



Mobdynamics Android Ad SDK Integration Documentation

Date	Version	Description	Developer
2017.04.26	V2.2.5	Improved Appwall	Hongwu Lu
2017.07.04	V2.2.6	Supported native AD	Hongwu Lu

Content

1. Introduction	3
1.1. SDK Introduction	3
1.2. Requirements	3
2. Apply App ID and Ad Unit ID	3
2.1. App ID	3
2.2. Ad Unit ID	3
3. Initialization SDK.....	3
3.1. Obfuscated code configuration	4
3.2. Initialization	4
3.3. Modify your AndroidManifest.xml	5
3.4 Add Permissions [Optional]	5
4. Integration with Appwall.....	5
4.1 Configuration	6
4.2 Preload market data	8
4.3 Start App Market	9
5. Integration with Native Ads	9
5.1 Configuration	10

1. Introduction

1.1. SDK Introduction

Use Mobdynamics Android Ad SDK to maximize your app's revenue streams and save time, support below ad formats:

Appwall

Native Ad

1.2. Requirements

Android 2.3 (API Version 9) and up.

2. Apply App ID and Ad Unit ID

2.1. App ID

Make sure you have registered on Mobdynamics, you will get an App ID after creating your own app.

2.2. Ad Unit ID

You can create an ad unit under the app, you will get an Ad Unit ID after creating the ad unit.

3. Initialization SDK

Below are the aar packages in the SDK:

aar package	Function	Required
• anative_common.aar	public base package	Yes
• anative_appwall.aar	appwall ads package	No
• anative_native.aar	native ads package	No

Add aar packages to your local project:

Add necessary aar packages under the 'libs' folder of your project, and they will be served as a dependency library for the project.

- Some code show blow should be added into the dependencies of the project's build.gradle file:

```
compile 'com.google.code.gson:gson:2.8.0' // Required

compile(name: anative_common, ext: 'aar') // Required

compile(name: anative_appwall, ext: 'aar') // Optional, used in Appwall

compile(name: anative_native, ext: 'aar') // Optional, used in Native Ad

compile 'com.google.android.gms:play-services-ads:8.4.0' // Optional, Admob Native Ad
compile 'com.facebook.android:audience-network-sdk:4.23.0' // Optional, Facebook Native Ad
```

3.1. Obfuscated code configuration

Some code should be added in your proguard file according to the integrated package:

```
# Facebook Native Ad, Optional
-keep class com.facebook.** {*; }
-dontwarn com.facebook.**

# Admob Native Ad, Optional
-keep class com.google.android.gms.** {*; }
-dontwarn com.google.android.gms.**
```

3.2. Initialization

Call the initialization method in Application's onCreate:

```
Adsdk.initialize(Context context, String appId);
```

Parameter description:

- context: Application Context.
- appId: App ID that generated after creating your App.

NOTICE : Adsdk.initialize() must be called in main thread!

3.3. Modify your AndroidManifest.xml

If you had integrated with paly-services-ads, please add below configuration in your AndroidManifest.xml:

```
<activity
    android:name="com.google.android.gms.ads.AdActivity"

    android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|
    smallestScreenSize"

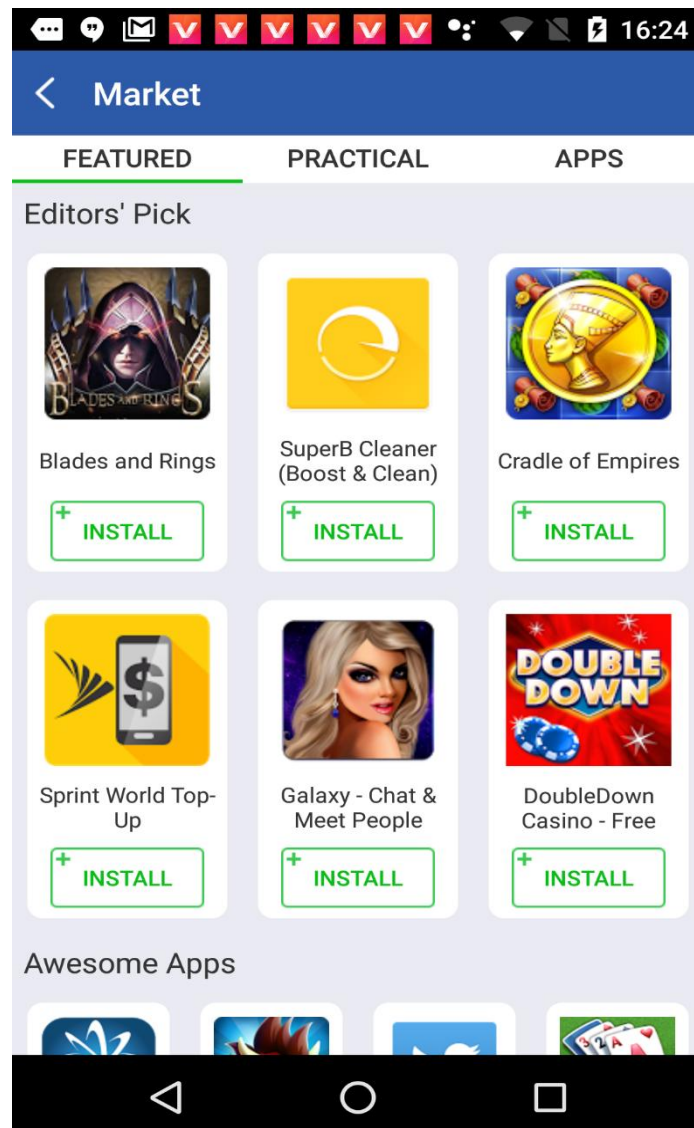
    android:theme="@android:style/Theme.Translucent" />
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

3.4 Add Permissions [Optional]

Some advertisers need sdk feedback the device id, so we recommond developers add permission: **android.persmission.READ_PHONE_STATE** in your convenience.

4. Integration with Appwall

A complete ad market's Activity or Fragment can be returned via the App Market interface with simple settings, below is the sample UI:



4.1 Configuration

You can customize App Market according to your product requirement:

1. Set market name:

```
public static void setAppMarketName(Context context, String name)
```

Parameter description:

- context: Application context
- name: the name of your App Market page (default value is : “Market”)

2. Config App Market style:

```
public static void setMarketStyle  
(Context context, HashMap<String, Integer> marketStyle)
```

Parameter description:

- context: Application context
- marketStyle : support to config below style, **value is resource id**. If set to null, mean default value:

- TITLE_BACKGROUND_COLOR
- TITLE_TEXT_COLOR
- TITLE_TEXT_SIZE
- TITLE_BAR_HEIGHT
- TITLE_BACK_DRAWABLE (default: 20x36 pixel)
- TABLE_BACKGROUND_COLOR
- TABLE_TEXT_COLOR
- TABLE_TEXT_SIZE
- TABLE_BAR_HEIGHT
- TABLE_INDICATOR_COLOR
- DK_BUTTON_BACKGROUND_COLOR
- DK_BUTTON_TEXT_COLOR
- INSTALL_TEXT_BACKGROUND_DRAWABLE (default: 140x60 pixel)
- INSTALL_TEXT_COLOR
- STATUS_COLOR
- NAVIGATION_COLOR
- WALL_BACKGROUND_COLOR
- AD_TITLE_TEXT_COLOR
- AD_DESCRIPTION_TEXT_COLOR
- CATEGORY_TEXT_COLOR
- CATEGORY_TEXT_SIZE
- CATEGORY_OF_RECOMMEND
- CATEGORY_OF_POPULAR
- CATEGORY_OF_LIKE
- AD_CLICK_COVER_LAYER_TRANS_BACKGROUND

Example:

```
Protected void onCreate(Bundle savedInstanceState) {  
    Adsdk.initialize(this);  
    setMarketStyle();  
}  
  
private void setMarketStyle() {  
    marketStyle.put(Constants.MarketStyle.TITLE_BACKGROUND_COLOR, R.color.white);  
    marketStyle.put(Constants.MarketStyle.TITLE_TEXT_COLOR, R.color.gray);  
    marketStyle.put(Constants.MarketStyle.TITLE_BAR_HEIGHT, R.dimen.title_bar_height);  
    marketStyle.put(Constants.MarketStyle.TITLE_TEXT_SIZE, R.dimen.anative_appwall_title_text_size);  
    marketStyle.put(Constants.MarketStyle.TITLE_BACK_DRAWABLE, R.drawable.appwall_adrss_ic_back);  
}
```

```

marketStyle.put(Constants.MarketStyle.TABLE_BACKGROUND_COLOR, R.color.white);
marketStyle.put(Constants.MarketStyle.TABLE_TEXT_COLOR, R.color.gray);
marketStyle.put(Constants.MarketStyle.TABLE_TEXT_SIZE, R.dimen.table_text_size);
marketStyle.put(Constants.MarketStyle.TABLE_INDICATOR_COLOR, R.color.light_gray);
marketStyle.put(Constants.MarketStyle.TABLE_BAR_HEIGHT, R.dimen.table_bar_height);

marketStyle.put(Constants.MarketStyle.DK_BUTTON_BACKGROUND_COLOR, R.color.black);
marketStyle.put(Constants.MarketStyle.DK_BUTTON_TEXT_COLOR, R.color.white);
marketStyle.put(Constants.MarketStyle.INSTALL_TEXT_BACKGROUND_DRAWABLE,
R.drawable.anative_appwall_adress_button_type1);
marketStyle.put(Constants.MarketStyle.INSTALL_TEXT_COLOR, R.color.gray);
marketStyle.put(Constants.MarketStyle.STATUS_COLOR, null);
marketStyle.put(Constants.MarketStyle.NAVIGATION_COLOR, null);
marketStyle.put(Constants.MarketStyle.WALL_BACKGROUND_COLOR, null);
marketStyle.put(Constants.MarketStyle.AD_TITLE_TEXT_COLOR, R.color.gray);
marketStyle.put(Constants.MarketStyle.AD_DESCRIPTION_TEXT_COLOR, R.color.light_gray);
marketStyle.put(Constants.MarketStyle.AD_CLICK_COVER_LAYER_TRANS_BACKGROUND,
R.color.cover_layer_background);

marketStyle.put(Constants.MarketStyle.CATEGORY_TEXT_COLOR, R.color.gray);
marketStyle.put(Constants.MarketStyle.CATEGORY_TEXT_SIZE, R.dimen.category_text_size);
marketStyle.put(Constants.MarketStyle.CATEGORY_OF_RECOMMEND, R.string.recommend);
marketStyle.put(Constants.MarketStyle.CATEGORY_OF_POPULAR, R.string.popular);
marketStyle.put(Constants.MarketStyle.CATEGORY_OF_LIKE, R.string.like);

AdSdk.setMarketStyle(this, marketStyle);
}

```

4.2 Preload market data

In order to optimize AD fill, conversion and improve the speed of loading market page ,
strongly suggest developers to preload appwall data. It will have benefit on revenue:

```
public static void preloadMarketData (Context context)
```

Parameter description:

- context: ApplicationContext

Example:

```

@Override
protected void onResume() {
    super.onResume();
    preloadMarketWall();
}

```



```
private void preloadMarketWall() {  
    AdSdk.preloadMarketData(this.getApplicationContext());  
}
```

4.3 Start App Market

We can launch App Market via Activity or Fragment

1. Launch with Activity:

```
public static void showAppMarket(Context context)
```

2. Launch with Fragment:

Call below API at first:

```
public static void setMarketFragmentMode(Context context, boolean isFragmentMode)
```

Parameter description:

- context: Application Context
- isFragmentMode: true for fragment mode

Use below method to get fragment instance when you need

```
Fragment fr = AdSdk.getFeatureFragment(Context context);
```

5. Integration with Native Ads

Sample UI:



5.1 Configuration

Copy anative_common.aar, anative_native.aar to your project

1. Be sure that the native ad sdk has been initialized in your application
2. Integration with 3rd party ad sources, please add the related dependencies in build.gradle file:
compile 'com.google.android.gms:play-services-ads:8.4.0' // AdMob Native Ad
compile 'com.facebook.android:audience-network-sdk:4.23.0' // Facebook Native Ad
3. Initialization and preload
Build a NativeAd object and pass into the unitId (the Ad Uni ID that was generated after creating the Ad Unit in Mobdynamics), set up ad load listening; pull the ad to the client side before show the ad to enhance the user experience and maximum your revenue.

```
INativeAd nativeAd = new NativeAd(Context activityContext, String unitId);
nativeAd.setNativeAdListener(new NativeAdListener() {
    @Override
    public void onAdLoaded() { // Ad load successfully callback
    }
    @Override
    public void onError(String error) {
    }
})
mNativeAd.load(); // load ad
```

4. Show ad

When the ad is preloaded successfully, call the interface to show the ad; adContainer is the container used to display the native ad.

```
if (nativeAd.isLoaded()) {  
    nativeAd.show(ViewGroup adContainer);  
}
```

Ad container configuration reference:

```
<FrameLayout  
    android:id="@+id/native_container"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"/>
```