

## **POKKT SDK v4.1.0 Integration Guide for ShiVa (Android)**

## **Contents:**

1. Overview
2. Configuration steps
3. Implementation steps
4. Important Points

## 1. Overview

Thank you for choosing Pokkt SDK Plugin v4.1 for ShiVa. Pokkt SDK supports Offerwall as well as Reward-Ad campaigns feature. This document contains all the information that is needed by you to setup the SDK with your project. Please follow these steps as per your integration requirement (Reward/Offerwall/Both). The current plugin supports mediation for various third party ad-networks.

These are:

- AdColony
- AppLovon
- Chartboost
- Fyber
- InMobi
- SuperSonic
- TapJoy
- Vungle
- AdMob
- Facebook

A separate set of documents is provided for each of these, explaining the implementation process.

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. Configuration Steps

The Pokkt SDK v4.1.0 for ShiVa comes in two file. One is ".ste" files and 2nd is ".zip" file. Here is the details

1. **com.pokkt.extension\_4.1.ste**: This is the plugin which you need to import in project from ShiVa editor (**Data Explorer->Import->Archive**).
2. **PokktSample.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 mplement PokktSDK.
3. **res.zip**: This is the resource file which is needed. This contains **drawable**, **layout** and **values** folder. You need to copy in your android project which gets exported from ShiVa tool.

### *Android Manifest xml update:*

1. Add following permissions in your App xml under android manifest, If not already there.

```
<!-- These permissions are mandatory to run Pokkt SDK -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- These permissions are strongly recommended and will result in higher performance -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<!-- This permission is optional but will improve SDK feature-->
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
```

2. Add the following activity in your App xml under android manifest for OfferWall Integration

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

3. Add the following activity in your App xml under android manifest for Video Integration

```
<activity
    android:name="com.app.pokktsdk.PlayVideoCampaignActivity"
    android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize"
    android:label="@string/app_name"
```

```

        android:screenOrientation="landscape"

        android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified" >
    </activity>

```

#### 4. Add Following Broadcast receiver for OfferWall Integration in App xml under android manifest

```

<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>

```

#### 5. Add the following activity in your App xml under android manifest for Pokkt Interstitial Integration

```

<activity

    android:name="com.app.pokktsdk.InterstitialActivity"

    android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|smallest
    ScreenSize"

    android:label="@string/app_name"

    android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified" >

</activity>

```

#### 6. Add Following meta tag for google play services (Goole play services is required and should be part of your project, if not, please add the GooglePlayService component in project.)

```

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

```

#### 7. Add a meta data tag for offerwall campaign. you will have to implement the *IOfferwallDelegate* interface in your project to listen for all offerwall related events. (refer to pokkt sample for example).

```

<meta-data android:name="offerwallDelegate" android:value="com.pokkt.plugin.common.PokktOfferwallDelegate" />

```

#### 8. Add following Service and receiver in manifest for google analytics (Optional).

```

<receiver android:name="com.google.android.gms.analytics.AnalyticsReceiver" android:enabled="true">

    <intent-filter>

        <action android:name="com.google.android.gms.analytics.ANALYTICS_DISPATCH" />

    </intent-filter>

</receiver>

```

```
<service android:name="com.google.android.gms.analytics.AnalyticsService"  
    android:enabled="true"  
    android:exported="false"/>
```

9. Add following Service in manifest for in-app notification.

```
<service android:name="com.app.pokktsdk.notification.NotificationService"  
    android:label="PokktNotificationService"  
    android:exported="false"/>
```

**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. Implementation Steps

Import the POKKT extension ([com.pokkt.extension\\_4.1.ste](#)) and follow the below steps:

- Import [com.pokkt.extension\\_4.1.ste](#) in ShiVa Editor from Data Explorer. Now drag this plugin from Data Explorer into Game Editor under plugin section.
- Now import [PokktSample.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 ("[onCachingCompletedHandle](#)") from [AI\\_POKKT](#).

```
user.sendEvent ( application.getCurrentUser ( ), "AI_PlayVideo", "onCachingCompletedHandle",
screenName,isRewarded,backButtonDisabled,defaultSkipTime,shouldAllowSkip,shouldAllowMut
e,shouldSkipConfirm,skipConfirmMessage,skipConfirmYesLabel,skipConfirmNoLabel,
skipTimerMessage,incentiveMessage,rewardAmount)
```

and handling this event in [AI\\_PlayVideo](#) AIModel. Please check our given sample.

- 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.
- Now export android project of your ShiVa .stk file using 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.

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 and 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.

- Common

1. First we need to set *SET\_APPLICATION\_ID* and *SET\_SECURITY\_KEY* which you will get once you create new App in Pokkt.com and then call *INIT\_POKKT*. Please check *AI\_MainMenu* in our given example.
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" )*.
4. Android MinSDKVersion should be  $\geq 14$ .
5. To use google analytics, please set AnalyticsType and Analytics ID:  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_SELECTED\_ANALYTICS\_TYPE, PokktNativeExtension.GOOGLE\_ANALYTICS )*  
  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_GOOGLE\_ANALYTICS\_ID, "ID" )*
10. To use flurry analytics please set AnalyticsType and Flurry Application Key:  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_SELECTED\_ANALYTICS\_TYPE, PokktNativeExtension.ANALYTICS\_TYPE\_FLURRY)*  
  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_FLURRY\_APPLICATION\_KEY, "ID" )*
11. To use mix panel analytics please set AnalyticsType and Mix PanelProject Token:  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_SELECTED\_ANALYTICS\_TYPE, PokktNativeExtension.ANALYTICS\_TYPE\_MIXPANEL)*  
  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_MIX\_PANEL\_PROJECT\_ID, "ID" )*
12. To use mix panel analytics please set AnalyticsType and Fabric Token:  
*PokktNativeExtension.performOperation(PokktNativeExtension.SET\_SELECTED\_ANALYTICS\_TYPE, PokktNativeExtension.ANALYTICS\_TYPE\_FABRIC)*



14. Please call *TRACK\_IAP* to send any in-app purchase information to Pokkt.

Like below you can do this.

```
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_CURRENCY_CODE, "<Country code>" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_DESCRIPTION, "<description>" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_PRICE, "<price>" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_PRODUCT_ID, "<productid>" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_PURCHASE_DATE, "<date>" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_PURCHASE_SIGNATURE, "purchaseSignature" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_TITLE, "title" )
PokktNativeExtension.performOperation(PokktNativeExtension.IAP_STORE_SET_PURCHASE_STORE, PokktNativeExtension.IAP_STORE_TYPE_AMAZON )

PokktNativeExtension.performOperation(PokktNativeExtension.TRACK_IAP, "" )
```

- Session

1. We have option to start session for tracking: startSession and endSession so call these like below:

```
PokktNativeExtension.performOperation(PokktNativeExtension.START_SESSION, "" )
PokktNativeExtension.performOperation(PokktNativeExtension.END_SESSION, "" )
```

2. You should call startSession at the start of his application if you want to use this but this is the optional and call it after setting application id and security key.
3. You should call endSession at the end of his application.

- OfferWall

1. For OfferWall, you can set two additional parameters which are *SET\_OFFERWALL\_ASSETS\_VALUE* and *SET\_CLOSE\_ON\_SUCCESS\_FLAG*. *SET\_OFFERWALL\_ASSETS\_VALUE* is required if you only want to show campaigns of certain value on OfferWall. So when you call 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 *InitPokkt*.
3. To show OfferWall please call

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

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 `onCoinResponseWithTrld` 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.

- AdConfig

1. There are few extra setting which need to do for caching/showing Ad. There are few constants are available which need to call using PerformOperation method like below: `PokktNativeExtension.performOperation(PokktNativeExtension.ADCONFIG_IS_SCREEN_NAME, "start")`. So set `ADCONFIG_SCREEN_NAME` and `ADCONFIG_IS_REWARDED`. This screen name will be created on pokkt dashboard.
2. developer can also `ADCONFIG_DEFAULT_SKIP_TIME`, `ADCONFIG_SKIP_CONFIRMATION_MESSAGE`, `ADCONFIG_BACK_BUTTON_DISABLED`, `ADCONFIG_SHOULD_ALLOW_MUTE`, `ADCONFIG_SHOULD_SKIP_CONFIRM`, `ADCONFIG_SKIP_CONFIRM_YES_LABEL`, `ADCONFIG_SKIP_CONFIRM_NO_LABEL`, `ADCONFIG_SKIP_TIMER_MESSAGE`, `ADCONFIG_INCENTIVE_MESSAGE`. These values can be used to configure the behaviour ad.
3. If you want to enable/disable the skip button on video screen please set `ADCONFIG_SHOULD_ALLOW_SKIP` as "true"/"false". The default value for set `ADCONFIG_SHOULD_ALLOW_SKIP` is "true"
4. If you have enabled skipped button by setting `ADCONFIG_SHOULD_ALLOW_SKIP` as true then you can control after how many seconds the skip button will be visible in video by setting `ADCONFIG_DEFAULT_SKIP_TIME` to appropriate value. Since most videos will be 30 sec or less please set `ADCONFIG_DEFAULT_SKIP_TIME` as 10 or less. You can also give your own skip message by setting `ADCONFIG_SKIP_CONFIRMATION_MESSAGE`.
5. The `ADCONFIG_SCREEN_NAME` has default value as `default` and can be used by you to give different screen name for different places in your app where you are showing ads. You will control ad targeting based on these screen names which should match exactly with screen names defined in dashboard. `ADCONFIG_SCREEN_NAME` can not contain white spaces and only special characters allowed are hyphen and underscore.
6. You can choose to show ad with or without incentive to user by setting `ADCONFIG_IS_REWARDED` as true or false. Ad gratification will only happen for

incentivized playback

7. You can disable the back button while video is playing by setting `ADCONFIG_BACK_BUTTON_DISABLED` by passing "true"/"false"
8. You can configure the ad skip dialog yes/no labels by setting `ADCONFIG_SKIP_CONFIRM_YES_LABEL` and `ADCONFIG_SKIP_CONFIRM_NO_LABEL`.
9. You can configure the ad incentive message by setting `ADCONFIG_INCENTIVE_MESSAGE`.
10. You can configure the ad skip timer message by setting `ADCONFIG_SKIP_TIMER_MESSAGE`. The message must contain a `##` placeholder to show skip time value, which will keep changing as per the time.

- [Rewarded/Non-Rewarded Ad](#)

1. You need to set "true"/"false" for rewarded or non-rewarded ad in `ADCONFIG_IS_REWARDED`.
2. You will have to call `PokktNativeExtension.performOperation ( PokktNativeExtension.CACHE_AD_CAMPAIGN, "" )` to start caching ads on device.
3. You can call `PokktNativeExtension.performOperation ( PokktNativeExtension.CHECK_AD_AVAILABILITY, "" )` to check if the campaign are available for a particular adConfig before you try to show ad.
4. You can call `PokktNativeExtension.performOperation ( PokktNativeExtension.SHOW_AD, "" )` to show ad.
5. Please reward user only from the `OnAdGratified` callback method.

- [Mediation Info](#)

Pokkt SDK now supports 10 ad networks which you can integrate in your application for better monetisation.

Please download mediation documentation from Pokkt and follow the process for each mediation network.

- [Export Logs](#)

1. Developer should call `PokktNativeExtension.performOperation ( PokktNativeExtension.EXPORT_LOG, "" )` to export the Pokkt SDK logs to folder of your choice.
2. This API shows a folder chooser dialog where user can choose a particular folder.

3. User can also create a new folder where user wants to export the logs.

- 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 testAllMethods in *AI\_MainMenu* AIModel.

- In-App Notifications

Developer can add In-App notifications in their dashboard.

Add Notification

**Basic**

Name

App

Platform

☐ iOS ☐ Android ☒ All

**Filters**

Countries

App Version

Last Seen

Min

Max

**Schedule**

Repeat

Dates

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time

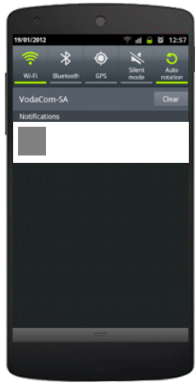
O'clock

**Message**

Message

Title

Add Image



Cancel

Save

Repeat schedule can be daily, weekly monthly.

Daily Repeat can have options like frequency of repeat and time in hours of notification.

The screenshot shows the 'Schedule' and 'Message' sections of the POKKT SDK V4.1.0 interface. In the 'Schedule' section, the 'Repeat' dropdown is set to 'Daily'. Below it, the 'Every' field is set to '1' with a unit of 'Day(s)'. The 'Time' field is set to '12' with a unit of 'O'clock'. In the 'Message' section, there is a 'Title' input field, a large text area for the message, and an 'Add Image' button. To the right of the message area is a preview of a smartphone screen showing the notification. At the bottom right, there are 'Cancel' and 'Save' buttons.

Weekly repeat can have options like frequency of repeat in weeks, days of repeat and time in hours of notification.

The screenshot shows the 'Schedule' and 'Message' sections of the POKKT SDK V4.1.0 interface. In the 'Schedule' section, the 'Repeat' dropdown is set to 'Weekly'. Below it, the 'Every' field is set to '1' with a unit of 'Week(s)'. There is a grid of days for selection: Mon, Tue, Wed, Thu, Fri, Sat, and Sun. The 'Time' field is set to '12' with a unit of 'O'clock'. In the 'Message' section, there is a 'Title' input field, a large text area for the message, and an 'Add Image' button. To the right of the message area is a preview of a smartphone screen showing the notification. At the bottom right, there are 'Cancel' and 'Save' buttons.

Monthly repeat can have options like frequency of repeat in months dates of repeat and time in hours for notification.

Repeat: Monthly

Every 1 Month(s)

Dates:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time: 12 O'clock

Message

Title

Add Image

Cancel Save

For don't repeat case, there are options like dates and time in hour for notification.

The notifications are listed and can be edited. Notifications can also be deactivated/activated

#### List Notifications

Id	Name	App Id	Header	Status	Action
1	push	1000125	Hi There	ACTIVE	<a href="#">Edit</a> <a href="#">Deactivate</a>
2	in app	1000125	Hi There	ACTIVE	<a href="#">Edit</a> <a href="#">Deactivate</a>

## 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.