

SwipeSDK for Android Access Guide

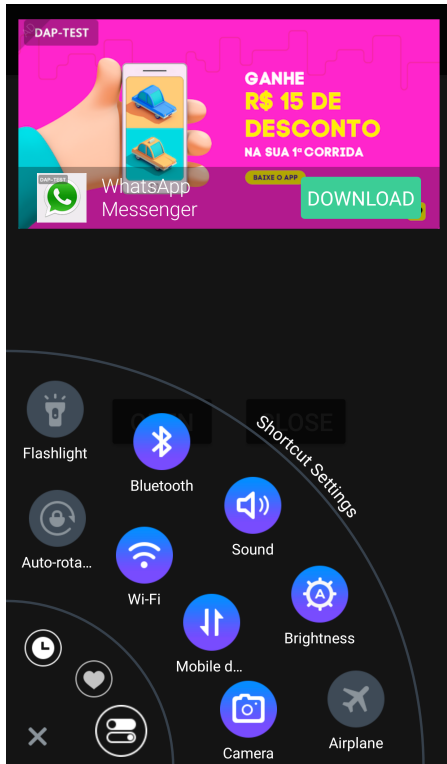
SwipeSDK V1.0.2

Baidu Online Network Technology(Beijing) Co.,Ltd

No.	DUAd10120150810
Date	2017-09-05
Ver.	1.0.2
Email	support_duad@baidu.com

Contents

1. Obtain Identity	1
2. Load SDK and Configuration	2
2.1 Load SwipeSDK.....	2
2.2 Configure AndroidManifest.xml	2
2.3 Obfuscate Code.....	4
3. Initialization	4
4. Related interface	5



A sample of Swipe Ads

1. Obtain Identity

Please refer to the chapter 3 in HW or CW version of DUADplatform SDK Access Guide to obtain necessary identities.

When applying for the DAP Placement ID , please make sure the app format you choose is **【Swipe】** .

Create placement ✕

* Placement name :

* Steps to trigger Ads : (i)

* Ad Format : Native Interstitial Offerwall
 Banner Video Trigger Caller
 Weather Wizard Swipe

Third-party platform ID configuration (For ads from third-party platform, fill in this field)

2. Load SDK and Configuration

This section describes the way of integrate SwipeSDK in Android app, configurate AndroidManifest.xml, and obfuscate code based on project needs.

2.1 Load SwipeSDK

Please refer to the chapter 4.1 in HW or CW version of DUADplatform SDK Access Guide to load DU SDK.

Notice: Swipe is not supported in the following cases

- (1) Android version is less than 11;
- (2) Android version is higher than 25 and no SYSTEM_ALERT_WINDOW permission ;
- (3) Phone with MIUI System ;

Currently DU SwipeSDK must rely on DU Ad Platform_SDK HW1.0.9.7 or CW1.0.9.6 (included) plus SDK version.

For accessing DuSwipe, the below operations are needed.

1) Copy the SwipeSDK-Vx.x.aar and nineoldandroids.jar to your Android Project, under the libs directory in root directory

2) Then configure build.gradle:

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile(name: 'DuappsAd-xW-xxx-release', ext: 'aar')
    // SwipeSDK must rely on DU AD SDK
    compile(name: 'SwipeSDK-Vx.x', ext:'aar')
}
```

* Note: The assigned directory of flatDir is where the aar file is placed.

2.2 Configure AndroidManifest.xml

Please refer to the chapter 4.1 in HW or CW version of DUADplatform SDK Access Guide to configure AndroidManifest.xml

A. Add a user-permission element to the manifest.

For accessing SwipeSDK, **MUST provide** the following permissions:

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

In order to ensure that the shortcut of SwipeSDK function can be used, please add the following permissions.

```
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
```

Notice: Please ignore the editor warning "Permission is only granted to system apps".

B. Add a meta-data element to the application element, and fill your DAP App ID as the value of "app_license" .

```
<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="xxxxxxxx" />
    <provider
        android:name="com.duapps.ad.stats.DuAdCacheProvider"
        android:authorities="packagename.DuAdCacheProvider"
        android:exported="false">
    </provider>
```

Notice : Replace the *packagename* with your app' s full package name.

C. Add Activity and BroadcastReceiver.

```
<activity
    android:name="com.swipe.BrightnessChangeActivity"
    android:excludeFromRecents="true"
    android:launchMode="singleInstance"
    android:noHistory="true"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.swipe.EmptyActivity"
    android:excludeFromRecents="true"
```

```
        android:launchMode="singleInstance"
        android:noHistory="true"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<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 Obfuscate Code

Please refer to the chapter 4.3 in HW or CW version of DUADplatform SDK Access Guide to Obfuscate Code.

3. Initialization

Please refer to the chapter 5 in HW or CW version of DUADplatform SDK Access Guide to finish SDK initialization.

For accessing Swipeads, please add a "native" tag in Json, and input your corresponding placement ID.

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

- Method

Use the following methods to initialize the SDK from onCreate in your Application class.

* Note: Please use this interface as described below, otherwise it may cause the initialization to be invalid.

```
/**
 * initialize SwipeSDK
 * @param pid DAP placement id
 */
SwipeMgr.onAppStart(this);
SwipeMgr.getInstance().setAdSid(int pid);
```

- Interface Instruction:

```
public static void setAdSid(int pid);
```

Parameters	Description
int pid	The pid of Swipe Ads.

4. Related interface

- (1) To switch the swipe function. The return value shows whether the feature was successfully opened

```
public boolean SwipeMgr.getInstance().setSwipeOpen(boolean value)
```

- (2) To determine whether the swipe function can open(see [2.1](#) not supported cases).

```
SwipeMgr.getInstance().isSwipeEnable();
```

- (3) Manually call the guide animation (the animation will be called automatically when Swipe turned on first time).

```
public void SwipeMgr.getInstance().showGuide();
```