

# DU Ad Platform\_SDK for Unity 接入手册

---

DU Unity SDK v1.0

百度在线网络技术（北京）有限公司

# 目 录

1. 获取身份 .....	1
1.1 APP_ID .....	1
1.2 DAP 广告位 ID.....	1
2. 加载与配置 .....	1
2.1 加载 SDK 文件 .....	1
2.2 配置 AndroidManifest.xml .....	2
2.3 混淆代码.....	2
3.初始化-Json 配置 .....	3
4. 插屏广告使用 .....	4
4.1 构造方法.....	4
4.2 插屏广告回调.....	4
4.3 广告接口.....	5
4.3.1 预加载.....	5
4.3.2 加载 .....	6
4.3.3 展示 .....	6
4.3.4 销毁 .....	6
5. 广告墙使用 .....	7
5.1 构造方法.....	7
5.2 广告展示.....	7
6. 横幅广告使用 .....	8
6.1 构造方法.....	8
6.2 横幅广告回调.....	8
6.3 横幅广告加载.....	9
6.4 横幅广告销毁.....	9
7. 视频广告使用 .....	10
7.1 构造方法.....	10
7.2 视频广告回调.....	10
7.3 视频广告拉取.....	11
7.4 判断是否有可以播放的广告.....	11
7.5 播放广告.....	11
7.6 广告销毁.....	11

前提：DU Unity SDK 目前支持 Unity 5.0b19 及以上版本

## 1. 获取身份

本章描述 DU Ad Platform\_SDK 接入过程中需要的两个 ID：APP\_ID, DAP 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。

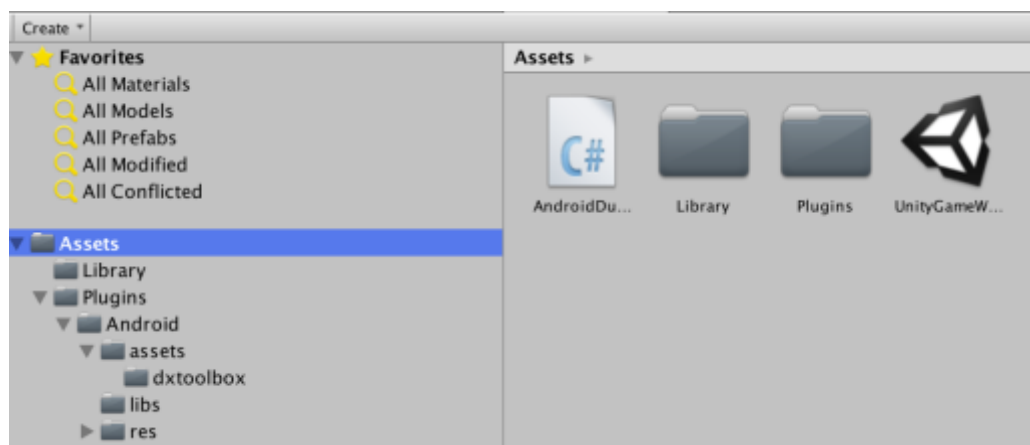
C. 代码

pid

## 2. 加载与配置

### 2.1 加载 SDK 文件

将 DAPUnity.unitypackage 加入 Unity3D 项目中



## 2.2 配置 AndroidManifest.xml

请按照如下步骤修改 *Assets/Plugins/Android* 文件夹下的 **AndroidManifest.xml** 文件

- A. 将 **YOUR\_PackageName** 字段更新为您的包名。  
请保证此处填写的包名与 DAP 平台上注册的一致，否则会出现广告获取问题。

- B. 将 **YOUR\_DAP\_APP\_ID** 字段更新为您的包名。

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="YOUR_PackageName">
    <application
    . . .
    <meta-data
        android:name="app_license"
        android:value="YOUR_DAP_APP_ID" />
    . . .
    <provider
        android:name="com.duapps.ad.stats.DuAdCacheProvider"
        android:authorities="YOUR_PackageName.DuAdCacheProvider"
        android:exported="false" />
    . . .
    </application>
</manifest>
```

- C. 添加 **PACKAGE\_ADDED** 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 *
-keepclassmembenames 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>;}
-keep class com.duapps.ad.banner.BannerListener { *; }
```

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

### 3.初始化-Json 配置

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

- 插屏和横幅广告：请将 pid 填入 native 部分
- 应用墙广告：请将 pid 填入 offerwall 部分
- 视频广告：请将 pid 填入 video 部分

```
{
  "native": [
    {
      "pid": "YOUR_DAP_PLACEMENT_ID(for interstitial & bannerad )"
    },
    {
      "pid": "YOUR_DAP_PLACEMENT_ID"
    }
  ],
  "offerwall": [
    {
      "pid": "YOUR_DAP_PLACEMENT_ID(for Offerwall)"
    }
  ],
}
```

```
"video": [
  {
    "pid": "YOUR_DAP_PLACEMENT_ID(for video)"
  }
]
}
```

4. 插屏广告使用

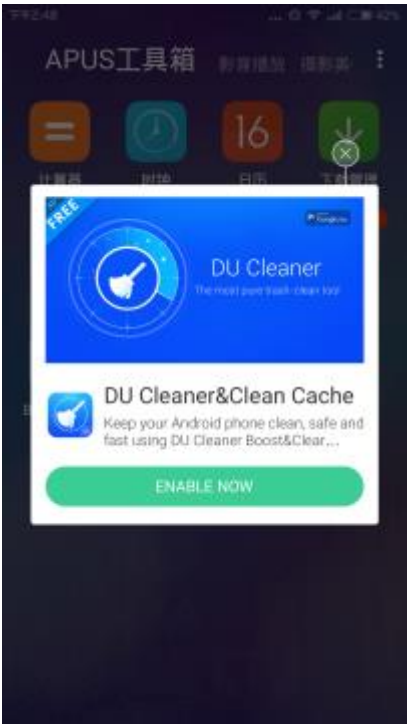


图 1： 半屏插屏广告样式



图 2： 全屏插屏广告样式

4.1 构造方法

- 接口说明：  
**public** InterstitialAd (int type, int pid);

参数	说明
int type	InterstitialAd.Type.SCREEN: 全屏广告 InterstitialAd.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

请按如下方法设置广告相关回调

```
interstitialAd.InterstitialAdReceive = delegate() {
    //广告获取成功回调
    Debug.Log ("InterstitialAdReceive");
    interstitialAd.ShowAd ();
};
interstitialAd.InterstitialAdPresent = delegate() {
    //广告展示回调
    Debug.Log ("InterstitialAdPresent");
};
interstitialAd.InterstitialAdClicked = delegate() {
    //广告点击回调
    Debug.Log ("InterstitialAdClicked");
};
interstitialAd.InterstitialAdDismissed = delegate() {
    //广告被用户关闭回调
    Debug.Log ("InterstitialAdDismissed");
};
interstitialAd.InterstitialAdError = delegate(int errorCode) {
    //广告错误回调
    Debug.Log ("InterstitialAdError : " + errorCode);
};
```

表 1 获取广告数据失败的错误码（AdError）

常量	错误码	说明
NETWORK_ERROR_CODE	1000	客户端网络错误
NO_FILL_ERROR_CODE	1001	没有获取到广告数据
LOAD_TOO_FREQUENTLY_ERROR_CODE	1002	请求接口过频繁
IMPRESSION_LIMIT_ERROR_CODE	1003	展示次数超出限制
SERVER_ERROR_CODE	2000	服务器错误
INTERNAL_ERROR_CODE	2001	服务器网络错误
TIME_OUT_CODE	3000	获取广告数据等待时间超时
UNKNOW_ERROR_CODE	3001	未知错误

### 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 FillAd()接口可以提前缓存广告，在 LoadAd()广告时可以更快展示。

建议：在广告展示的前置场景调用 FillAd()。

**\*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 ();`
- 代码示例：

```
interstitialAd.InterstitialAdReceive = delegate() {  
    Debug.Log ("InterstitialAdReceive");  
    interstitialAd.ShowAd ();  
};
```

#### 4.3.4 销毁

- 接口说明：  
`public void Dispose();`  
在退出插屏广告展示界面的时候,需要销毁插屏广告对象, 以防止内存泄漏
- 代码示例：

```
void OnApplicationQuit ()  
{  
    Debug.Log ("OnApplicationQuit");  
    If(interstitialAd != null)  
    {  
        interstitialAd.Dispose ();  
    }  
}
```



## 5. 广告墙使用

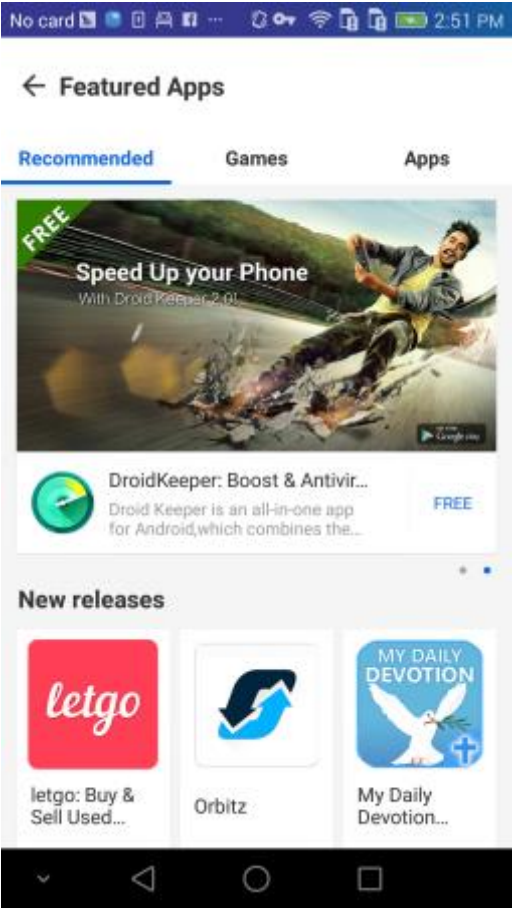


图 3 广告墙样式示例

### 5.1 构造方法

- 接口说明：  
`public OfferwallAd(int pid);`

参数	说明
int pid	广告位 ID, 该 pid 注册在 json 的 offerwall 数组中

### 5.2 广告展示

- 接口说明：  
`public void Show();`

- 代码示例：

```
OfferwallAd offerwallAd = new OfferwallAd (YOUR_DAP_PLACEMENT_ID);
offerwallAd.Show();
```

## 6. 横幅广告使用

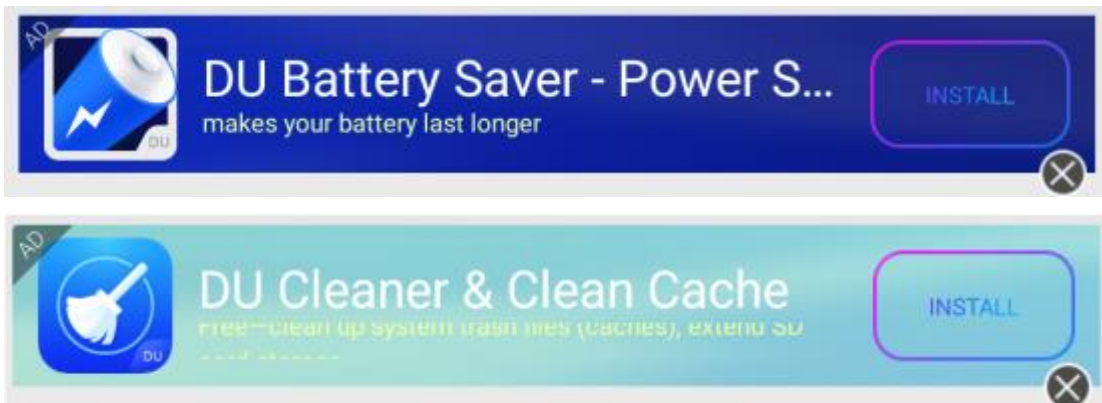


图 4:横幅广告样式

### 6.1 构造方法

- 接口说明：

**public** BannerAd (int pid, int cacheSize, int bgStyle, int closeStyle, int positionStyle);

参数	说明
int pid	广告位 ID, 该 pid 注册在 json 的 native 数组中
int cachesize	缓存广告个数
int bgSize	设置横幅广告背景颜色 BannerStyle.STYLE_BLUE:蓝色背景; BannerStyle.STYLE_GREEN:绿色背景;
int closeStyle	设置横幅广告关闭按钮的位置 BannerCloseStyle.STYLE_BOTTOM:右下角; BannerCloseStyle.STYLE_TOP:右上角
int positionStyle	设置广告位置 BannerCloseStyle.STYLE_BOTTOM:右下角; BannerCloseStyle.STYLE_TOP:右上角

### 6.2 横幅广告回调

请按如下方法设置广告相关回调，广告错误码见 4.2

```
// banner 广告各个回调
bannerAd.BannerAdLoaded = delegate() {
    //广告加载成功
    Debug.Log ("BannerAdLoaded");
};
bannerAd.BannerAdError = delegate(string errorMessage) {
    //广告错误回调
    Debug.Log ("BannerAdError : " + errorMessage);
};
```

## 6.3 横幅广告加载

- 接口说明：  
`public void LoadAd ();`  
\*注:请先设置广告监听器,再加载广告.

## 6.4 横幅广告销毁

- 接口说明：  
`public void Dispose();`  
在退出插屏广告展示界面的时候,需要销毁横幅广告对象, 以防止内存泄漏
- 代码示例：

```
void OnApplicationQuit ()
{
    Debug.Log ("OnApplicationQuit");
    If(bannerAd!= null)
    {
        bannerAd.Dispose ();
    }
}
```

## 7. 视频广告使用



图 5:视频广告播放界面



图 6:视频广告结果页

### 7.1 构造方法

- 接口说明：

**public** VideoAd(int pid);

参数	说明
int pid	广告位 ID, 该 pid 注册在 json 的 video 数组中

### 7.2 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 4.2

```
videoAd.VideoAdEnd = delegate(bool isSuccessfulView, bool isCallToActionClicked) {  
    //视频广告完全结束时回调  
    // bool isCallToActionClicked 返回用户是否点击了 CallToAction 按钮  
    Debug.Log ("isSuccessfulView : " + isSuccessfulView + ", isCallToActionClicked  
: " + isCallToActionClicked);  
};
```

```
videoAd.VideoAdError = delegate(string errorMessage) {  
    //广告错误回调  
    Debug.Log ("errorMessage : " + errorMessage);  
};  
videoAd.VideoAdPlayable = delegate() {  
    //广告已经准备好，可以调用 PlayVideo()方法  
    Debug.Log ("VideoAdPlayable");  
};  
videoAd.VideoAdStart = delegate() {  
    //广告开始播放时回调  
    Debug.Log ("VideoAdStart");  
};
```

## 7.3 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

## 7.4 判断是否有可以播放的广告

- 接口说明：

public boolean 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

## 7.5 播放广告

- 接口说明：

public void PlayAdVideo();

注：视频广告将根据设备的屏幕方向自动旋转。

## 7.6 广告销毁

- 接口说明：

public void Dispose();

在退出广告展示界面的时候,需要销毁横幅广告对象，以防止内存泄漏