



## POKKT SDK Integration Guide (v 2.0)

iOS

<b>Overview</b>	<b>1</b>
<b>Project Configuration</b>	<b>2</b>
Dependencies	2
Frameworks Required	2
Info.plist	2
Other Settings	4
<b>Implementation Steps</b>	<b>4</b>
SDK Configuration	4
Ad Types	5
Video	5
Rewarded	5
Non Rewarded	5
Ad Delegates	5
Video	5
Pokkt ad player configuration	6
User Details	9
Debugging	10
Analytics	11
Google Analytics	11
Flurry Analytics	11
MixPanel Analytics	11
Fabric Analytics	11
IAP(In App Purchase)	12
<b>Limitation:</b>	<b>13</b>

### Overview

Thank you for choosing **Pokkt SDK** for **iOS**. This document contains all the information required to set up the SDK with your project.

Before implementing it is mandatory to go through [project configuration](#) and [implementation steps](#), as these sections contain mandatory steps for basic SDK integration.

You can download our SDK from [pokkt.com](https://pokkt.com).



In the package downloaded above you will find:

1. Docs:  
Contains documentations for step wise step integration for SDK.
2. PokktAds Demo  
Source code for *PokktAds Demo*(Sample app) which showcase implementation of Pokkt SDK through code for better understanding.
3. PokktSDK:  
*PokktSDK.framework* add in to your project

**ScreenName:** This one parameter is accepted by almost all API's of Pokkt SDK. This controls the placement of ads and can be created on Pokkt Dashboard.

We will be referencing **PokktAds Demo** app provided with SDK during the course of explanation in this document. We suggest you go through the sample app for better understanding.

## Project Configuration

### Dependencies

- Add *PokktSDK.framework* to your project's settings at "Build Phases -> Link Binary with Libraries and add the PokktSDK.framework.

### Frameworks Required

```
CoreData.framework
Foundation.framework
MediaPlayer.framework
SystemConfiguration.framework
UIKit.framework
CoreTelephony.framework
EventKit.framework
AdSupport.framework
CoreGraphics.framework
CoreMotion.framework
MessageUI.framework
EventKitUI.framework
CoreLocation.framework
AVFoundation.framework
libc++.tbd
```

### Info.plist

Add the below exceptions to your application's info.plist.

```

<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>
  <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>

```

## Other Settings

- Please make sure that your app project has **-ObjC** set as Other linker flag in Build Settings.

## Implementation Steps

### SDK Configuration

1. Set **Application Id** and **Security key** in Pokkt SDK. You can get it from Pokkt dashboard from your account. We generally assign unique application Id and Security key.

```
PokktAds.setPokktConfigWithAppId:(NSString*) appId securityKey:(NSString*) securityKey
```

2. If you are using server to server integration with Pokkt, you can also set **Third Party UserId** in PokktAds.

```
PokktAds.setThirdPartyUserId:(NSString*) userId
```

3. When your application is under development and if you want to see Pokkt logs and other informatory messages, you can enable it by setting **ShouldDebug** to **true**. Make sure to disable debugging before release.

```
[PokktDebugger setDebug: YES/ NO];
```

## Ad Types

### Video

- Video ad can be rewarded or non-rewarded. You can directly call show for it.
- it will stream the video which may lead to buffering delays depending on the network connection.

### Rewarded

1. To show rewarded ad call:

```
[PokktVideoAds showRewardedVideoAd:(NSString*) screenName  
viewController:(UIViewController *)viewController];
```

### Non Rewarded

1. To show non-rewarded ad call:

```
[PokktVideoAds showNonRewardedVideoAd:(NSString*) screenName  
viewController:(UIViewController *)viewController];
```

## Ad Delegates

Ad actions are optional, but we suggest to implement them as it will help you to keep track of the status of your ad request.

### Video

```
[PokktVideoAdsDelegate setPokktVideoAdsDelegate:self];
```

## Pokkt ad player configuration

Pokkt Ad player works the way App is configured at Pokkt dashboard, but we provide a way to override those settings using **PokktAdPlayerViewConfig**.

Application should prefer configuration provided through code by developer or what's configured for the app in dashboard, can be controlled any time through the dashboard itself. If you want to make changes to this configuration after your app distribution, you can contact **Pokkt Team** to do the same for your app through admin console.

```
PokktAdPlayerViewConfig * adPlayerViewConfig = [[PokktAdPlayerViewConfig alloc] init];
// set properties values to adPlayerViewConfig
PokktAds.setAdPlayerViewConfig(adPlayerViewConfig );
```

Various properties that can be managed through this are:

**1. Default skip time**

Defines the time after which user can skip the Ad.

**Property name:** DefaultSkipTime

**Values:**

Any Integer value.

Default value is 10 seconds .

**2. Should allow skip**

Defines if user is allowed to skip the Ad or not.

**Property name:** ShouldAllowSkip

**Values:**

True = User can skip Ad.

False = User can't skip Ad.

**3. Should allow mute**

Defines if user is allowed to mute the Video Ad or not.

**Property name:** ShouldAllowMute

**Values:**

True = User can mute video Ad.

False = User can't mute video Ad.

**4. Should confirm skip**

Defines if confirmation dialog is to be shown before skipping the Ad.

**Property name:** ShouldConfirmSkip

**Values:**

True = Confirmation dialog will be shown before skipping the video.

False = Confirmation dialog will not be shown before skipping the video.

**5. Skip confirmation message**

Defines what confirmation message to be shown in skip dialog.

**Property name:** SkipConfirmMessage

**Values:**

Any String message.

Default value is "Skipping this video will earn you NO rewards. Are you sure?".

**6. Affirmative label for skip dialog**

Defines what should be the label for affirmative button in skip dialog.

**Property name:** SkipConfirmYesLabel

**Values:**

Any String message.  
Default value is "Yes".

**7. Negative label for skip dialog**

Defines what should be the label for affirmative button in skip dialog.

**Property name:** SkipConfirmNoLabel

**Values:**

Any String message.  
Default value is "No".

**8. Skip timer message**

Defines message to be shown before enabling skip button. Don't forget to add placeholder "##" in your custom message.

This placeholder is replaced by property "Default skip time" assigned above.

**Property name:** SkipTimerMessage

**Values:**

Any String message.  
Default value is "You can skip video in ## seconds"

**9. Incentive message**

Defines message to be shown during video progress, that after what time user will be incentivised.

**Property name:** IncentiveMessage

**Values:**

Any String message  
Default value is "more seconds only for your reward !"

**10. Should collect feedback**

Defines message to be shown during video progress, that after what time user will be incentivised.

**Property name:** setShouldCollectFeedback

**Values:**

True = If you want to collect feedback from the user for the Ad.  
False = If you don't want to collect feedback from the user for the Ad.

## User Details

For better targeting of ads you can also provide user details to our SDK using.

```
PokktUserDetails *pokktUserDetails = [PokktUserDetails alloc] init];
pokktUserDetails.Name = "";
pokktUserDetails.Age = "";
pokktUserDetails.Sex = "";
pokktUserDetails.MobileNo = "";
pokktUserDetails.EmailAddress = "";
pokktUserDetails.Location = "";
pokktUserDetails.Birthday = "";
pokktUserDetails.MaritalStatus = "";
pokktUserDetails.FacebookId = "";
pokktUserDetails.TwitterHandle = "";
pokktUserDetails.Education = "";
pokktUserDetails.Nationality = "";
pokktUserDetails.Employment = "";
pokktUserDetails.MaturityRating = "";

[PokktAds setPokktUserDetails: pokktUserDetails]
```

## Debugging

Other than enabling debugging for Pokkt SDK, it can also be used to:



## Analytics

We support various analytics in Pokkt SDK.

Below is mentioned how to enable various analytics with Pokkt SDK.

### Google Analytics

Google analytics Id can be obtained from Google dashboard.

```
PokktAnalyticsDetails *analyticsDetail = [[PokktAnalyticsDetails alloc] init];
analyticsDetail.eventType = GOOGLE_ANALYTICS;
analyticsDetail.googleTrackerID = @"xyz";
[PokktAds setPokktAnalyticsDetail:analyticsDetail];
```

### Flurry Analytics

Flurry application key can be obtained from Flurry dashboard.

```
PokktAnalyticsDetails *analyticsDetail = [[PokktAnalyticsDetails alloc] init];
analyticsDetail.eventType = FLURRY_ANALYTICS;
analyticsDetail.flurryTrackerID = @"xyz";
[PokktAds setPokktAnalyticsDetail:analyticsDetail];
```

### MixPanel Analytics

MixPanel project token can be obtained from MixPanel dashboard.

```
PokktAnalyticsDetails *analyticsDetail = [[PokktAnalyticsDetails alloc] init];
analyticsDetail.eventType = MIXPANNEL_ANALYTICS;
analyticsDetail.mixPanelTrackerID = @"xyz";
[PokktAds setPokktAnalyticsDetail:analyticsDetail];
```

### Fabric Analytics

Analytics Id is not required in case of Fabric.

```
PokktAnalyticsDetails *analyticsDetail = [[PokktAnalyticsDetails alloc] init];
analyticsDetail.eventType = FABRIC_ANALYTICS;
analyticsDetail.fabricTrackerID = @"xyz";
[PokktAds setPokktAnalyticsDetail:analyticsDetail];
```

#### IAP(In App Purchase)

Call trackIAP to send any In App purchase information to Pokkt.

```
InAppPurchaseDetails * inAppPurchaseDetails = [ InAppPurchaseDetails alloc]init];
inAppPurchaseDetails.ProductId = "<productId>";
inAppPurchaseDetails.PurchaseData = "<purchaseData>";
inAppPurchaseDetails.PurchaseSignature = "<purchaseSignature>";
inAppPurchaseDetails.PurchaseStore = IAPStoreType.GOOGLE;
inAppPurchaseDetails.Price = <100.00>;

[PokktAds trackIAP: inAppPurchaseDetails]
```

#### Limitation:

- Any web-enabled ad content is not supported, such as MRAID/VPAID/HTML ads etc.
- Currently caching for video/images is not possible.
- There is no close-button for video, start & end cards.
- No mute-button for video ads.