

---

# POKKT Native Android SDK Integration Guide

v2.0.5

---

---

## Overview

Pokkt android sdk v2.0.xx supports offerwall as well as video ad campaigns feature. With help of this document, any application developer / publisher can integrate either feature or both the features in their application.

## Setting up the SDK

### Step 1: Add the Pokkt library (“pokkt\_xxx.jar”) to your project

Steps to follows:

- Copy **pokkt\_xxx.jar** to your project's root directory or in **libs** folder
- Right click your project name and select **Properties**.
- Select Java Build Path→Add External JARs
- Select **pokkt\_xxx.jar**. The POKKT library is now added to your project

### Step 2: Configure “AndroidManifest.xml” by adding permissions

Required permissions:

Common:

- READ\_PHONE\_STATE
- INTERNET
- ACCESS\_NETWORK\_STATE

Video specific:

- WRITE\_EXTERNAL\_STORAGE
- WAKE\_LOCK

Add following code snippet within <application...> tag in manifest file.

Common:

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

Video specific:

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

### Step 3: Add Activity definition

Add following Activities in manifest. Add both activities to support both features (offerwall and video).

Offerwall:

```
<activity
    android:name="com.app.pokktsdk.PokktManager"
    android:configChanges="keyboard|keyboardHidden|navigation|
        orientation|screenLayout|uiMode|screenSize"
    android:windowSoftInputMode="adjustPan">
</activity>
```

Video:

```
<activity
    android:name="com.app.pokktsdk.PlayVideoCampaignActivity"
```

---

```
    android:configChanges="keyboard|keyboardHidden|navigation|
        orientation|screenLayout|uiMode|screenSize"
    android:screenOrientation="landscape"
    android:windowSoftInputMode="adjustPan">
</activity>
```

#### Step 4: Add BroadcastReceiver (required only for offerwall)

Add following code snippet within </application> tag in manifest.

Offerwall:

```
<receiver android:name="com.app.pokktsdk.AppInstallBroadcastReceiver" >
    <intent-filter android:priority="1000" >
        <action android:name="android.intent.action.PACKAGE_INSTALL" />
        <action android:name="android.intent.action.PACKAGE_ADDED" />
        <data android:scheme="package" />
    </intent-filter>
</receiver>
```

#### Step 5: Add meta-data information

Add the following within </application> tag in manifest.

```
<meta-data
    android:name="security_key"
    android:value="{security key shared by POKKT}">
</meta-data>
<meta-data
    android:name="application_id"
    android:value="{app id shared by POKKT}">
</meta-data>
<meta-data
    android:name="U_ID"
    android:value="{user id shared by POKKT}">
</meta-data>
<meta-data
    android:name="integration_type"
    android:value=" integration_type_value">
</meta-data>
```

-- optionally you can also add the following meta data tag if you want to listen for additional events for the offerwall

```
<meta-data
    android:name="implementation_class"
    android:value="{your full class name which implements OfferWallEventListener}">
</meta-data>
```

#### Note:

- Value for integration is as following (by default, sdk supports both features i.e. default value is "0") :  
"1": Only offer wall  
"2": Only video
- For the time, when your app is in approval process and you have not received security key, app id, user id, etc. from Pokkt, you can use Pokkt sample app provided to you for demonstration purpose.
- You can skip adding meta tag and pass this information to PokktManager as a HashMap also as explained in *Invoke SDK into Application*.

---

## Step 7: Include Google Play Services SDK

Please follow the instructions below to include the Google Play Services SDK  
<http://developer.android.com/google/play-services/setup.html>

*Why is this required?*

Google has now changed policy w.r.t recognising the devices. It no longer allows the developer to read the Android\_ID. Instead a new Advertisers ID is needed to be used.

Please find more details below

<https://developer.android.com/google/play-services/id.html>

## Invoke SDK into Application

To invoke Pokkt sdk, create instance of PokktManager in application. This call will enable auto video caching.

`PokktManager.getInstance(Context ctxt).`

You can also create instance of PokktManager with meta information (as detailed in Step 5) supplied as HashMap. In this call you can choose to enable or disable auto video caching by setting parameter AUTO\_CACHE\_VIDEO as true or false. E.g.

```
HashMap<String, Object> metaMap = new HashMap<String, Object>();
metaMap.put(PokktManager.SECURITY_KEY, {security key shared by POKKT});
metaMap.put(PokktManager.APPLICATION_ID, {app id shared by POKKT});
metaMap.put(PokktManager.USER_ID, {user id shared by POKKT});
metaMap.put(PokktManager.INTEGRATION_TYPE, {integration type value});
metaMap.put(PokktManager.AUTO_CACHE_VIDEO, {true or false});
PokktManager.getInstance(Context ctxt, metaMap).
```

## OfferWall:

- Setup the CoinResponseListener to listen for pending coins for offerwall. Make sure to call pokktManager.getPendingCoins() function in your activity's onResume method.  
Add the following code to your activity or class which implements CoinResponseListener

```
@Override
public void earnedCoins(int coins) {
    if (points == -1) {
        //No Points Earned
    } else {
        //Points earned equal to coins
    }
}

@Override
public void earnedCoins(int coins, String transaction_id) {
    if (points == -1) {
        //No Points Earned
    } else {
        //Points earned equal to coins with transaction id
    }
}

@Override
public void requestFailed(String message) {
    //Called when pending coins request fails.
}
```

**There are two ways to invoke offerwall.**

---

### Open Asset Value

In this case, pokkt platform provides all offers with any asset value.

Sample code snippet:

```
PokktManager.getInstance(context).getCoins( Context context, boolean Close_On_Success);
```

### Fixed Asset Value

In this case, pokkt platform provides all offers with fixed asset value.

Sample code snippet:

```
PokktManager.getInstance(context).getCoins( Context context, String asset_value, boolean Close_On_Success);
```

### Pending Coins

In case after completing activity, if status of transaction is pending, then call `getPendingCoins()` method. You should always call this method in your calling activity's `onResume` method so that you can check for the pending coins for user.

Sample code snippet:

```
PokktManager.getInstance(context).getPendingCoins( Context context);
```

- **Optional** -- If you are using optional meta tag as mentioned in Step 5 and If you want to check whether offerwall campaigns are available before showing it to user then you can call **`pokktManager.checkCampaignAvailable(<context>);`** for the same. Also in your implementation class you will also get event when offer wall is closed. The interface `com.app.pokktsdk.OfferWallEventListener` provides two method for above mentioned events.

```
public void onOfferWallCampaignCheck(Context context,boolean campaignsAvailable); and  
public void onOfferWallClosed(Context context);
```

### Video:

In this feature, video file is cached on user's device. Before playing video or showing button to play video, Application should check whether video is cached or not by calling `pokktManager.isVideoAvailable()` or if downloading is undergoing then set listener `pokktManager.setDownloadCompletionListener (new DownloadCompletionListener)`. Please refer api documentation along with this documentation for more information.

Application can decide to play video as incent (*user will be gratified after watching complete video*) or non-incent (*user will not be gratified after watching complete video*). Application needs to pass 1<sup>st</sup> parameter(`alsIncentivized`) true for incent or false for non-incent of `pokktManager.playVideoCampaign(Boolean alsIncentivized, String aScreenName)`.

The response for the video playback and other video events can be handled in different listeners available for video playback as mentioned in API Guide. In Sample app the `VideoGratifiedListner` is being used to update the coins earned.

Developers also have option to not have auto caching for video, opting which they have to manually call the video caching using `pokktManager.cacheVideoCampaign(CurrentContext);`

