

腾讯社交广告转化追踪 Android SDK 接入说明文档 V1.2

文档

腾讯社交广告转化追踪 Android SDK 接入说明文档 V1.2

必读

1. SDK嵌入

1.1 步骤1：添加SDK到工程中

1.2 步骤2: 修改AndroidManifest.xml文件

2. 接入代码

2.1 激活事件统计接口

2.1.1 接入代码示例

2.1.2 激活事件主要API

2.2 更多转化类型统计接口

2.2.1 接入代码示例

2.2.2 注册事件主要API

2.3 会话（启动次数）统计接口

2.3.1 接入代码示例

2.3.2 会话（启动次数）统计主要API

2.3.3 会话（启动次数）统计接口说明

3.注意事项

必读

由于部分广告主包名存在规范问题，例如大小写、Android和iOS共用包名等原因，在接入前，请联系广点通运营确认包名和APP ID关联是否存在异常。

联系方式：

QQ：2508973061 白璿

RTX：v_junbai，franciszhao

1. SDK嵌入

1.1 步骤1：添加SDK到工程中

请在工程根目录下创建一个名为 `libs` 的子目录，并将SDK的jar包 `GDTTrackSDK.min.1.1.0.jar` 拷贝到该目录下。

(1) 如果您使用的是Eclipse + ADT，而且ADT版本比较低的话，则需要在Eclipse中选中刚才的jar包，然后右键选择 `Build Path->Add to Build Path`（ADT17以上已经不需要手动导入）。

(2) 如果您使用的是Android Studio，需要在gradle中添加：

```
dependencies {
    compile files('libs/GDTTrackSDK.min.1.0.0.jar')
}
```

1.2 步骤2: 修改AndroidManifest.xml文件

添加权限声明：

```
<uses-permission android:name="android.permission.INTERNET" /> <!-- 允许联网 -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/> <!--
检测联网方式，区分设备当前网络是2G、3G、4G还是WiFi -->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" /> <!-- 获取
MAC地址，和设备标识一起作为用户标识 -->
<uses-permission android:name="android.permission.READ_PHONE_STATE"/> <!-- 获取设
备标识，标识用户 -->
```

目前支持的SDK版本，最低到API 9（即Android系统版本2.3）：

```
<uses-sdk android:minSdkVersion="9" android:targetSdkVersion="22" />
```

2. 接入代码

2.1 激活事件统计接口

2.1.1 接入代码示例

```
1. (完整的代码示例请参考demo工程中的GDTTrackerApplication.java)
2.
```

```

3. public class GDTTrackerApplication extends Application {
4.
5.     @Override
6.     public void onCreate() {
7.         super.onCreate();
8.
9.         /**
10.          * SDK初始化接口：必须在Application的onCreate中成功调用一次，否则其他接口都将无法
            使用。
11.          * 注意：只需要调用一次即可，第一次调用之后的任何调用都将无效。Logcat中会有初始化信
            息输出，请开发者注意。
12.          *
13.          * @param context    applicationContext
14.          * @param channel      渠道标识，注意：目前仅支持传入"ANDROID"、"UNIONANDROID"这
            两个字符串。传入其他字符串将无法被统计。
15.          */
16.
17.         GDTTracker.init(this, "ANDROID");
18.
19.         /**
20.          * SDK激活上报接口：Application创建的时候是激活接口调用的最佳时机，调用此接口会向
            广点通上报激活事件。
21.          * 注意：30天内App被打开只会被记录为一次激活。
22.          *
23.          * @param context    applicationContext
24.          */
25.         GDTTracker.activateApp(this);
26.     }
27. }

```

2.1.2 激活事件主要API

com.qq.e.track.Tracker

方法名	功能
init(Context context, String channel)	初始化方法，必须在程序的入口调用一次，该方法不会阻塞程序也不会影响程序性能。 注意：如果调用该方法时，channel参数必须是TrackConstants.APP_CHANNEL类中定义好的渠道。否则SDK将无法成功初始化，开发者也无法使用SDK提供的其他功能。
activateApp(Context context)	激活上报接口，context参数是程序的上下文，一般是ApplicationContext
activateApp(Context context, int value)	激活上报接口，context参数含义同上，value参数是激活事件的价值，由开发者根据自己需要来定义，应该是大于0的整数；如果是金额，则以分为单位，例如1元对应的参数应该是100

2.2 更多转化类型统计接口

2.2.1 接入代码示例

```
1.      (完整的代码示例请参考demo工程中的GDTTrackerActivity.java)
2.      // 当用户在App产生需要某一个事件，我们希望统计这个事件的发生次数时，可以调用logEvent
   接口进行上报。
3.      buttonRegister.setOnClickListener(new View.OnClickListener() {
4.          @Override
5.          public void onClick(View v) {
6.              // 对应的事件类型
7.              String conversionType = null;
8.              switch (v.getId()) {
9.                  case R.id.button_reg:
10.                     conversionType = TrackConstants.CONVERSION_TYPE.REGISTER;
11.                     break;
12.                  case R.id.button_add_to_cart:
13.                     conversionType = TrackConstants.CONVERSION_TYPE.ADD_TO_CART;
14.                     break;
15.                  case R.id.button_purchase:
16.                     conversionType = TrackConstants.CONVERSION_TYPE.PURCHASE;
17.                     break;
18.                  case R.id.button_score:
19.                     conversionType = TrackConstants.CONVERSION_TYPE.SCORE;
20.                     break;
21.                  case R.id.button_search:
22.                     conversionType = TrackConstants.CONVERSION_TYPE.SEARCH;
23.                     break;
24.                  default:
25.                     break;
26.              }
27.
28.              if (TextUtils.isEmpty(conversionType)) {
29.                  return;
30.              }
31.              /**
32.               * SDK事件统计接口：当用户发生"注册"、"添加到购物车"、"付费"、"评分"、
               "搜索"行为时，可以向广点通上报对应的事件。
33.               *
34.               * @param context      applicationContext
35.               * @param eventName      事件类型，注意：目前只能统计
               到"TrackConstants.CONVERSION_TYPE"类中预定义过的事件类型，如果开发者传入其他字符串，
```

将无法在统计到这个事件。

```
36.         */
37.         GDTTracker.logEvent(context, conversionType);
38.     }
39. });
```

2.2.2 注册事件主要API

com.qq.e.track.Tracker

方法名	功能
logEvent(Context context, String eventName)	注册事件上报接口，context参数是程序的上下文，一般是ApplicationContext，eventName是事件类型。注意：目前SDK只支持注册事件的上报，即开发者只能给eventName传入值为"MOBILEAPP_REGISTER"的字符串
logEvent(Context context, String eventName, int value)	注册事件上报接口，context参数和eventName含义同上，value参数是激活事件的价值，由开发者根据自己需要定义，一般是大于0的整数。

2.3 会话（启动次数）统计接口

2.3.1 接入代码示例

```
1.     /**
2.      * 会话统计接口onSessionStart和onSessionEnd注意事项：
3.      *
4.      * 1.确保在所有的Activity中都成对地调用"GDTTracker.onSessionStart(this)"和"GDTTracker.onSessionEnd(this)"方法。
5.      * 2.如果Activity之间有继承或者控制关系请不要同时在父和子Activity中重复添加这两个方法，否则会造成重复统计。
6.      * 3.SDK默认当应用在后台运行超过30秒再回到前台时，是两个独立的session(启动次增加1)。例如按下home键，经过一段时间（大于30s）后再返回应用，启动次数会增加1次。
7.      */
8.     @Override
9.     protected void onResume() {
10.         super.onResume();
11.         Log.i(TAG, "onResume");
12.
13.     /**
14.      * 会话开始接口
15.      *
16.      * @param context 应用上下文，本身是Activity的话就传入Activity
17.      */
```

```

18.         GDTTracker.onSessionStart(this);
19.     }
20.
21.     @Override
22.     protected void onPause() {
23.         super.onPause();
24.         Log.i(TAG, "onPause");
25.
26.         /**
27.          * 会话结束接口
28.          *
29.          * @param context    应用上下文，本身是Activity的话就传入Activity
30.          */
31.         GDTTracker.onSessionEnd(this);
32.     }

```

2.3.2 会话（启动次数）统计主要API

`com.qq.e.track.Tracker`

方法名	功能
<code>onSessionStart(Context context)</code>	会话开始接口，context参数是上下文信息，如果context本身是一个Activity就传入Activity。
<code>onSessionEnd(Context context)</code>	会话结束接口，参数含义同上。注意：1. 会话结束接口一定要和会话开始接口成对地在Activity中调用才能正确统计启动次数；2. 如果Activity之间有继承或者控制关系请不要同时在父和子Activity中重复添加这两个方法，避免重复统计。

2.3.3 会话（启动次数）统计接口说明

1. 为了准确统计应用的会话，需要在每一个在每个Activity的onResume方法中调用onSessionStart，在onPause方法中调用onSessionEnd。当一个onSessionStart方法与上一个onSessionEnd方法时间相差30s以上时，标志着一个新的session开始。
2. 如果Activity之间有继承或者控制关系请不要同时在父和子Activity中重复添加这两个方法，否则会造成重复统计。

3.注意事项

代码混淆：如果需要使用proguard混淆代码，请注意不要混淆SDK的代码。在proguard配置文件中添加如下配置：

```
1.  -keep class com.qq.e.** {  
2.      *;  
3.  }
```

SDK初始化：必须先调用GDTTracker.init(Context context, String channel)接口，成功初始化SDK后，才能调用激活、注册事件和会话统计等其他的接口。init方法中的channel参数目前只能是ANDROID和UNIONANDROID这两个字符串中的一个，开发者如果传入了错误的channel参数，SDK将抛出异常，无法成功初始化，也不会进行任何工作。

SDK下载链接：

http://imgcache.qq.com/qzone/biz/gdt/dev/sdk/android/release/GDT_Android_Track_SDK.zip