POKKT SDK Unity3d Integration Guide v2.0.6

Contents:

1. Introduction:

Thank you for choosing Pokkt SDK for Unity3d. This document contains all the information that is needed by you to setup the SDK with your project. Kindly note that these instructions are for Unity3d Version 4.x and above, older versions of Unity3d are not supported at this moment. Again the SDK is limited to Android platform as of now. You will be notified, as further platform-supports are available.

There is a sample app provided with the SDK. We will be referencing this app during the course of explanation in this document. It is suggested that you should check that app to understand the following process in detail.

2. Installation:

All we need is the file provided: PokktUnityDemo_v2.X.Xm.unitypackage

Export this package into your unity app/game. The “PokktLib.dll” and the java jars are mandatory. You can choose to ignore all the other stuff, but for the sake of this document, it is recommended that you should export everything.

3. AndroidManifest.xml Setup:

Step 1: 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 2: 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 3: 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 4: 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=" com.pokkt.unity.OfferWallEventListenerImpl">
</meta-data>
```

Note:

Value for integration is as follows

(SDK supports both features By default, i.e., default value is "0"):

"1": Only offerwall

"2": Only video

For the time being, 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 demo-purpose.

You can skip adding meta-tag and pass this information to PokktManager as one of the params-value also as explained in Invoke SDK into Application.

Step 6: 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 recognizing the devices. It no longer allows the developer to read the Android_ID. Instead a new Advertisers ID is needed to be use.
Please find more details below
https://developer.android.com/google/play-services/id.html

4. SDK Setup on Unity3d:

Initialize PokktManager

You can start with setting up the following values.

- Security Key
- Application Id
- User Id
- Integration Type
- Auto Cache Video

These values can also be set in AndroidManifest.xml. In order to disable auto-caching of videos, you are required to enter these values in your code and NOT in AndroidManifest.xml.

```
// SET THESE UP AS PER VALUES PROVIDED TO YOU
PokktManager.SecurityKey = "<your security key>";
PokktManager.ApplicationId = "<your application id>";
PokktManager.UserId = "<your user id>";
PokktManager.IntegrationType = "0";
```

After setting these values, make sure to set the auto cashing option in the SDK ONLY AFTER you set the params. Ref.:

```
// make sure to set auto caching once the params are set
PokktManager.GetInstance().AutoCaching = <true/false>;
```

5. Functionalities:

5.1 Offerwall

There are five events to listen too. These are:

- CoinResponseEvent
- CoinResponseWithTransIdEvent
- CoinResponseFailedEvent
- CampaignAvailablityEvent
- OfferwallClosedEvent

(Ideally)Add handlers to these events in the Awake() method of your MonoBehaviour class. Below are the references on how to use them:

Reference on how to consume them:

```
// handle pokkt offerwall events
PokktManager.GetInstance().Dispathcer.CoinResponseEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.CoinResponseWithTransIdEvent +=
(string message) =>
{
      string[] items = message.Split(
            new char[] { ',', ' ' },
            System.StringSplitOptions.RemoveEmptyEntries);
      string points = items[0];
      string transId = items[1];
};

PokktManager.GetInstance().Dispathcer.CoinResponseFailedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.CampaignAvailabilityEvent +=
(string message) =>
{
      bool available = message == "true";
};

PokktManager.GetInstance().Dispathcer.OfferwallClosedEvent +=
(string message) =>
{
};
```

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().GetFreeCoins();
```

Fixed Asset Value: In this case, POKKT platform provides all offers with fixed asset value.
Sample code snippet:

```
PokktManager.GetInstance().GetFreeCoins(_assetValue);
```

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().GetPendingCoins();
```

[Optional] Check for campaign availability: If you are using optional meta-tag as mentioned in Step 5, you can call the following to check for campaign availability:

```
PokktManager.GetInstance().CheckOfferwallCampaign();
```

The result will be noticed with the event:

```
CampaignAvailabilityEvent
```

Moreover, you can listen to the following event to know whether offerwall has been closed or not:

```
OfferwallClosedEvent
```

5.2 Video:

There are 7 events to manage the video caching and its playback, these are:

- VideoClosedEvent
- VideoDisplayedEvent
- VideoSippedEvent
- VideoCompletedEvent
- VideoGratifiedEvent
- ShowIntentEvent
- DownloadCompletedEvent
- DownloadFailedEvent

Add listeners to these events in the init() method of your Node or similar class. These are all of EventCustom type. Below are the reference on how to use them:

Reference on how to consume them:

```
// handle pokkt video ad events
PokktManager.GetInstance().Dispathcer.VideoClosedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoSkippedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoCompletedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoGratifiedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoDisplayedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.ShowIntentEvent +=
(string message) =>
{
};
```

```
PokktManager.GetInstance().Dispathcer.DownloadCompletedEvent +=
(string message) =>
{
      string text = "Video VC is: " + message;
};

PokktManager.GetInstance().Dispathcer.DownloadFailedEvent +=
(string message) =>
{
};
```

A video file is cached on user's device. You can set the auto-caching option in the beginning, as mentioned earlier in this document. In case of manual caching, call the following to start video caching:

```
PokktManager.GetInstance().StartVideoCaching();
```

Before playing video or showing button to play video, Application should check whether video is cached or not by calling the following:

```
PokktManager.GetInstance().IsVideoAvailable()
```

You should listen to "DownloadCompletedEvent" to check whether download is completed or not, you can show the play buttons once you receive this event.

Furthermore, 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). Followings are the method calls to me made:

```
PokktManager.GetInstance().GetVideo();
PokktManager.GetInstance().GetVideoNonIncent();
```

Next, you can listen to VideoGratifiedEvent to get the coins earned, if at all, by watching the last video.

5.3 Logs and Toasts:

You can use the following command to log some debug messages:

```
PokktManager.GetInstance().showLog("pokkt init...");
```

You can use the following command to display a message as Toast on android device:

```
PokktManager.GetInstance().showToast("pokkt init...");
```

This concludes the integration documentation. It is highly suggested that you should check the sample app that is provided to you to understand it better.