - 2.2 配置 AndroidManifest.xml.....2
  - 2.3 混淆代码 .....3
- 3.初始化 .....4
- 4. 功能使用 .....4

前提: WeatherWizard SDK 需要依赖 DU Ad Platform\_SDK HW/CW 版本。  
在接入 WeatherWizard 之前需要完成 HW/CW 初始化, 加载, 代码混淆三个部分。

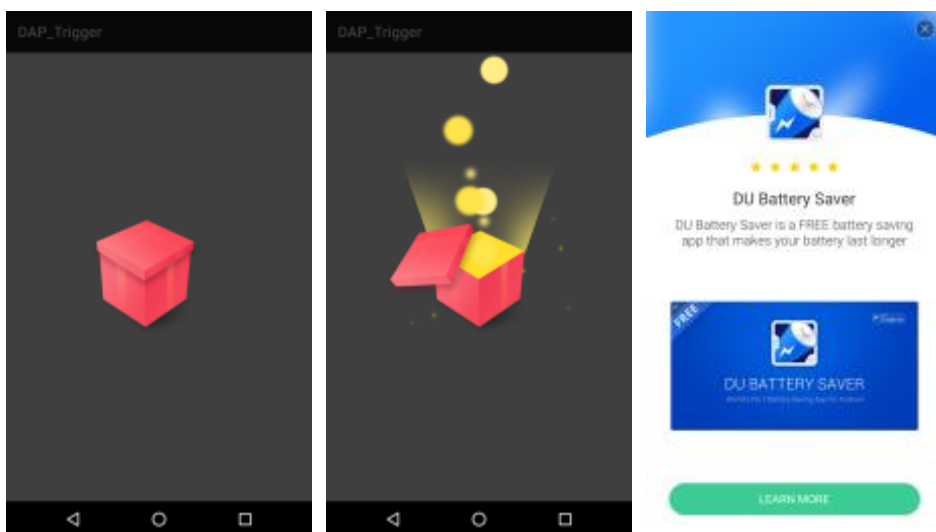
触发式广告可以在产品的界面寻找合适的广告入口添加触发 icon, icon 有动画交互, 点击后能够弹出广告内容。

广告图标样式:



图标动画: 显示后震动

点击后:



## 1. 获取身份

请参照 HW/CW 版 DUADplatform SDK 文档 第 3 章;  
申请 Trigger 广告位 ID 时, 需申请的广告类型为「触发式」。

创建广告位

\* 广告位名称:

请输入广告位名称

\* 广告触发步骤:

\* 广告形式:

☐原生

☐插屏

☐广告墙

☐横幅

☐视频

☒触发式

☐来电提醒

☐天气通

☐拇指快划

☐ 第三方平台广告位ID配置 (当选择的广告来源为第三方平台时填写)

保存

## 2. 加载与配置

### 2.1 加载 SDK 文件

- 1) 将 DuappsAd-Trigger-xxx.aar 加入工程 libs 文件夹下:
- 2) 配置工程 build.gradle:

```
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile(name: 'DuappsAd-xW-xxx-release', ext: 'aar')
    // TriggerSDK 必须依赖 DAP 广告 SDK
    compile(name: 'DAPSDK_Weather-release-xxx', ext: 'aar')
}
```

\*注: flatDir 指定的位置 即为 aar 存放的位置

### 2.2 配置 AndroidManifest.xml

- A. 添加强权。WeatherSDK 使用的最低权限如下：

```
<!-- 定位权限 -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<!-- 悬浮框权限 -->
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
```

- B. 在 app\_license 的 value 中填入已申请的 APP\_ID。

```
<application
    android:name="com.mobula.sample.MobulaApplication"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/mobulaTheme" >
    <meta-data
        android:name="app_license"
        android:value="xxxxxxxxxx" />
    <provider
        android:name="com.duapps.ad.stats.DuAdCacheProvider"
        android:authorities="packagename.DuAdCacheProvider"
        android:exported="false">
    </provider>
```

\*注：“packagename”为开发者 APP 的包名全称。

## C. 注册 APP 安装广播监听

```
<receiver android:name="com.duapps.ad.base.PackageAddReceiver" >
<intent-filter>
    <action android:name="android.intent.action.PACKAGE_ADDED" />
    <data android:scheme="package" />
</intent-filter>
</receiver>
```

## D. 添加 Trigger Activity

```
<activity
    android:name="com.trigger.ad.DuTriggerADActivity"
    android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
    android:screenOrientation="portrait"
    android:excludeFromRecents="true"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
```

## 2.3 混淆代码

请参照 CW 版 DUADplatform SDK 文档 4.3 章节 混淆代码。

## A 把 DU Ad Platform\_SDK 中的类排除在混淆之外；

```
-keep class com.duapps.ad.**{*}
```

## B 将以下类添加到 proguard 配置：

```
-keep class com.dianxinos.DXStatService.stat.TokenManager {
    public static java.lang.String getToken(android.content.Context);
}
-keep public class * extends android.content.BroadcastReceiver
-keep public class * extends android.content.ContentProvider
-keepnames @com.google.android.gms.common.annotation.KeepName class *
-keepclassmembernames class * {
    @com.google.android.gms.common.annotation.KeepName *;
}
-keep class com.google.android.gms.common.GooglePlayServicesUtil {
    public <methods>;
}
-keep class com.google.android.gms.ads.identifier.AdvertisingIdClient {
    public <methods>;
}
-keep class com.google.android.gms.ads.identifier.AdvertisingIdClient$Info {
    public <methods>;
}
```

\*注：混淆方法参见 **Android 官方混淆文档**：[\\${ android-sdk }/tools/proguard/](#)

### 3.初始化

请参照 CW 版 DUADplatform SDK 文档 第 5 章初始化。

- 方法:

在 application 的 OnCreate 方法中使用 *DuAdNetwork.init()*

**\*注：请按上述要求使用该接口.否则可能导致初始化无效。**

开发者可以在 application 的 OnCreate 方法中使用 *DuAdNetwork.setLaunchChannel()*;用以区分不同 app 投放渠道的数据。此接口可选择使用，不是必须。

如需接入 Trigger 广告，请在 json 中添加如下标签，并传入相应广告位 ID。

```
{
  "native": [
    {
      "pid": "YOUR_DAP_PLACEMENT_ID"
    }
  ]
}
```

注：用来初始化 *DuAdNetwork.init()*，json 数组部分具体参见 HW & CW 版本初始化章节。

- 接口说明：

**public static void** *init*(Context context, String pidsJson);

参数	说明
<b>Context context</b>	ACTIVITY CONTEXT
<b>String pidsJson</b>	Placement_ID 与广告位 ID 的对应关系

- 接口说明：

**public static void** *setLaunchChannel* (String channelName);

参数	说明
<b>String channelName</b>	此接口可以帮你根据你的 APP 投放渠道，区分数据。此接口可选择使用，不是必须。

### 4. 功能使用

在需要展示触发式广告入口的页面 xml 添加广告入口自定义 view，并做相关自定义属性配置。

自定义属性包括：

(1) Trigger:Pid

触发式广告位的 pid。该值配置在 json 文件的“**native**”数组中。

**该属性必须填写。**

## (2) Trigger:AutoStart

入口图标动画是否自动播放，如不配置，默认 true(自动播放)；

## (3) Trigger:Delay

配置动画播放延时时长，单位为 ms，不配置时缺省为 500ms；

注：该属性只有在配置 Trigger:AutoStart= "true"时才能生效。

代码示例：

```
<com.trigger.view.TriggerIconView
xmlns:Trigger="http://schemas.android.com/apk/res-auto"
    android:id="@+id/trigger_icon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    Trigger:Pid="YOUR_PID"
    Trigger:AutoStart="true"
    Trigger:Delay="500"/>
```

**注意：**如果用户配置 Trigger:AutoStart="false"，即入口动画不自动播放，则需在使用时主动调用入口动画播放接口

### ● 接口说明

public void startAnim(int delay)

参数	说明
int delay	动画播放延时时长，单位为 ms

### ● 代码示例：

```
TriggerIconView icon = (TriggerIconView)
                        findViewById(R.id.trigger_icon);
icon.startAnim(500);
```