POKKT SDK v1.0.0 Integration Guide for ShiVa (Android)

Contents:

1. Introduction:

Thank you for choosing Pokkt SDK for ShiVa. This document contains all the information which is needed to implement Pokkt SDK.

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:

The Pokkt SDK v1.0.0 for ShiVa comes in two file. One is ".ste" files and one ".zip" file. Here is the details

1. **com.pokkt.extension.ste:** This is the plugin which is need to import in project from ShiVa editor (**Data Explorer->Import->Archive)**.

2. **PokktPluginSample.ste:** This is the Sample Project and it contains few AIModel and HUD. This sample project will show the video campaign and offerwall campaign. One of the AI_Model (which is **AI_POKKT**) from this sample is very important to implement PokktSDK.

3. **res.zip:** This is the resource which is needed. This contains **drawable**, **layout** and **values** folder.

Note: Please **do not copy** the code points from this PDF file as it may introduce unwanted characters and space in your code. Instead please refer to sample app source code provided with the sample app.

3. Manifest Changes:

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.ShowOfferwallActivity"
        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|scre
        enLayout|uiMode|screenSize"
        android:label="@string/app_name"
        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="offerwallDelegate"
android:value=" OfferwallEventsHandler" />

<meta-data android:name="videoDelegate"
android:value=" VideoEventsHandler" />
```

Step 5: 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 with respect to recognising 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. Code Integration:

Implementation Steps

Import the POKKT extension (com.pokkt.extension.ste) and follow the below steps:

- Import com.pokkt.extension.ste in ShiVa Editor from Data Explorer. After importing, this can be seen in project under plugin folder as PokktNativeExtension. Now drag this plugin from Data Explorer into Game Editor under plugin section.

- Now import PokktPluginSample.ste where you will see multiple AIModel and HUD. This sample project contains one important AIModel which is AI_POKKT. **This is mandatory to use in your project**. This sample project contains few more AIModel which listen event and those gets trigger from AI_POKKT handler. So update the AI_POKKT AIModel handler according to your AIModel.
  Here is the example how to do that:
  This is one of the event ("onDownloadCompleted Handle") from AI_POKKT.
  user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", "onDownloadCompleted Handle", values)
  Now I want to dispatch event from this model to other model where we actually need it so Here "AI_PlayVideo" is the AIModel name where we want to listen "onDownloadCompleted" event. So this you need to change according to your AIModel.

- Now export the game from Data Explore. On clicking export button will show one popup where select "Runtime Package (.stk)" and select Android as option and then export it. Then will generate .stk file.

- Once project has been exported then open ShiVa Authoring tools and do the all required setting for Android in Settings option. There you need to assign path for Android SDK, Android NDK and Cygwin path.

  Requirement for making Android build as suggested by ShiVa team:

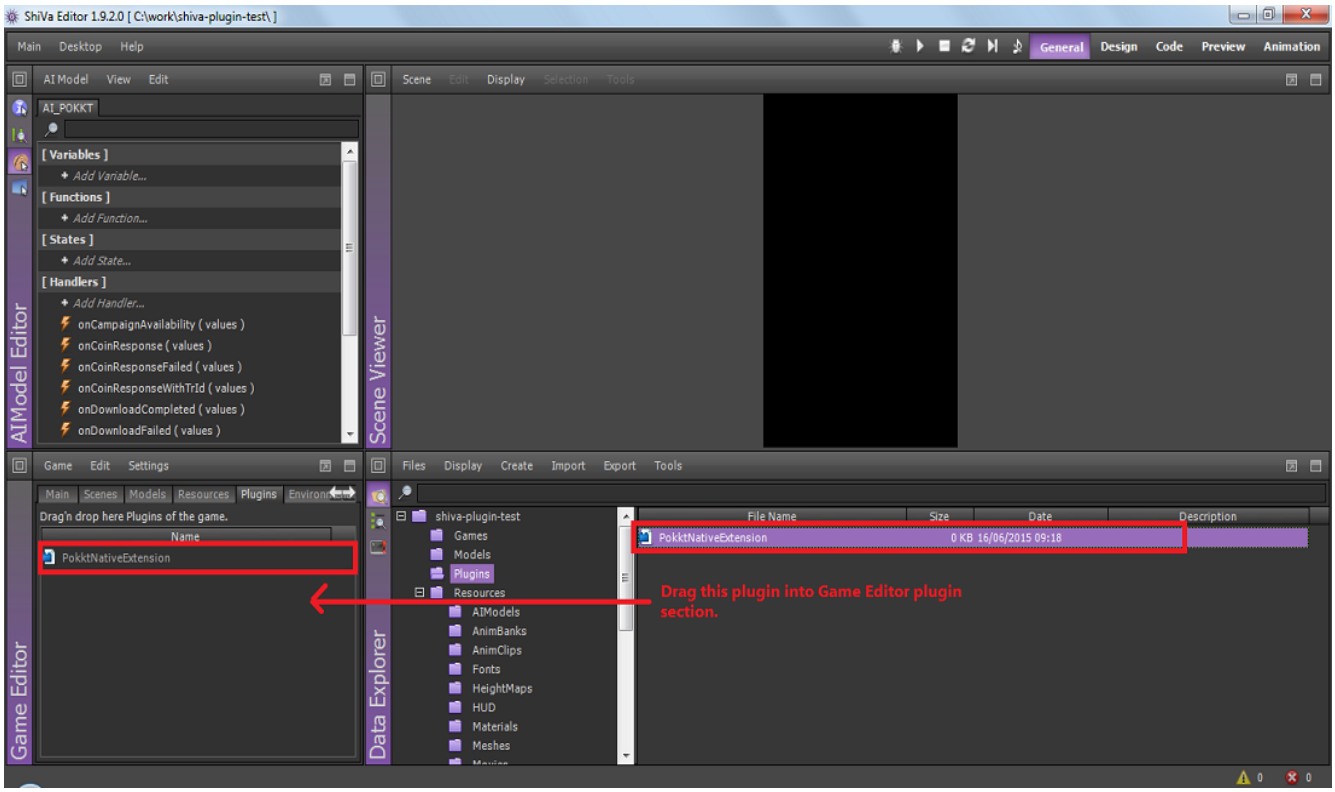
  - Android SDK.
    Target SDK: Min 4.0 and Max 4.2.
  - Android NDK r9d.
  - Cygwin

  Once it is done then follow the below steps:

  1. Select Content tab and give the path of ".stk" file.
  2. Select Authoring tab and choose Authoring type as "Project". Also give the Bundle identifier which will be the package ID and Version and Version code.

Once all these two steps are done then click on Build button which will export the Android Project which you can easily import in Android ID like Eclipse or other ID After importing the project in eclipse or android editor, need to do some changes. Here is the steps:

1. First update AndroidManifest.xml as suggested in 3rd point of this document.
2. Check the downloaded folder which you got from POKKT and get the res.zip. This res.zip contains **values**, **layout** and **drawable** folder. So extract the zip file and add it in imported project.



Screen shot: How to add Pokkt plugin in your project.

- Common

1. First we need to perform operation *PokktNativeExtension. SET_PARAMS* which will set the important parameters for POKKT. It includes *application id*, *security key* and *integration type* which are must for all types of integrations.

2. If you are doing server to server integration then please perform operation: *PokktNativeExtension.SET_THIRD_PARTY_USER_ID.*

3. While in development please call *PokktNativeExtension.performOperation (PokktNativeExtension.SET_DEBUG, "true")* to see Pokkt debug log and for showing toast please call

PokktNativeExtension.performOperation (PokktNativeExtension .SHOW_TOAST, "message").

please make sure to change this for production build.

PokktNativeExtension.performOperation (PokktNativeExtension .SET_DEBUG, "false").

- Session

1. Pokkt SDK is adding session tracking for which we have *PokktNativeExtension. START_SESSION .*
2. *START_SESSION* should be called at the start of the application and once only also will be need to provide *applicationId*, *securityKey* and *IntegrationType* so for this please call setParam before calling *START_SESSION*.
3. We should call *PokktNativeExtension.END_SESSION* at the end of his application and once only

- OfferWall

1. For OfferWall you can set two additional parameters which are *offerWall Asset Value* and *SET_CLOSE_ON_SUCCESS_FLAG. OfferWall Asset Value* is required if you only want to show campaigns of certain value on OfferWall. So when you call the show offerwall campaign there if you pass the value then it will show such campaign which contains certain value. And if you don't pass then It has default value as empty. *SET_CLOSE_ON_SUCCESS_FLAG* is required if you wish to close the OfferWall after user has completed one offer. It's default value is false.
2. Before calling another method for offerWall, please make sure that you have already called *SET_PARAMS* first.
3. To show OfferWall please call
   PokktNativeExtension.performOperation (PokktNativeExtension.GET_COINS, "");
   or
   PokktNativeExtension.performOperation (PokktNativeExtension.GET_COINS, "400")
   where 400 is a fix assets value for offerwall campaign and if we pass empty then all offerwall can be seen in offerwall campaign popup.
4. In the screen or activity where you have button to show offerwall, in that activity onResume you should call *GET_PENDING_COINS*
   so that you get a callback to award points to the user after he has come back

to game/app after finishing with OfferWall. You will get a callback for this call in your *AI_POKKT AIModel where onCoinResponse or onCoinResponseWithTrId will trigger in case of success otherwise onCoinResponseFailed will trigger in case of failure.*

5. You can call *PokktNativeExtension.performOperation (PokktNativeExtension. CHECK_OFFERWALL_CAMPAIGN, "")* to check whether the campaigns are available before showing OfferWall button to user. You will get a callback for this call in your *AI_POKKT.onCampaignAvailability* implementation.

- Video

1. For video you can set five additional parameter which are autoCacheVideo, skipEnabled, defaultSkipTime, screenName and incentivised.

2. autoCacheVideo is required if you want to automatically cache video on user device. You need to set this parameter in SET_PARAMS as true. If you set it as false then video will not be automatically cached and will have to call *PokktNativeExtension.performOperation (PokktNativeExtension .CACHE_VIDEO_CAMPAIGN, "")* to start caching videos on device.

3. If you want to enable/disable the skip button on video screen please set *SET_SKIP_ENABLED* as "true"/"false". The default value for *SET_SKIP_ENABLED* is false. You can call like this:
PokktNativeExtension.performOperation (PokktNativeExtension. SET_SKIP_ENABLED, "true")

- Optional Parameters

- There are also provision for developers to provide extra user data available with them to pokkt. We currently support following data points: *name, age, sex, mobileNo, emailAddress, location, birthday, maritalStatus, facebookId, twitterHandle, education, nationality, employment and maturityRating.* For reference, please check *AI_MainMenu* AIModel.

5. Functionalities:

5.1 Offerwall

There are five events to listen. These are:

1. onCampaignAvailability
2. onCoinResponse
3. onCoinResponseFailed
4. onCoinResponseWithTrId
5. onOfferwallClosed

Reference on how to consume them:

```
// handle pokkt offerwall events

function AI_POKKT.onCampaignAvailability ( values )
-------------------------------------------------------------------------------
user.sendEvent ( application.getCurrentUser ( ), "AI_Offerwall", "onCampaignAvailableHandle" )-
-------------------------------------------------------------------------------
end
```

Here values doesn't have any data. It is empty and it is not useful.

```
function AI_POKKT.onCoinResponse (earnedPoints)
-------------------------------------------------------------------------------
user.sendEvent ( application.getCurrentUser ( ), "AI_Offerwall", "onCoinResponseHandle",
earnedPoints)
-------------------------------------------------------------------------------
end
```

Here earnedPoint contains earned point in string. So you can use this earned point according to need.

```
function AI_POKKT.onCoinResponseFailed ( values )
-------------------------------------------------------------------------------
user.sendEvent ( application.getCurrentUser ( ), "AI_Offerwall", "onCoinResponseFailedHandle"
)
-------------------------------------------------------------------------------
end
```

Here values doesn't have any data. It is empty and it is not useful.

```
function AI_POKKT.onCoinResponseWithTrId ( earnedPoints, transactionID )
-------------------------------------------------------------------------------
 user.sendEvent ( application.getCurrentUser ( ), "AI_Offerwall",
"onCoinResponseWithTrIDHandle", earnedPoints , transactionID)
-------------------------------------------------------------------------------
end
```

Here we receive **earnedPoint** and **transactionID**. So use it according to need.

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:
  PokktNativeExtension.performOperation (PokktNativeExtension.GET_COINS, "")

- **Fixed Asset Value:** In this case, POKKT platform provides all offers with fixed asset value.
  Sample code snippet:
  PokktNativeExtension.performOperation (PokktNativeExtension.GET_COINS, "400")
  -- This will load only such campaign which match with 400 assets value.

Pending Coins: In case after completing activity, if status of transaction is pending, then check GET_PENDING_COINS. You should always call this method in your calling activity's onResume method so that you can check for the pending coins for user. No need to pass anything in parameter. Keep it empty string.
Sample code snippet:
PokktNativeExtension.performOperation (PokktNativeExtension.GET_PENDING_COINS, "")

[Optional] Check for campaign availability: If you want to check is there Offerwall campaign is/are available or not then call PokktNativeExtension.performOperation (PokktNativeExtension. CHECK_OFFERWALL_CAMPAIGN, "") . and this if campaigns are available then onCampaignAvailability will trigger.

5.2 Video:

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

- onDownloadCompleted
- onDownloadFailed
- onVideoClosed
- onVideoCompleted
- onVideoDisplayed
- onVideoGratified
- onVideoSkipped

Reference on how to consume them:

```
function AI_POKKT.onDownloadCompleted ( values )
--------------------------------------------------------------------------------
user.sendEvent ( application.getCurrentUser ( ), "AI_PlayVideo", "onDownloadCompleted
Handle", values )
--------------------------------------------------------------------------------
end
```

This event will trigger on video download completed and values contains total earned points.

function AI_POKKT. **onDownloadFailed** (values)

---

user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", " onDownloadFailedHandle", values)

---

end

This event will trigger on video download failed. Here values doesn't have any data. It is empty and it is not useful.

function AI_POKKT. **onVideoClosed** (values)

---

user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", " onVideoClosedHandle", values)

---

end

This event will trigger on video close. Here values doesn't have any data. It is empty and it is not useful.

function AI_POKKT. **onVideoCompleted** (values)

---

user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", " onVideoCompletedHandle", values)

---

end

This event will trigger on video completed. Here values doesn't have any data. It is empty and it is not useful.

function AI_POKKT. **onVideoDisplayed** (values)

---

user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", " onVideoDisplayedHandle", values)

---

end

This event will trigger on video displayed. Here values doesn't have any data. It is empty and it is not useful.

function AI_POKKT. **onVideoGratified** (values)

---

user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", " onVideoGratifiedHandle", values)

---

end

This event will trigger on video gratified. So here values will have earned points in string format.

function AI_POKKT. **onVideoSkipped** (values)

---

user.sendEvent (application.getCurrentUser ( ), "AI_PlayVideo", " onVideoSkippedHandle", values)

This event will trigger on video skipped. Here values doesn't have any data. It is empty and it is not useful.

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

```
PokktNativeExtension.performOperation ( PokktNativeExtension.CACHE_VIDEO_CAMPAIGN, "" );
```

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

```
PokktNativeExtension.isVideoAvailable ( )
```

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). You must provide the screen-name parameter for it. Followings are the method calls to me made:

```
PokktNativeExtension.performOperation ( PokktNativeExtension.GET_VIDEO, "screen_name" )
PokktNativeExtension.performOperation ( PokktNativeExtension.GET_VIDEO_NON_INCENT, " screen_name " )
```

Next, AI_POKKT.onVideoGratified callback will call to get the coins earned, if at all, by watching the last video.

6. Debugging and Logging

You can enable the SDK logs by setting the debugging option to true anytime.

```
PokktNativeExtension.performOperation ( PokktNativeExtension.SET_DEBUG, "true" )
```

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

```
PokktNativeExtension.performOperation ( PokktNativeExtension.SHOW_LOG, "showing log" )
```

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

```
PokktNativeExtension.performOperation ( PokktNativeExtension.SHOW_TOAST, "testing toast message" )
```

7. Important Points

- Please do not copy the code points from this pdf as it may introduce unwanted characters and space in your code. Instead please refer to sample app source code in pokkt bundle.

- Please also refer to sample app source code for better understanding of implementation.