DU Ad Platform_SDK for Cocos
接入手册

DU Cocos SDK v1.0

百度在线网络技术（北京）有限公司

目 录

1. 获取身份

本章描述 DU Ad Platform_SDK 接入过程中需要的三个 ID：APP_ID, DAP 广告位 ID, Facebook 广告位 ID。

1.1 APP_ID

A. 定义
APP_ID 是开发者的应用在广告平台的唯一标识。

B. 获取方式
访问百度开发者平台 http://ad.duapps.com 进行申请。

C. 代码
app_license

1.2 DAP 广告位 ID

A. 定义
广告位 ID 是开发者平台上广告所在的广告位置的标识。开发者可以创建多个广告位。

B. 获取方式
访问百度开发者平台 http://ad.duapps.com 进行申请。

C. 代码
pid

2. 加载与配置

本章描述在 Cocos 环境下如何加载 DU Ad Platform_SDK SDK，如何配置 AndroidManifest.xml，以及根据项目需要配置混淆代码。

请严格按照本章进行配置，否则有可能会出现运行异常。

2.1 加载 SDK 文件

A. 将 DuAd_Cocos_SDK_1.0/Classes 文件夹下所有文件放在工程 Classes 文件夹下
B. 将 DuAd_Cocos_SDK_1.0/cpp 文件夹下所文件放在工程 app/src/org/cocos2dx/cpp 文件夹下
C. 将 DuAd_Cocos_SDK_1.0/libs/android 文件夹下 aar 包及 jar 包放在工程 app/libs 目录下
D. 修改 Android 项目的 build.gradle

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile(name: 'DuappsAd-CW-xxx-release', ext: 'aar')
    compile(name: 'DuVideoSdk-xxx-release', ext: 'aar')
    compile(name: 'support-v4-23.1.0', ext: 'aar')
    compile 'com.android.support:palette-v7:23.4.0'
}
```

注：

1. DuappsAd-CW-vXX-release.aar 为原生广告和插屏广告 SDK，DuVideoSdk-vXX-release.aar 为视频广告 SDK，可以自行替换官网最新版本
2. 'com.android.support:palette-v7:23.4.0'为视频 SDK 专用，具体版本号请根据项目情况更改，建议配置为 21 以上。如果编译报错，请更改为：

```
compile 'com.android.support:palette-v7:23.4.0' {
transitive = false
}
```

2.2 配置 AndroidManifest.xml

请按照如下步骤修改 **AndroidManifest.xml 文件**

A. 添加权限。DU Ad Platform_SDK 使用的最低权限如下：

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

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。

请正确添加该监听，否则会影响您的变现效率。

```
<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>
```

2.3 混淆代码

请务必按如下混淆规则添加到 proguard 配置，对应用代码进行混淆,否则有可能会出现运行异常:

A 把 DU Ad Platform_SDK 中的类排除在混淆之外；

```
-dontwarn com.duapps.ad.**
-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.app.Activity
-keep public class * extends android.app.Application
-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. SDK 初始化

在完成 DU Ad Platform_SDK 接入操作之前，安卓应用首先需要对 DU Ad Platform_SDK 做初始化。**没有进行初始化的广告位 id 无法拉取广告。**

3.1 配置 Json

请按照如下步骤修改 ***assets/dxtoolbox*** 文件夹下的 **dxtoolbox.json 文件**

- 原生和插屏广告：请将 pid 填入 native 部分
- 视频广告：请将 pid 填入 video 部分

```
{
    "native": [
        {
            "pid": "YOUR_DAP_PLACEMENT_ID(for interstitial ads)"
        }
    ],
    "video": [
      {
            "pid": "YOUR_DAP_PLACEMENT_ID(for video)"
       }
    ]
}
```

3.2 初始化

请在 AppActivity 的 OnCreate 方法中使用初始化方法。

//初始化 SDK

DuAdNetwork.init(Cocos2dxActivity.getContext(), getConfigJSON(this));

DuAdNetwork.setLaunchChannel("cocos2dx");

//初始化视频 SDK（不使用视频广告可省去该方法）

DuVideoAdSDK.init(Cocos2dxActivity.getContext(), getConfigJSON(this));

- **代码示例：**

```
private static String TOOLBOX_AD_CONFIG = "dxtoolbox/dxtoolbox.json";

private String getConfigJSON(Context context) {
    BufferedInputStream bis = null;
    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    try {
        bis = new BufferedInputStream(context.getAssets().open(TOOLBOX_AD_CONFIG));
```

```
            byte[] buffer = new byte[4096];
            int readLen = -1;
            while ((readLen = bis.read(buffer)) > 0) {
                bos.write(buffer, 0, readLen);
            }
        } catch (IOException e) {
            Log.e("", "IOException :" + e.getMessage());
        } finally {
            closeQuietly(bis);
        }
        return bos.toString();
}
```

4. 插屏广告使用

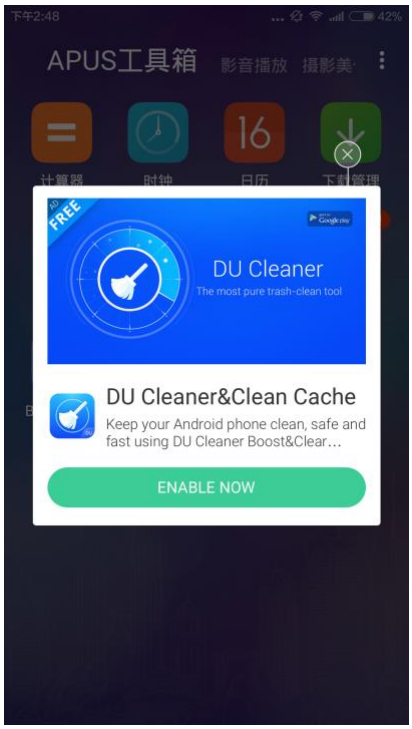

图 2： 半屏插屏广告样式

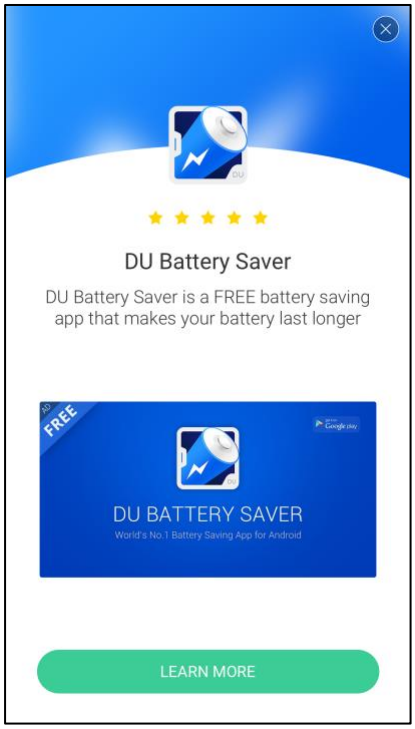

图 3： 全屏插屏广告样式

4.1 构造方法

- 接口说明：
  public DAPInterstitialAd(int placementID, int type);

参数	说明
int type	*DAPInterstitialAd:: TYPE_FULL_SCREEN*: 全屏广告 *DAPInterstitialAd::TYPE_NORMAL*: 半屏广告

	此参数缺省时，默认为半屏
int pid	广告位 ID，该 pid 注册在 json 的 native 数组中

4.2 插屏广告回调

请先注册接收广告数据的回调，然后获取广告数据接口。

广告数据获取成功或失败，点击事件的响应是通过回调接口返回的。此过程与广告数据获取过程是异步的，不会阻塞开发者的线程。

- **接口说明：**

  public void onAdReceive(InterstitialAdBridgeCallback adDidLoad);

  获取广告成功，开发者可以进行展示

  public void onAdPresent(InterstitialAdBridgeCallback adDidShow);

  广告展示回调，通知开发者该插屏广告被展示

  public void onAdClicked(InterstitialAdBridgeCallback adDidClick);

  广告点击回调，通知开发者该插屏广告被点击

  public void onAdDismissed(InterstitialAdBridgeCallback adDidClose);

  广告关闭回调，通知开发者该插屏广告被关闭

  public void onAdError(InterstitialAdBridgeErrorCallback adFailWithError);

  获取广告失败，开发者可以通过 int error 获取错误码。获取广告数据失败的错误码及含义，参见 4.3

4.3 广告接口

4.3.1 预加载

开发者可根据自己产品的需求，选择时机使用填充广告缓存接口。

- 调用 fill()接口可以提前缓存广告，在 load()广告时可以更快展示。
  建议在广告展示的前置场景调用 fill()。

  ***注：**广告数据会缓存到客户端内存中，**不会缓存广告的图片数据，只会缓存图片的 Url 地址，缓存数据量小。**

- **接口说明：**

  public void fillAd();

4.3.2 加载

- **接口说明：**

  public void loadAd();

  *注:请先设置插屏广告监听器,再加载广告.

4.3.3 展示

- **接口说明：**
  public void showAd();

5. 视频广告使用

图 4:视频广告播放界面

图 5:视频广告结果页

5.1 构造方法

- **接口说明：**
  public DAPVideoAd(int placementID);

5.2 视频广告回调

请按如下方法设置广告相关回调，广告错误码见 4.3

- **接口说明：**
  public void onAdPlayable(VideoAdBridgeCallback adPlayable);
  视频广告已经准备好，可以调用 playAd() 方法

public void onAdStart(VideoAdBridgeCallback adStart);

广告开始播放回调，通知开发者该视频广告开始播放

public void onAdEnd(VideoAdBridgeEndCallback adEnd);

广告播放结束时回调

public void adEnd(bool isSuccessfulView, bool isCallToActionClicked)

bool isSuccessfulView

返回用户是否完整观看了视频广告

bool isCallToActionClicked

返回用户是否点击了 CallToAction 按钮

public void onAdError(VideoAdBridgeErrorCallback adFailWithError);

获取广告失败

5.3 视频广告拉取

- 接口说明：

  public void loadAd();

  此接口只需调用一次，视频广告会在后台线程持续拉取，拉取到广告后会通过回调通知。请在执行 load 操作前先进行设置好数据监听接口。

注：视频文件拉取需要时间较长，建议在广告对象创建后立即进行数据监听和 load 操作

5.4 判断是否有可以播放的广告

- 接口说明：

  public bool isAdPlayable();

  返回当前是否有可以播放的广告，有返回 true，没有则返回 false

5.5 播放广告

- 接口说明：

  public void playAd();

  注：视频广告将根据设备的屏幕方向自动旋转。

6. DuAD_SDK_DEMO 使用方法

A. 将 android/Android.mk 文件放在工程 app/jni 文件夹下：该文件为 demo 示例配置文件，请修改该文件，定向到本地项目路径

B. 将 assets 目录下的文件复制到工程 Resources 文件夹下：该文件夹为广告位配置文件，及 demo 需要的资源图片

C. 将 Classes 文件夹下所有文件放在工程 Classes 文件夹下 :文件为广告展示 demo 代码，包括插屏广告、视频广告、原生广告；接入者可参考 demo 进行接入

D. 请保证 android 项目 package name 为 com.mobula.sample，同时正确修改 build.gradle