

推啊广告 **Android SDK**—接入说明文档

目 录

目 录	2
版本记录	3
修订记录	4
1 SDK 嵌入	5
2 接入代码	7
3 注意事项	31

版本记录

版本号	版本支持	版本说明	升级建议
2.2.0	Android 4.0 及以上	此版本对原有版本进行了代码和性能的优化，解决了之前版本存在的问题。	建议升级

修订记录

修改日期	修改人	修改内容简述	版本号
2017 年 04 月 20 日	咸红伟	SDK 的打包	2.1.0
2017 年 04 月 20 日	咸红伟	更改广告字段，避免应用市场审核 SDK 支持 gif，吸引眼球，增加点击率	2.1.0
2017 年 05 月 25 日	黄文奇	修改 2.10 节 ms_item_id/material_id/adslot_id 字段 由 int 改为 long 类型； activity_id 由原来的 int 改为 string 类型	2.1.0
2017 年 06 月 26 日	咸红伟	SDK 的打包	2.2.0
2017 年 06 月 26 日	咸红伟	根据线上日志优化问题	2.2.0
2017 年 07 月 06 日	罗泽宇	修改 2.10 节 request_id 字段由 int 改为 string	2.2.0

1 SDK 嵌入

1.1. 步骤 1：添加 SDK 到工程中

如果当前工程中无 **libs** 目录，请在工程文件根目录中创建一个名为 **libs** 的子目录，将推啊 SDK 的 **aar** 包拷贝到 **libs** 目录下。

在 **studio** 项目 **module** 的 **build.gradle** 进行如下配置：

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

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile(name: 'tuia-2.2.0', ext: 'aar')
}
```

1.2. 步骤 2：修改 **AndroidManifest.xml** 文件

添加权限声明：

```
<!-- 必选权限 -->
<uses-permission android:name="android.permission.MANAGE_USERS" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission
    android:name="android.permission.PACKAGE_USAGE_STATS"
    tools:ignore="ProtectedPermissions" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.BATTERY_STATS"
/>
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.WRITE_SETTINGS"
/>
<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_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.GET_TASKS" />
    <!-- 可选权限，不声明此部分权限将放弃部分设备信息的采集，对数据分析及设备指纹的
精度有一定影响 -->
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_UPDATES" />
    <uses-permission
android:name="android.permission.CHANGE_NETWORK_STATE" />
    <uses-permission
android:name="android.permission.CHANGE_WIFI_STATE" />

```

在 **Application** 下增加 **meta-data**，填入在平台上申请到的 **app_key**，该 **app_key** 需要预先在推啊媒体平台 <https://ssp.tuia.cn> 注册申请（如已申请，请忽略本提示）。填写在 **meta-data** 的 **value** 值中。

```

<!-- 填写平台上申请到的 appkey -->
<meta-data
    android:name="TUIA_APPKEY"
    android:value="28MLSNMkp88LYjjRbqSkzy2jxULg"/>

```

接入任意广告（**Banner**、横幅、应用墙、插屏、信息流、浮标、开屏、自定义广告），都需要在推啊媒体后台申请相应广告位置的唯一标识广告位 **id** 即 **adslot_id**，在具体接入代码中填充，例见下 2.2 节。注意：为了更精准的数据统计以及分成结算，一个位置对应一个广告位 **id**。如有对以上内容有理解上的偏差，请咨询推啊媒体运营对接人员

2 接入代码

2.1. 初始化

在自定义的 **Application** 的 **onCreate** 方法中，调用以下方法：（详细内容请参考压缩包中的代码示例）。

```
TaSDK.init(this);
```

2.2. 横幅广告接入代码（详细内容请参考 **demo** 中的示例代码）

```
<com.db.ta.sdk.TMBrTmView
    android:id="@+id/TMBrView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:TMBR_size="lander_TMBr"/>
```

```
public class SbannerActivity extends BaseActivity {

    private TMBrTmView mTMBrAdView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sbanner);
        mTMBrAdView = (TMBrTmView) findViewById(R.id.TMBrView);
        mTMBrAdView.setAdListener(new TmListener() {
            @Override
            public void onReceiveAd() {
                Log.d("=====", "onReceiveAd");
            }

            @Override
            public void onFailedToReceiveAd() {
                Log.d("=====", "onFailedToReceiveAd");
            }

            @Override
            public void onLoadFailed() {
                Log.d("=====", "onLoadFailed");
            }

            @Override
            public void onCloseClick() {
```

```

        Log.d("=====", "onCloseClick");
    }

    @Override
    public void onAdClick() {
        Log.d("=====", "onClick");
    }

    @Override
    public void onAdExposure() {
        Log.d("=====", "onAdExposure");
    }
});
mTMBrAdView.loadAd(458); //加载对应广告 id
}

@Override
protected void onDestroy() {
    if (mTMBrAdView != null) {
        mTMBrAdView.destroy();
    }
    super.onDestroy();
}
}

```


2.3. Banner 广告接入代码（详细内容请参考 demo 中的示例代码）

```
<com.db.ta.sdk.TMBrTmView
    android:id="@+id/TMBrView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:TMBr_size="TMBr"/>
```

```
public class BannerActivity extends BaseActivity {

    private TMBrTmView mTMBrAdView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_banner);
        mTMBrAdView = (TMBrTmView) findViewById(R.id.TMBrView);
        mTMBrAdView.setAdListener(new TmListener() {

            @Override
            public void onReceiveAd() {
                Log.d("=====", "onReceiveAd");
            }

            @Override
            public void onFailedToReceiveAd() {
                Log.d("=====", "onFailedToReceiveAd");
            }

            @Override
            public void onLoadFailed() {
                Log.d("=====", "onLoadFailed");
            }

            @Override
            public void onCloseClick() {
                Log.d("=====", "onCloseClick");
            }

            @Override
            public void onAdClick() {
                Log.d("=====", "onAdClick");
            }
        });
    }
}
```

```

        @Override
        public void onAdExposure() {
            Log.d("=====", "onExposure");
        }
    });

    //      mTMBrAdView.loadAd(43);
    mTMBrAdView.loadAd(462);
    //      mTMBrAdView.loadAd(343);
    }

    @Override
    protected void onDestroy() {
        if (mTMBrAdView != null) {
            mTMBrAdView.destroy();
        }
        super.onDestroy();
    }
}

```

2.4. 插屏广告嵌入代码

```
private TMitTm mTMitAd;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_interstitial);
    mTMitAd = new TMitTm(this);
    RefWatcher refWatcher = SdkApplication.getRefWatcher(this);
    refWatcher.watch(this);
    mTMitAd.setAdListener(new TmListener() {
        @Override
        public void onReceiveAd() {
            Log.d("=====", "onReceiveAd");
        }

        @Override
        public void onFailedToReceiveAd() {
            Log.d("=====", "onFailedToReceiveAd");
        }

        @Override
        public void onLoadFailed() {
            Log.d("=====", "onLoadFailed");
        }

        @Override
        public void onCloseClick() {
            Log.d("=====", "onCloseClick");
        }

        @Override
        public void onAdClick() {
            Log.d("=====", "onAdClick");
        }

        @Override
        public void onAdExposure() {
            Log.d("=====", "onAdExposure");
        }
    });
}

// mTMitAd.loadAd(73);
```

```
        mTMItAd.loadAd(459);  
//        mTMItAd.loadAd(658);  
    }  
    @Override  
    protected void onDestroy() {  
        if (mTMItAd != null) {  
            mTMItAd.destroy();  
        }  
        super.onDestroy();  
    }  
}
```

2.5. 信息流广告嵌入代码

信息流广告的展现形式更加多样，以 **HTML** 代码形式返回，需做如下 **schema** 配置

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <com.db.ta.sdk.TMNaTmView
        android:id="@+id/TMNaView"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        tools:TMNa_size="TMNa_750_180"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@color/background"/>

    <com.db.ta.sdk.TMNaTmView
        android:id="@+id/TMNaView1"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        tools:TMNa_size="TMNa_750_420"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@color/background"/>

</LinearLayout>
```

```
public class NativeAdActivity extends BaseActivity {

    private TMNaTmView mTMNaAdView;
```

```

private TMNaTmView mTMNaAdView1;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_native_ad);
    RefWatcher refWatcher = SdkApplication.getRefWatcher(this);
    refWatcher.watch(this);
    mTMNaAdView = (TMNaTmView) findViewById(R.id.TMNaView);
    mTMNaAdView1 = (TMNaTmView) findViewById(R.id.TMNaView1);
    mTMNaAdView.setAdListener(new TmListener() {
        @Override
        public void onReceiveAd() {
            Log.d("=====", "onReceiveAd");
        }

        @Override
        public void onFailedToReceiveAd() {
            Log.d("=====", "onFailedToReceiveAd");
        }

        @Override
        public void onLoadFailed() {
            Log.d("=====", "onLoadFailed");
        }

        @Override
        public void onCloseClick() {
            Log.d("=====", "onCloseClick");
        }

        @Override
        public void onAdClick() {
            Log.d("=====", "onAdClick");
        }

        @Override
        public void onAdExposure() {
            Log.d("=====", "onAdExposure");
        }
    });
    // mTMNaAdView.loadAd(53);
    // mTMNaAdView1.loadAd(52);
    mTMNaAdView.loadAd(461);

```

```
        mTMNaAdView1.loadAd(460);  
//        mTMNaAdView.loadAd(160);  
//        mTMNaAdView1.loadAd(59);  
    }  
  
    @Override  
    protected void onDestroy() {  
        if (mTMNaAdView != null) {  
            mTMNaAdView.destroy();  
        }  
        if (mTMNaAdView1 != null) {  
            mTMNaAdView1.destroy();  
        }  
        super.onDestroy();  
    }  
}
```

2.6. 应用墙广告嵌入代码说明

应用墙广告需要注意的是，即使在同一页面的多个位置，也需要申请多个广告位，否则会出现多个位置返回同一个广告的情况

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="horizontal"
    >

    <com.db.ta.sdk.TMAwView
        android:id="@+id/app1"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:scaleType="centerCrop"
        android:src="@drawable/btn_photo"
        tools:shape="circular"
        />

    <ImageView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:scaleType="centerCrop"
        android:src="@drawable/btn_photo"/>

    <com.db.ta.sdk.TMAwView
        android:id="@+id/app2"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:scaleType="centerCrop"
        android:src="@drawable/tm_btn_photograph"
        tools:shape="square"
        />

</LinearLayout>
```

```
public class AppWallActivity extends BaseActivity {
```



```

private TMAwView mTMAwView1;
private TMAwView mTMAwView2;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_app_wall);
    RefWatcher refWatcher = SdkApplication.getRefWatcher(this);
    refWatcher.watch(this);
    mTMAwView1 = (TMAwView) findViewById(R.id.app1);
    mTMAwView2 = (TMAwView) findViewById(R.id.app2);

    mTMAwView1.setAdListener(new TmListener() {
        @Override
        public void onReceiveAd() {
            Log.d("=====", "onReceiveAd");
        }

        @Override
        public void onFailedToReceiveAd() {
            Log.d("=====", "onFailedToReceiveAd");
        }

        @Override
        public void onLoadFailed() {
            Log.d("=====", "onLoadFailed");
        }

        @Override
        public void onCloseClick() {
            Log.d("=====", "onCloseClick");
        }

        @Override
        public void onAdClick() {
            Log.d("=====", "onAdClick");
        }

        @Override
        public void onAdExposure() {
            Log.d("=====", "onAdExposure");
        }
    });
}

```

```

        mTMaView2.setAdListener(new TmListener() {
            @Override
            public void onReceiveAd() {
                Log.d("=====", "onReceiveAd");
            }

            @Override
            public void onFailedToReceiveAd() {
                Log.d("=====", "onFailedToReceiveAd");
            }

            @Override
            public void onLoadFailed() {
                Log.d("=====", "onLoadFailed");
            }

            @Override
            public void onCloseClick() {
                Log.d("=====", "onCloseClick");
            }

            @Override
            public void onAdClick() {
                Log.d("=====", "onAdClick");
            }

            @Override
            public void onAdExposure() {
                Log.d("=====", "onAdExposure");
            }
        });
//        mTMaView1.loadAd(45);
//        mTMaView2.loadAd(45);
//
//        mTMaView1.loadAd(464);
//        mTMaView2.loadAd(464);
//
//        mTMaView1.loadAd(341);
//        mTMaView2.loadAd(342);
    }

    @Override
    protected void onDestroy() {

```

```
        if (mTMAwView1 != null) {  
            mTMAwView1.destroy();  
        }  
        if (mTMAwView2 != null) {  
            mTMAwView2.destroy();  
        }  
        super.onDestroy();  
    }  
}
```

2.7. 浮标广告嵌入代码说明

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <ListView
        android:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"></ListView>

    <com.db.ta.sdk.TMAwView
        android:id="@+id/TMAw1"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical|right"
        tools:shape="square"/>

</FrameLayout>
```

```
public class DobberActivity extends BaseActivity {

    private TMAwView mTMAwView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dobber);
        RefWatcher refWatcher = SdkApplication.getRefWatcher(this);
        refWatcher.watch(this);
        mListView = (ListView) findViewById(R.id.list);
        mTMAwView = (TMAwView) findViewById(R.id.TMAw1);
        mListView.setAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_expandable_list_item_1, getData()));
        mTMAwView.setAdListener(new TmListener() {
            @Override
            public void onReceiveAd() {
                Log.d("=====", "onReceiveAd");
            }
        });
    }
}
```

```

    }

    @Override
    public void onFailedToReceiveAd() {
        Log.d("=====", "onFailedToReceiveAd");
    }

    @Override
    public void onLoadFailed() {
        Log.d("=====", "onLoadFailed");
    }

    @Override
    public void onCloseClick() {
        Log.d("=====", "onCloseClick");
    }

    @Override
    public void onAdClick() {
        Log.d("=====", "onAdClick");
    }

    @Override
    public void onAdExposure() {
        Log.d("=====", "onAdExposure");
    }
});
//      mTMAwView.loadAd(44);
mTMAwView.loadAd(463);
//      mTMAwView.loadAd(1329);
//      mTMAwView.loadAd(1152);
}

@Override
protected void onDestroy() {
    if (mTMAwView != null) {
        mTMAwView.destroy();
    }
    super.onDestroy();
}

private List<String> getData() {

    List<String> data = new ArrayList<String>();

```

```
    for (int i = 0; i < 40; i++) {  
        data.add("测试数据" + i);  
    }  
    return data;  
}  
}
```

2.8. 开屏广告嵌入代码说明（详见 demo 代码）

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="@drawable/default_image_background">

    <com.db.ta.sdk.TMShTmView
        android:id="@+id/TMSh_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />

</RelativeLayout>
```

```
public class SplashActivity extends Activity {

    private TMShTmView mTMShTmView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        RefWatcher refWatcher =
SdkApplication.getRefWatcher(getApplicationContext());
        refWatcher.watch(this);
        if (Build.VERSION.SDK_INT >= 23) {
            int checkCallPhonePermission =
ContextCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE);
            if (checkCallPhonePermission !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_PHONE_STATE}, 0);
            }
        }
    }
}
```

```

mTMShTmView = (TMShTmView)findViewById(R.id.TMSH_container);
mTMShTmView.setTargetClass(this,MainActivity.class);
mTMShTmView.setAdListener(new TMSHAdListener(){
    @Override
    public void onTimeout() {
        Log.d("TMSHActivity", "onTimeout");
    }

    @Override
    public void onReceiveAd() {
        Log.d("TMSHActivity", "onReceiveAd");
    }

    @Override
    public void onFailedToReceiveAd() {
        Log.d("TMSHActivity", "onFailedToReceiveAd");
    }

    @Override
    public void onLoadFailed() {
        Log.d("TMSHActivity", "onLoadFailed");
    }

    @Override
    public void onCloseClick() {
        Log.d("TMSHActivity", "onCloseClick");
    }

    @Override
    public void onAdClick() {
        Log.d("TMSHActivity", "onClick");
    }

    @Override
    public void onAdExposure() {
        Log.d("TMSHActivity", "onAdExposure");
    }
});
// mTMSHAdView.loadAd(84);
mTMSHAdView.loadAd(466);

}

```



```
@Override
protected void onDestroy() {
    if (mTMShTmView != null) {
        mTMShTmView.destroy();
    }
    super.onDestroy();
}
}
```

2.9. 开屏广告传递参数说明（详见 demo 代码）

如果开发者要在开屏的 **activity** 里面向另一个 **activity** 传递数据，可以传入自定义的 **intent**，代码如下：

```
public class SplashActivity extends Activity {

    private TMShTmView mTMShTmView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        if (Build.VERSION.SDK_INT >= 23) {
            int checkCallPhonePermission =
ContextCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE);
            if (checkCallPhonePermission !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_PHONE_STATE}, 0);
            }
        }
        mTMShTmView = (TMShTmView) findViewById(R.id.TMSH_container);
        Intent intent = new Intent(this, MainActivity.class);
        intent.putExtra("name", "name");
        intent.putExtra("age", 10);
        mTMShTmView.setTargetClass(this, MainActivity.class, intent);
        mTMShTmView.setAdListener(new TMShTmListener() {
            @Override
            public void onTimeOut() {
                Log.d("TMSHActivity", "onTimeOut");
            }

            @Override
            public void onReceiveAd() {
                Log.d("TMSHActivity", "onReceiveAd");
            }

            @Override
            public void onFailedToReceiveAd() {
                Log.d("TMSHActivity", "onFailedToReceiveAd");
            }

            @Override
            public void onLoadFailed() {
```

```

        Log.d("TMSHActivity", "onLoadFailed");
    }

    @Override
    public void onCloseClick() {
        Log.d("TMSHActivity", "onCloseClick");
    }

    @Override
    public void onAdClick() {
        Log.d("TMSHActivity", "onClick");
    }

    @Override
    public void onAdExposure() {
        Log.d("TMSHActivity", "onAdExposure");
    }
});
//      mTMSHAdView.loadAd(84);
      mTMSHTmView.loadAd(466);
    }

    @Override
    protected void onDestroy() {
        if (mSplashTmView != null) {
            mSplashTmView.destroy();
        }
        super.onDestroy();
    }
}

```

2.10. 非标准广告嵌入代码说明

非标准广告即自定义广告，前面 7 种广告皆为标准广告，所有界面展示和渲染由推啊 sdk 自身负责，开发者只需相应调用即可。但是非标准广告需要为推啊 sdk 提供接口，返回 json 字符串数据，由开发者自身负责数据的渲染与广告的展示，以及广告曝光/点击行为的主动上报。

非标广告的接口返回参数请见下文：

参数	参数类型	参数定义	参数说明
request_id	string	广告请求 id	对应的请求 id，该参数在请求时 sdk 自动生成
error_code	string	错误码	
adslot_id	long	广告位 id	
ad_title	string	推广标题	
activity_title	string	广告活动标题	
activity_id	string	广告活动 id	
ad_icon	string	“广告”小图标	
ad_icon_visible	boolean	是否展示 ad_icon	
ad_close	string	“关闭”小图标	
ad_close_visible	boolean	是否展示 ad_close	
description	string	广告描述	
img_url	string	广告活动素材地址	
img_width	int	广告活动素材宽度	
img_height	int	广告活动素材高度	
ad_content	string	广告活动内容	仅当 ad_type=2 时有内容
ad_type	int	广告位类型	0:插屏 1:横幅 2:信息流 3:banner 4:浮标 5:应用墙 6:开屏 7- 非标准广告
expire	long	有效期	默认为 0
wdata_token	string		
server_time	string		
click_url	string	素材 url	
material_id	long	素材 id	

dcm	string		
material_list	List<MaterialRsp>	返回的素材列表	用语多个素材需求
activity_way	int		
material_way	int		
data1	string	扩展字段 1	
data2	string	扩展字段 2	
rid	string	唯一请求 id	
source	int	活动来源	
app_id	long	应用 id	
device_id	String	设备 id	

MaterialRsp

ms_item_id	long		
item_type	int		
image_url	string	素材 url	
image_width	int		
image_height	int		
text	string	素材文案描述	

注：以上**红色加粗**字体需要特别注意

获取非标准广告数据，给 **NonStandardAd** 对象设置 **NsAdListener** 回调接口，**sdk** 获取到数据后会将原始数据通过 **onReceiveAd(String result)** 方法返回给开发者。

```
public class NonStandarActivity extends BaseActivity {
    private NonStandardTm mNonStandardTm;

    private TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_non_standar);

        textView = (TextView)findViewById(R.id.content_text);
        mNonStandardTm = new NonStandardTm(this);
        mNonStandardTm.loadAd(465);
        mNonStandardTm.setAdListener(new NsTmListener() {
            @Override
            public void onReceiveAd(String result) {
                textView.setText(result);
            }
        });
    }
}
```

```

        @Override
        public void onFailedToReceiveAd() {

        }

    });

    textView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mNonStandardTm.adExposed();
            mNonStandardTm.adClicked();
        }
    });
}

@Override
protected void onDestroy() {
    if (mNonStandardTm != null) {
        mNonStandardTm.destroy();
    }
    super.onDestroy();
}
}

```

非标准广告的点击和曝光事件是通过 **NonStandardAd** 对象的 **adExposed()**和 **adClicked()** 两个接口进行上报。

需要注意，adExposed()为曝光，需要在曝光的时候调用，adClicked()为点击，需要在点击的时候调用

3 注意事项

3.1. 代码混淆

如果您需要使用 **proguard** 混淆代码，需确保不要混淆 **SDK** 的代码。

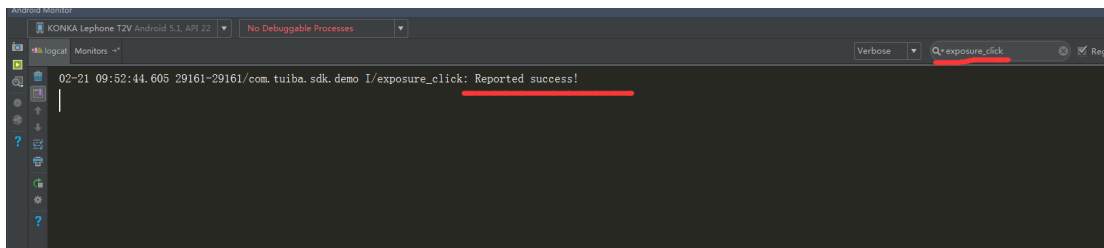
请在 **proguard-rules.pro** 文件(或其他混淆文件)尾部添加如下配置：

```
-keep class com.db.ta.sdk.** { *; }
#TONGDUN
-dontwarn android.os.**
-dontwarn com.android.internal.**
-keep class cn.tongdun.android.**{*;}

```

3.2. 关于曝光与点击上报是否成功的查看

2.0.1 版本之前的 **SDK** 没有曝光与点击的回调接口，若要查看是否上报成功，可通过控制台的 **log** 日志查看，输入 **exposure_click**：



3.3. 关于广告位 id

以上示例代码中，**loadAd** 方法里面填的广告位 **id** 仅为示例，开发者需要在推啊媒体平台重新申请。

3.4. 关于自定义广告的点击和曝光

自定义广告中，**adExposed()**为曝光方法，需要在广告位展示的时候调用，**adClicked()**为点击方法，需要在用户点击的时候调用。

3.5. 关于权限

广告位在请求的时候 **sdk** 需要获取 **device_id**，用来保证广告请求的唯一性，为此，如果使用的 **API** 版本在 **6.0** 以上的时候，需要动态获取 **READ_PHONE_STATE** 权限

```
if (Build.VERSION.SDK_INT >= 23) {
    int checkCallPhonePermission =
ContextCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE);
    if (checkCallPhonePermission !=
PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_PHONE_STATE}, 0);
    }
}
```