

POKKT SDK v4.1.0 Integration Guide for ShiVa (iOS)

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 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 Ad). The current plugin supports mediation for various third party ad-networks.

These are:

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

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

Step 1:

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. **PokktSDKResource.bundle:** This is the resource file which is needed. You need to copy in your xcode project which gets exported from ShiVa tool.

Step 2:

If you are ready to build the XCode project using ShiVa Editor then please add below four header file under **Additional file** option. You will get these files in SDK folder.

1. PokktNetworkDelegate.h
2. PokktManager.h
3. NetworkModel.h
4. PokktConfig.h
5. InAppPurchaseRequest.h
6. AdNetwork.h
7. AdDelegate.h
8. AdConfig.h

Once your project is built and the XCode project is exported, open this exported project using XCode(It may open itself once you build the project from ShiVa editor). Ensure the followings frameworks are present(linked) inside project setting's "**Build-Phases -> Link Binaries With Libraries**", if not then add them manually:

- **CoreData.framework**
- **Foundation.framework**
- **MediaPlayer.framework**
- **SystemConfiguration.framework**
- **UIKit.framework**
- **CoreTelephony.framework**
- **EventKit.framework**
- **AdSupport.framework**

Step 3:

Ensure that "**-ObjC**" flag is set inside project setting's "**Build Settings -> Other Linker Flags**".

Step 4.

In order to use PokktSDK's background fetch functionality, enable "**Capabilities -> Background Modes -> Background Fetch**" inside project settings. Then write the following code-snippet inside "**didFinishLaunchingWithOptions**" method of app-delegate class (S3DEngine_AppDelegate.m). You will get this class once xcode project gets imported.

```
[application setMinimumBackgroundFetchInterval:
    UIApplicationBackgroundFetchIntervalMinimum];
```

Step 5:

In order to enable local notifications for **InApp Notifications**, mention the following inside "**didFinishLaunchingWithOptions**" method of the app-delegate class:

```
[application setMinimumBackgroundFetchInterval:
    UIApplicationBackgroundFetchIntervalMinimum];
UIUserNotificationSettings *settings = [UIUserNotificationSettings settingsForTypes:
    (UIRemoteNotificationTypeBadge | UIRemoteNotificationTypeSound
    UIRemoteNotificationTypeAlert)categories:nil];
[application registerUserNotificationSettings:settings];
```

Step 6.

Further, implement/update the background-fetch delegate methods in app-delegate class. Invoke "**callBackgroundTaskCompletionHandler**" method from "**performFetchWithCompletionHandler**". Observe the following code-snippet for reference:

```
-(void)application:(UIApplication *) application performFetchWithCompletionHandler:
(void(^)(UIBackgroundFetchResult))completionHandler
{
    [PokktManager callBackgroundTaskCompletionHandler:
        ^(UIBackgroundFetchResult result)
        {
            completionHandler(result);
        }];
}
```

Step 7.

Invoke "**inAppNotificationEvent**" if the user taps on local notification, do this in the "**didReceiveLocalNotification**" inside app-delegate class. Check the following reference:

```
-(void)application:(UIApplication *) application
didReceiveLocalNotification:(UILocalNotification *) notification
{
    [PokktManager inAppNotificationEvent:notification];
}
```

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 [iOS](#) as option and then export it. Then will generate .stk file.
- Now export xcode project from exported .stk file using ShiVa Authoring tools and do the all required setting for iOS in Settings option. There you need to assign path for Xcode Application.

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 xcode Project.

Once xcode project gets open, add resource.bundle file which will be downloaded folder.

- 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. 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")
5. 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")
6. 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.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.

- AdConfig

1. There are few extra setting which you can 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 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_CAMPAGN, "")` 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 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

Schedule

Repeat

Dates

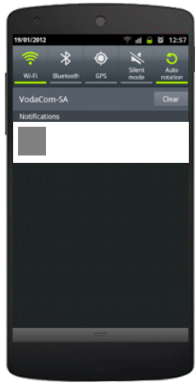
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time

O'clock

Message

Message



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 notification configuration interface. In the 'Schedule' section, the 'Repeat' dropdown is set to 'Daily'. Below it, 'Every' is set to '1' with a unit of 'Day(s)'. The 'Time' is set to '12' O'clock. The 'Message' section contains a 'Title' input field, a large text area for the message body, and an 'Add Image' button. To the right of the message input is a preview of a smartphone screen displaying 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 notification configuration interface for a weekly repeat. In the 'Schedule' section, the 'Repeat' dropdown is set to 'Weekly'. Below it, 'Every' 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' is set to '12' O'clock. The 'Message' section is identical to the daily repeat interface, with a 'Title' input, a message text area, an 'Add Image' button, and a smartphone preview. 'Cancel' and 'Save' buttons are at the bottom right.

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	Edit Deactivate
2	in app	1000125	Hi There	ACTIVE	Edit Deactivate

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.