DuCaller SDK for Android

Access Guide

DU Caller SDK v1.2.2.1

Contents

Precondition:

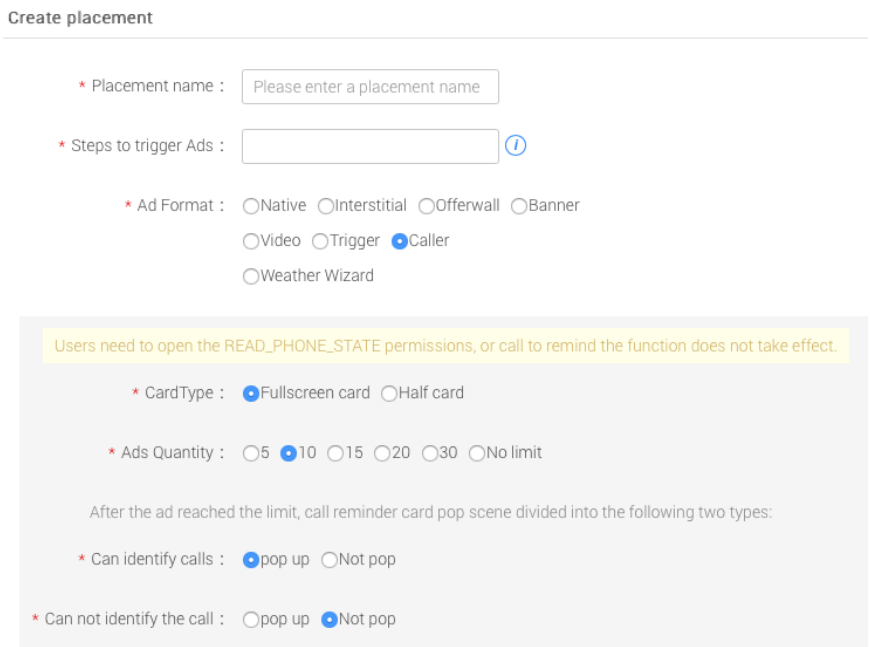
Currently DU Caller SDK must rely on DU Ad Platform_SDK HW1.0.9.7 or CW1.0.9.6 (included) plus SDK version.

If your Facebook SDK version is higher than v4.22, then you have to use DU Ad Platform_SDK HW1.0.9.8 or CW 1.0.9.7 (included) plus SDK version.

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 【Caller】.



DUCaller v1.2 add a function to control the DuCaller card type. We recommend you to use the fullscreen card to avoid violating Google policy.

2. Load SDK and Configuration

2.1 Load DU Ad Platform SDK

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

For accessing DuCaller, the below operations are needed.

1) Copy the DuCallerSDK-Vx.x.aar 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')     // DUcallerSDK must rely on DU AD SDK
    compile(name: 'DuCallerSDK-xxx', ext:'aar')
}
```

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 DuCaller, except the basic permissions:

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

Additional needed permissions is as below：

Permission name	**Description**	**Affect on monetization with Ad**	**Must have**
READ_PHONE_STATE	To access information from user's device	if without, can't pop up DuCaller	Yes
PROCESS_OUTGOING_CALLS	Get status of outgoing calls	if without, can't pop up DuCaller for outgoing calls (50% of all calls)	No
READ_CONTACTS	To read user's contacts on device	Can't identify the user contacts	No
RECEIVE_BOOT_COMPLETED	Allow to open DuCaller function when booting phone	Increasing call-receiving rate.	No
SYSTEM_ALERT_WINDOW	Allow to pop up window	if without, can't pop up DuCaller	Yes

*Note：**READ_PHONE_STATE**: it is mandatory；

PROCESS_OUTGOING_CALLS: It's recommended to have this permission, affect 50% monetization.

READ_CONTACTS: won't affect the monetization if without this permission, but will affect 56% user experiences.

SYSTEM_ALERT_WINDOW: it is mandatory for this SDK version.

- **Code Sample**：

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
```

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=" Your_PackageName.MobulaApplication"
    android:label="@string/app_name"
    . . . >
    <meta-data
        android:name="app_license"
         android:value="@string/Your_DAP_APP_ID" />
</application>
```

C. Declare the com.duapps.ad.stats.DuAdCacheProvider in the manifest. Replace the below packagename with your app’s full package name. Please make sure the package name at here is exactly the same as the package name you filled on DAP when registering you app. Otherwise, it will fail to get ad from DAP.

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

D. Register the BroadcastReceiver for receiving app install event.

Statically register the PACKAGE_ADDED Receiver in AndroidManifest.xml.

```
    <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>
```

E. Add DuCaller provider

- **Code Sample**：

```
<provider
    android:name="com.ducaller.fsdk.provider.FsdkContentProvider"
    android:authorities="your_package_name.FsdkContentProvider"
    android:exported="false" >
  </provider>
```

2.3 Obfuscate Code

A. Exclude classes of **DU Caller Ad Platform SDK** when obfuscating;

-keep class com.duapps.ad.**{*;}

B. Please refer to the chapter 4.3 in HW or CW version of DUADplatform SDK Access Guide, and DU Caller SDK Demo to Obfuscate Code.

```
-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.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>;}
```

*Note: For more about obfuscation methods, please refer to the official Android obfuscation document at: ${ android-sdk }/tools/proguard/

*

C: If accessing Facebook ads, please add the below class to proguard configuration.

```
-keep class com.facebook.ads.NativeAd
```

3. Initialization

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

For accessing DuCaller ads, please add a “ducaller” tag in Json, and input your corresponding placement ID.

```
{
    "ducaller": [
      {
      "pid": "YOUR_DAP_PLACEMENT_ID",
      "fbids":
              [ "YOUR_FACEBOOK_PLACEMENT_ID"]
       }
    ]
}
```

- **Method：**

  Add a call to DuCallerSDKHelper.init() from onCreate in your Application class

- **Interface Instruction：**

  public static void init(Context context，String pidsJson);

Parameters	Description
Context context	Activity Context
String pidsJson	The relationship between DAP Placement ID and Facebook Placement ID.

- **Code Sample：**

```
/**
  * Initialization Ducaller SDK
  * @param context Application Context
  * @param pidsJson configure json
  */
DuCallerSDKHelper.init(Context context, String pidsJson)
```

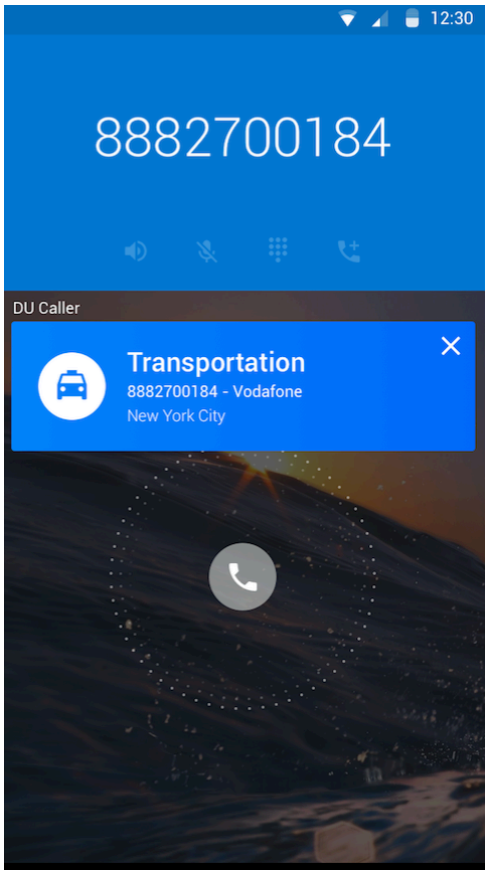
4. Other customized settings

```
/**
  * Set to show your app name (@param appLabel) on DuCaller page. If don’t set, your name will not be displayed on DuCaller page
* @param appLabel */
DuCallerSDKHelper.displayAppLabel(String appLabel)


/**
  * User disables DuCaller function
  */
DuCallerSDKHelper.disable()


/**
  * User enables DuCaller function
  */
DuCallerSDKHelper.enable()


/**
  * To see if DuCaller is enabled
```

* @return true when DuCaller is enabled， return false when DuCaller is disabled
*/

DuCallerSDKHelper.isEnabled()

/**
* Set a floating window before answering incoming calls or before making outgoing calls. Only for non-user contacts’ calls.
* @param disable. “true” for don’t show the floating window ;
“false” for show the floating window.
If don’t set, will show the floating window by default. */
public static void setFlowViewDisable(boolean disable)



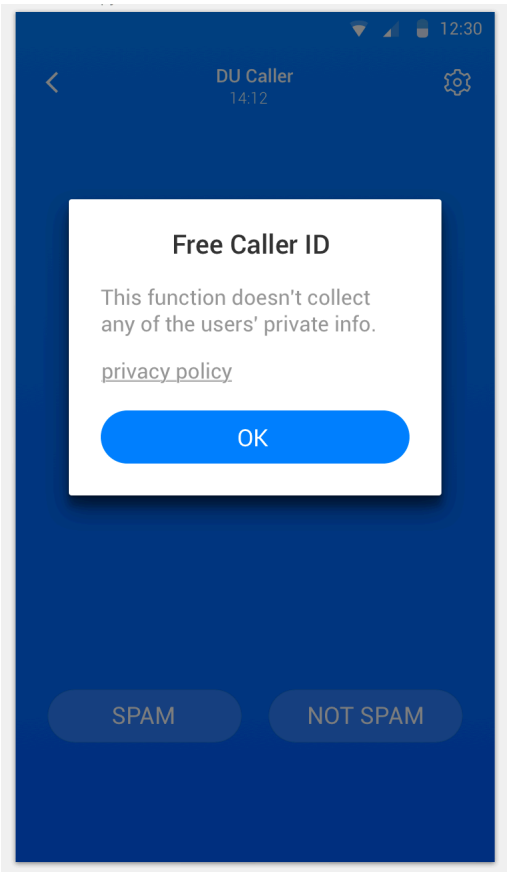
/**
* Set privacy policy State

* @param enable. “true” for enable ,the below privacy policy will be showed at the first time ; “false” for disable , will not show privacy policy。

If don’ t set, will NOT show privacy policy window by default.

*/

public static void setPrivacyPolicyState(boolean enable)



/**

* Guide user to enable DuCaller function

* Setting this means need to guide user to enable DuCaller function. If use this interface, DuCaller card will be shown only when isUserAccepted return “true”. If don’ t user this interface, then don’ t need to guide user to enable DuCaller function by default.

* @param need =ture means ducaller SDK will guide user to enable DuCaller function , need =false means don’ t need ducaller SDK to guide user to enable DuCaller function, (developers need to guide their user to enable Ducaller

```
function ) .
*/
public static void needSDKGuide(boolean need)



/**
 * When developers want to guide their user to enable Ducaller function, use this interface to enable Ducaller function when user accept to do that.
 * must use {@link com.ducaller.fsdk.main DuCallerSDKUtils} to set if guidance is needed.
 */
public static void setUserAccepted()

/**
 *To see if the user accept to enable DuCaller function.
If user has accepted, then show DuCaller card. If not, then continue guide user to accept.
 * @return “true” means user accept , “false” means user refuse */
public static boolean isUserAccepted()
```