

Pokkt iOS SDK Integration Guide

Please follow these steps as per your integration requirement

Configuration Steps

1. Please download the *ios_sdk_v4.1.0.zip* file and extract the downloaded file. The contents of downloaded folder should look like following

```
ios_sdk_v4.1.0
| --- Pokkt_iOS_SDK_Integration_Guide_4.1.0.pdf
| --- PokktLib-universal.zip
| --- SampleApp.zip
| --- SwiftSample.zip
```

2. Extract the *PokktLib-universal* file. The contents of folder should look like following

```
PokktLib-universal
| --- libPokktSDK.a
| --- PokktSDK
|         | --- AdNetwork.h
|         | --- AdConfig.h
|         | --- NetworkModel.h
|         | --- PokktConfig.h
|         | --- AdDelegate.h
|         | --- PokktManager.h
|         | --- PokktNetworkDelegate.h
|         | --- InAppPurchaseRequest.h
| --- PokktSDKResource.bundle
```

3. Go to your project's settings's "*Build Phases -> Link Binary with Libraries*" and add the *libPokktSDK.a*
4. Add all the *header files* to your application.
5. Make sure to add *PokktSDKResource.bundle* file to application

6. Please add following exceptions in your application *info.plist* file (please edit as source for this.)

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>pokkt.com</key>
    <dict>
      <key>NSIncludesSubdomains</key>
      <true/>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
      <key>NSExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSExceptionMinimumTLSVersion</key>
      <string>TLSv1.2</string>
      <key>NSThirdPartyExceptionAllowsInsecureHTTPLoads</key>
      <false/>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <true/>
      <key>NSThirdPartyExceptionMinimumTLSVersion</key>
      <string>TLSv1.2</string>
      <key>NSRequiresCertificateTransparency</key>
      <false/>
    </dict>
  </dict>
  <key>cloudfront.net</key>
  <dict>
    <key>NSIncludesSubdomains</key>
    <true/>
    <key>NSExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSExceptionRequiresForwardSecrecy</key>
    <false/>
    <key>NSExceptionMinimumTLSVersion</key>
    <string>TLSv1.2</string>
    <key>NSThirdPartyExceptionAllowsInsecureHTTPLoads</key>
    <false/>
    <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
    <true/>
    <key>NSThirdPartyExceptionMinimumTLSVersion</key>
    <string>TLSv1.2</string>
    <key>NSRequiresCertificateTransparency</key>
    <false/>
  </dict>
</dict>
</dict>
```

7. Your Project needs to have following frameworks to use PokktSDK

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

8. Please make sure that your app project has **-ObjC** set as *Other linker flag* in *Build Settings*.
9. Need to enable background fetch mode in Xcode for PokktSDK background fetch

Project Header -> Targets -> Capabilities -> Background Modes -> Enable Background fetch

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

(Write this in DidFinishLaunchingWithOptions delegate method)

10. Need to implement the background fetch delegate methods in *AppDelegate* class

```
-(void)application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler
```

11. Import the *PokktManager* Class in *AppDelegate* and call the *callBackgroundTaskCompletionHandler* method from *performFetchWithCompletionHandler* Method.

```
-(void)application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler{

    [PokktManager callBackgroundTaskCompletionHandler:^(UIBackgroundFetchResult result)
    {completionHandler(result);}];

}
```

12. Enable the local notification for InApp Notifications in *AppDelegate* class.

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

[application registerUserNotificationSettings:settings];
```

Note: Write this lines in DidFinishLaunchingWithOptions: delegate method in app delegate class

13. Implement LocalNotification delegate method in *AppDelegate* Class

```
-(void)application:(UIApplication *)application didReceiveLocalNotification:(UILocalNotification *)notification
```

14. You will have to call the *inAppNotificationEvent* Method, When user tap on local notification.

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

15. You should call the *notifyAppInstall* method from *PokktManager*, When your application launches for first time.

Implementation Steps

- Common

1. For all invocation of Pokkt SDK developer will make use of methods available in *PokktManager* class. This class only have static methods.
2. You will have to implement the *AdDelegate* protocol in your class to listen for all ad related events. Please set the the delegate in init method.
3. Before calling any other methods from the *PokktManager* please make sure that you have called the *initPokkt:withDelegate:* already (This does not apply to session related methods namely *startSession* and *endSession*).
4. For *initPokkt* method call *PokktConfig* instance and *AdDelegate* protocol implemented class instance is required. *PokktConfig* is plain NSObject object which will hold all the values required by the SDK which you need to assign.
5. In *PokktConfig* you can assign *applicationId* and *securityKey* which are must for all type of integrations.
6. *PokktConfig* also has provision for developer to provide multiple analytics trackers available with them to pokkt. Those are Google, Flurry and MixPanel. Which need the trackerID. Currently supported : *googleTrackerID*, *flurryTrackerID* and *mixPanelTrackerID*.
7. *PokktConfig* providing the event type for the type of analytic used in application side . Which is *eventType* (Event types are provided in *PokktConfig* class only)
8. If you are doing server to server integration with pokkt you can also mention *thirdPartyUserId* in *PokktConfig*.
9. Apart from above mentioned parameters you can assign additional ones based on your integration type.(please refer to Ad sections below.)
10. While in development please call *[PokktManager setDebug:YES];* to see pokkt debug logs and toast messages. please make sure to change this to *[PokktManager setDebug:NO];* for production build.
11. Please call *[PokktManager.trackIAP: InAppPurchaseRequest]* to send any in-app purchase information to Pokkt.
12. For almost all ad related methods call, *AdConfig* instance is required. *AdConfig* is NSObject object which will hold all the values required by the SDK which you need to assign for getting an Ad.

- Session

1. Starting with this version Pokkt SDK is adding session tracking for which we have *startSession* and *endSession* methods in *PokktManager*.
2. You should call *startSession* at the start of your application and once only.
3. You should call *endSession* at the end of your application and once only.

- AdConfig

1. In *AdConfig* you can assign *screenName*, *isRewarded*, *shouldAllowSkip*, *defaultSkipTime*, *skipConfirmMessage*, *shouldAllowMute*, *shouldSkipConfirm*, *skipConfirmYesLabel*, *skipConfirmNoLabel*, *skipTimerMessage* and *incentiveMessage* . These values can be used to configure the behaviour of ad.
2. If you want to enable/disable the skip button on video screen please set *shouldAllowSkip* as true/false. The default value for *shouldAllowSkip* is true.
3. If you have enabled skipped button by setting *shouldAllowSkip* as true then you can control after how many seconds the skip button will be visible in video by setting *defaultSkipTime* to appropriate value. Since most videos will be 30 sec or less please set *defaultSkipTime* as 10 or less. You can also give your own skip message by setting *skipConfirmMessage* on *AdConfig*
4. *screenName* 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. ScreenName can not contain white spaces and only special characters allowed are hyphen and underscore.
5. You can choose to show ad with or without incentive to user by setting *isRewarded* as true or false. Video gratification will only happen for incentivised playback.
6. You can configure the ad skip dialog yes/no labels by setting *skipConfirmYesLabel* and *skipConfirmNoLabel*.
7. You can configure the ad incentive message by setting *incentiveMessage*.

8. You can configure the ad skip timer message by setting *skipTimerMessage*. 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 will have to call *[PokktManager.cacheAd: (AdConfig *) adConfig]*; to start caching ads on device.
2. You will need to create *AdDelegate* implementation class as mentioned in [step 2](#) in [implementation steps](#).
3. You can call *[PokktManager checkAdAvailability: (AdConfig *)adConfig]* to check if the campaign are available for a particular adConfig before you try to show ad.
4. You can call *[PokktManager.showAd: (AdConfig *)adConfig]*; to show ad.
5. You will get different callbacks as given in *AdDelegate* implementation class for ad display.
6. Please reward user only from the *onAdGratified* method in *AdDelegate* implementation class.

Export Logs

1. Developer should call *[PokktManager exportLog: (UIViewController *) viewController]* to export the Pokkt SDK logs to some folder.
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

- *PokktConfig* also has 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*.

Mediation Info

- Pokkt SDK now supports 12 ad networks which you can integrate in your application for better monetisation.
- To integrate these networks through Pokkt, please visit the mediation menu on downloads page and download *native mediation zip* and *documentation zip* files.
- Please follow the mediation integration documents shipped for each network.
- In *native mediation zip* we have provided sample application project as an example integrating all these networks.
- You will need to create account on these networks and add the network details in your Pokkt dashboard after login into your account on pokkt website.
- You will also need to do the mapping of Pokkt screens with the corresponding ad networks' placement id/zone id/ad unit etc in the dashboard.

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.