

Pokkt IOS SDK Integration Guide

Overview

Pokkt IOS SDK v3.1 supports Video ad campaigns feature. With help of this document, any application developer / publisher can integrate either feature or both the features in their application. Please follow these steps as per your integration requirement (Video).

Configuration Steps

1. Add the libPokktSDK.a to your project's lib directory.
Project Header -> Targets -> Build Phase -> Link Binary With Libraries -> Plus -> Others -> Choose libPokktSDK.a
2. Add SDK Public header files to application. (Drag and drop the all .h files to application.)
3. Add PokktSDKResource.bundle file to application.
4. 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>

```

5. Import PokktController class, Then you will have to implement the PokktDelegate Delegate methods in your project to listen for all video related events. (refer to sample app for example)

```
[PokktManager setPokktDelegate:self];
```

6. Import PokktController class, Then you will have to pass the current view controller instance to the SDK to present the video

```
[PokktManager setPresentView:self];
```

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

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 DidFinishLaunchWithOptions delegate method)
```

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

```

-(void)application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)(
    UIBackgroundFetchResult))completionHandle

```

11. Import the PokktManager Class in AppDelegate. Then You will have to 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:(UIRemoteNotificationTypeBadel
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. PokktManager class provide the `LogExport` method to get log file.

```

+ (void)exportPokktSDKLog

```

16. You will have to call the `appInstallFirstTime` method, When application launch first.

```

+ (void)appInstallFirstTime

```

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. Before calling any other methods from the *PokktManager* please make sure that you have called the *initPokkt* already. (This does not apply to session related methods namely *startSession* and *endSession*)
3. For almost all methods call *PokktConfig* instance is required. *PokktConfig* is plain NSObject object which will hold all the values required by the SDK which you need to assign.

4. In *PokktConfig* you can assign *applicationId* and *securityKey* which are must for all type of integrations.
5. If you are doing server to server integration with pokkt you can also mention *thirdPartyUserId* in *PokktConfig*.
6. Apart from above mentioned parameters you can assign additional ones based on your integration type.(please refer to Video sections below.)
7. 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.

- 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 his application and once only. You will need to provide pokktConfig instance for this method with *applicationId* and *securityKey* assigned.
3. You should call *endSession* at the end of his application and once only.

- Video

1. In *PokktConfig* for Video you can set five additional parameters which are *autoCacheVideo*, *skipEnabled*, *defaultSkipTime*, *screenName* and *incentivised*.
2. *autoCacheVideo* is required if you want to automatically cache video on user device. It has default value as true. if you set it as false then video will not be automatically cached and you will have to call *[PokktManager cacheVideoCampaign:pokktConfig];* to start caching videos on device.
3. If you want to enable/disable the skip button on video screen please set *skipEnabled* as true/false. The default value for *skipEnabled* is false.
4. If you have enabled skipped button by setting *skipEnabled* 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 *customSkipMessage* on *PokktConfig*
5. *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 video 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.

6. You can choose to show video with or without incentive to user by setting *incentivised* as true or false. Video gratification will only happen for incentivised playback.
7. You will need to create *PokktDelegate* implementation class as mentioned in step 4 in configuration steps.
8. You can call *[PokktManager isVideoAvailable]* to check if the campaign are available before you try to play video.
9. You can call *[PokktManager playVideoCampaign:pokktConfig];* to play video.
10. You will get different callbacks as given in *PokktDelegate* implementation class for video playback.
11. Please reward user only from the *onVideoGratified* method in *PokktDelegate* implementation class.

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*.
- *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*.
- *PokktConfig* providing the eventtype for the type of analytic used in application side . Which is *eventType* (Event types are provided in PokktConfig class only)

- 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

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

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.

Schedule

Repeat

Daily

Every

1

Day(s)

Time

12


O'clock

Message

Message

Title

Add Image



Cancel

Save

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

Schedule

Repeat

Weekly

Every

1

Week(s)

Mon

Tue

Wed

Thu

Fri

Sat

Sun

Time

12


O'clock

Message

Message

Title

Add Image



Cancel

Save

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

Repeat

Every

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

O'clock

Message

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.