

DU Ad Platform_SDK for Android Access Guide

Version: DuVideoSDK_1.1.0

Precondition:

DuVideo SDK currently supports Android 4.0 API level 14 (included) plus system versions.

DuVideo SDK must rely on DU Ad Platform_SDK CW1.0.9.9 or HW1.1.0.0 (included) plus SDK version.

Before accesing DuVideoSDK, You need to finish initialization, loading, and code obfuscation with HW or CW version.

DU Ad Platform_SDK for Android Access Guide

1. Obtain Identity
2. Load SDK and Configuration
 - 2.1 Add Video SDK
 - 2.2 Configure AndroidManifest.xml
3. Initialization
4. Request Video Ad
 - 4.1 Constructor of Video Ad
 - 4.2 Get Video Ad Instance
 - 4.3 Set Listener of Video Ad
 - 4.4 Load video ad
 - 4.5 Play video ad

A sample of DU Video Ad (playing page and download page)



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 **Video**.

Type-Incentive Ad : Force users to completely watched the video and can not skip.

Type-Non Incentive Ad : Users can choose to skip after 3s video playback.

2. Load SDK and Configuration

Please follow the below rules to configure. Otherwise, there might be exceptions at run time.

Please refer to the chapter 4 in HW or CW version of DUADplatform SDK Access Guide to load DU SDK.

2.1 Add Video SDK

Copy the DuVideoSdk-xxxx-release.aar to your Android Project, under the libs directory in root directory. Then configure build.gradle:

```
repositories {  
    flatDir {
```

```

    dirs 'libs'
  }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile(name: 'DuappsAd-xw-xxx-release', ext: 'aar')
    compile(name: 'DuVideoSDK-xxx-release', ext: 'aar')

    //This dependency is used to extract the color of the advertising theme.
    The specific version number should be changed according to the situation of
    your project. suggest to use version 21 above.
    compile 'com.android.support:palette-v7:23.4.0'

    /*If get crash because of this dependency, please change it to below:
    compile 'com.android.support:palette-v7:23.4.0' {
        transitive = false
    }
    */
}

```

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

2.2 Configure AndroidManifest.xml

Add the additional needed permissions for DuVideo (This permission is used to obtain the WiFi connection status and determine whether to deliver the advertisement according to the platform settings)

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

3. Initialization

1. Create a json file with mappings for the DAP Placement ID and other platform id.

```

{
  "video": [
    {
      "pid": "YOUR_DAP_PLACEMENT_ID",
      "fbid": "YOUR_FACEBOOK_PLACEMENT_ID",
      "amid": "YOUR_ADMOB_PLACEMENT_ID"
    }
  ]
}

```

Note:

1. If some of the DAP placements don't need ads from Facebook, the "fbids" part for that "pid" could be removed. If FBID is bound, please make sure the Facebook Audience Network sdk version is not lower than FB 4.23.0.
2. If some of the DAP placements don't need ads from AdMob, the "amid" part for that "pid" could be removed.
3. Add a call to `DuVideoSDK.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 other platform.

Note: Please make sure you call `DuAdNetwork.init()` for DUAd SDK in advance.

`DuAdNetwork.init` and `DuVideoAdSDK.init` can use the same json file.

Code Sample:

```
public void onCreate() {
    super.onCreate();
    //Init DAP SDK
    DuAdNetwork.init(this, getConfigJSON(getApplicationContext()));
    //Init Video SDK
    DuVideoAdSDK.init(this, getConfigJSON(getApplicationContext()));
}
```

4. Request Video Ad

4.1 Constructor of Video Ad

```
public DuVideoAd
```

4.2 Get Video Ad Instance

```
public static DuVideoAd DuVideoAdsManager.getVideoAd(Context mContext, int pid)
```

Parameters	Description
Context context	ACTIVITY CONTEXT
int pid	DAP placement ID, this pid must declared on Json's video array

Code Sample

```
DuVideoAd duVideoAd = DuVideoAdsManager.getVideoAd(this, YOUR_PID);
```

4.3 Set Listener of Video Ad

```
public void setListener(DuVideoAdListener adListener);
```

Parameters	Description
DuVideoAdListener adListener	Callback function returns: ad error, ad data, and ad click event.

```
public interface DuVideoAdListener {
    //Retrieve video ad successfully, so interface playAd() could be called
    then.
    void onAdPlayable();

    //Retrieve video ad failed. The codes are same as HW &CW version.
    void onAdError(AdError error);

    //Return when video started to play.
    void onAdStart();

    //Deprecated and replaced by the following 3 listeners
    void onAdEnd(AdResult result);

    //Return when the ad is clicked
    void onAdClick();

    //Return when the video is completely played
    void onVideoCompleted();

    //Return when the ad is closed
    void onAdClose();
}
```

```
public class AdResult {
    //Return whether user clicked the CallToAction button.
    public boolean isCallToActionClicked();

    //Return whether user watched the video ad completely.
    public boolean isSuccessfulView();
}
```

4.4 Load video ad

```
public void load()
```

This interface only needs to be called once. The video ad retrieving process is asynchronous. Please set the listener of video ad before calling `load()`.

Notice: Video files need long time to load, so we highly recommended you set the listener and call `load()` after getting video ad instance.

4.5 Play video ad

```
public boolean isAdPlayable()
```

Return whether there is playable video ad. True for playable.

```
public void playAd(Context context)
```

play video ad when there is playable video ad.

The video ad will be automatically rotated based on device's screen orientation.

Java Code Sample:

```
if (duVideoAd.isAdPlayable()) {
    duVideoAd.playAd(mContext);
}
```

Kotlin Code Sample:

```
if (duRewardedVideoAd.isAdPlayable) {
    duRewardedVideoAd.playAd(this@MainKotlinActivity)
    skipping_video.isEnabled = false
}
```