POKKT SDK Integration Guide (v 7.6.0)

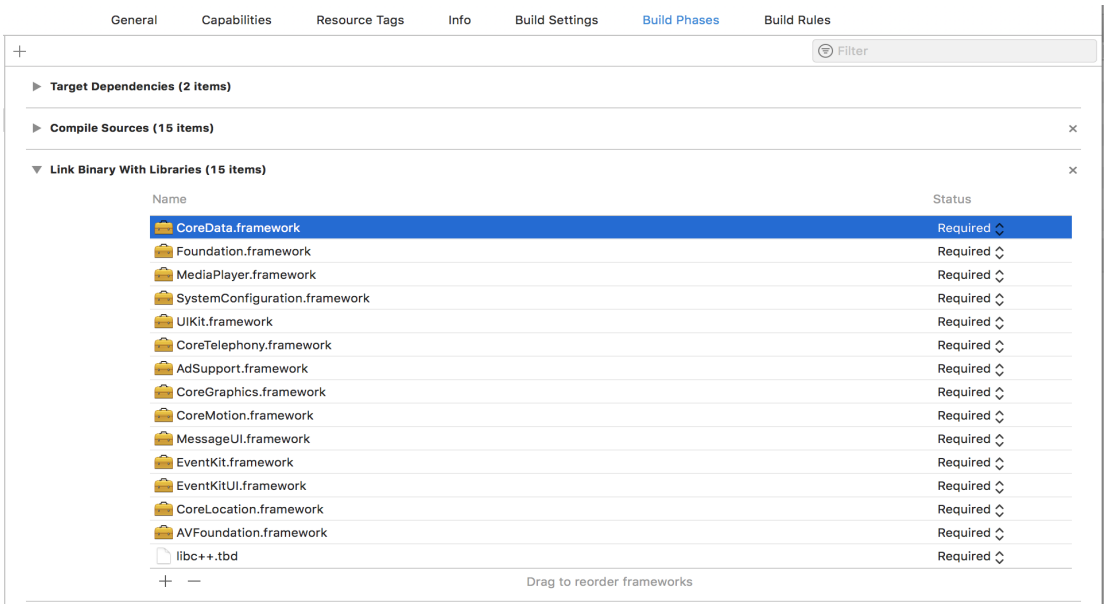
Contents 1
SDK Integration 1
SDK Configuration 2
Video Ads 2
Interstitial Ads 4
Banner Ads 5
OutStream Ads 5
Additional SDK Configuration 7
GDPR 7
Extra Parameter 7
ThirdPartyUser Id 8
Pokkt AdPlayer Configuration 9
Analytics 11
In App Notification 12
Other Settings 13

Contents

SDK Integration

- Download the latest SDK. [LINK]
- Unzip the downloaded file and drag the PokktSDK.framework directory into Xcode under Framework.
- *Add the required compiler flags to 'Other Linker Flags' field in your project's Build Settings.*
    - *-ObjC*
    - *-lxml2*

- Make sure that these frameworks are included to respect the library dependencies:
    - Add Webkit.Framework.



- 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. 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];
```

***For More SDK Configuration. See Section More Settings**

- Video Ads

1. Video ad can be rewarded or non-rewarded. You can either cache the ad in advance or directly call show for it.
2. We suggest you to cache the ad in advance so as to give seamless play behaviour, In other case it will stream the video which may lead to unnecessary buffering delays depending on the network connection.
3. To cache Video ad call:

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

```
[PokktVideoAds cacheRewardedVideoAd: (NSString*) screenName
```

```
[PokktVideoAds cacheNonRewardedVideoAd: (NSString*) screenName];
```

4. To Show Video Ad:

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

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

You can check if ad is available or not before making *show* request.

```
[PokktVideoAds isAdCached:(NSString *)screenName isRewarded:(BOOL)isRewarded];
```

5. 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.

```
- (void)videoAdCachingCompleted: (NSString *)screenName isRewarded: (BOOL)isRewarded reward: (float)reward;

- (void)videoAdCachingFailed: (NSString *)screenName isRewarded: (BOOL)isRewarded errorMessage: (NSString *)errorMessage;

- (void)videoAdCompleted: (NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)videoAdDisplayed: (NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)videoAdClicked: (NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)videoAdGratified: (NSString *)screenName reward:(float)reward;

- (void)videoAdSkipped: (NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)videoAdClosed:(NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)videoAdFailedToShow: (NSString *)screenName isRewarded: (BOOL)isRewarded errorMessage: (NSString *)errorMessage;
```

- # Interstitial Ads

1. To Cache Interstitial ad call:

```
[PokktInterstitial setPokktInterstitialDelegate:self];;
```

```
[PokktInterstitial cacheRewarded (NSString*) screenName];
```

```
[PokktInterstitial cacheNonRewarded: (NSString*) screenName];
```

2. To Show Interstitial ad call:

```
[PokktInterstitial showRewarded :(NSString*) screenName viewController:(UIViewController *)viewController];
```

```
[PokktInterstitial  showNonRewarded :(NSString*) screenName viewController:(UIViewController *)viewController];
```

You can check if ad is available or not before making *show* request.

```
[PokktInterstitial
 isAdCached:(NSString *)screenName isRewarded:(BOOL)isRewarded];
```

3. 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.

```
- (void)interstitialCachingCompleted: (NSString *)screenName isRewarded: (BOOL)isRewarded reward: (float)reward;

- (void)interstitialCachingFailed: (NSString *)screenName isRewarded: (BOOL)isRewarded errorMessage: (NSString *)errorMessage;

- (void)interstitialCompleted: (NSString *)screenName isRewarded: (BOOL)isRewarded; //TODO: Requirement check

- (void)interstitialDisplayed: (NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)interstitialClicked: (NSString *)screenName isRewarded: (BOOL)isRewarded;

- (void)interstitialGratified: (NSString *)screenName reward:(float)reward;

- (void)interstitialSkipped: (NSString *)screenName isRewarded: (BOOL)isRewarded; //TODO: Requirement check

- (void)interstitialClosed:(NSString *)screenName isRewarded: (BOOL)isRewarded;
```

```
- (void)interstitialFailedToShow: (NSString *)screenName isRewarded: (BOOL)isRewarded errorMessage: (NSString *)errorMessage;
```

- Banner Ads

1. Load banner Ad call:

```
PokktBannerView  *banner = [PokktBanner initWithBannerAdSize:CGRectMake(0, 0, [UIScreen mainScreen].bounds.size.width, 50)];
[self.view addSubview:banner];

[PokktBanner loadBanner:banner withScreenName:@"default" rootViewContorller:self];
```

2. Refresh banner:

   You can set banner refresh rate on pokkt dashboard. Refresh rate should be in range of 30 - 100

3. Destroy banner

```
[PokktBanner destroyBanner:<PokktBannerView>];
```

4. 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.

```
- (void)bannerLoaded:(NSString *)screenName;

- (void)bannerClicked:(NSString *)screenName;

- (void)bannerLoadFailed:(NSString *)screenName errorMessage:(NSString *)errorMessage
```

- OutStream Ads

1. OutStream ad will be non-rewarded only.
2. Ad will play only if ad is visible in bound, if it is not visible on the screen it will in pause state.
3. OutStream Ad contain 3 states as Play, Pause & Replay. After completing the video Replay button will be enable.
4. If You are requesting for full screen ad then you have to destroy OutStream ad.
5. OutStream ad allow only for following format.
   - ScrollView

1. Load OutStream Ad Call:

```
+(void)loadOutstreamAd:(NSString*) screenName placeHolder:(UIView *)container scrollView:(UIScrollView *)scrollView
```

2. Destroy OutStream Ad Call:

```
+(void)disappearScrollViewOutStreamAd:(NSString *)screenName inContainer:(UIView *)container
```

- WebView
    1. Load Ad:

```
+(void)loadOutstreamAd:(NSString*) screenName inWebView:(UIWebView *)webView
```

    2. Destroy Ad

```
+(void)disappearWebViewOutStreamAd:(NSString *)screenName inContainer:(UIWebView *)webView
```

- Tableview
    1. Load Ad

```
+(void)loadOutstreamAd:(NSString*) screenName atIndexPath:(NSIndexPath *)indexPath tableView:(UITableView *)tableView
```

    2. Destroy Ad

```
+(void)disappearTableViewOutStreamAd:(NSString *)screenName inContainer:(UITableView *)tableView
```

6. 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.

```
- (void)outStreamAdCompleted: (NSString *)screenName;

- (void)outStreamAdDisplayed: (NSString *)screenName;

- (void)outStreamAdReady:(NSString *)screenName;

- (void)outStreamAdClicked:(NSString *)screenName;

- (void)outStreamAdFailedToShow: (NSString *)screenName errorMessage: (NSString *)errorMessage;
```

Additional SDK Configuration

GDPR

As of May 25th, the General Data Protection Regulation (GDPR) will be enforced in the European Union.
Set ***GDPR consent*** in Pokkt SDK. **This must be called before calling any ad related API. Developers/Publishers must get the consent from user.** For more information on GDPR please refer https://www.eugdpr.org/ and https://www.eugdpr.org/gdpr-faqs.html. This API can again be used by publishers to revoke the consent. If this API is not called or invalid data provided then SDK will access the users personal data for ad targeting

```
PokktConsentInfo *consentInfo = [[PokktConsentInfo alloc] init];

consentInfo.isGDPRApplicable = true;

consentInfo.isGDPRConsentAvailable = false;

[PokktAds setPokktConsentInfo:consentInfo];
```

Extra Parameter

You can set extra parameters to POKKT SDK, to be passed back to your server via POKKT server callback. These Extra parameters will be in key-value pair.The key must be alphanumeric value. See the below example

```
NSMutableDictionary *dict = [[NSMutableDictionary alloc] initWithCapacity:0];
[dict setValue:@"value1" forKey:@"key1"];
[dict setValue:@"value2" forKey:@"key2"];
[dict setValue:@"value3" forKey:@"key3"];
[PokktAds setCallbackExtraParam:dict];
```

- ## ThirdPartyUser Id

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

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

- ## User Detail

  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]
```

- ## Export Log

  Export your log to your desired location, we generally have it in root directory of SD card, if permission for external storage is provided and in cache folder otherwise.

```
[PokktDebugger exportLog: (UIVIewController *)controller
```

- ## 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]
```

Pokkt AdPlayer 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. AudioEnabled
AudioEnabled set YES or NO to mute ad,if 'YES' ad will be mute. , default is NO**.**
***Property name*:** isAudioEnabled
***Values*:**
True = If you want to mute Ad.
False = If you don’t want to mute Ad.

11. 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.

- # Analytics

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

1. 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];
```

2. 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];
```

3. 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];
```

4. Fabrics Analytics
   Analytics Id is not required in case of Fabric.

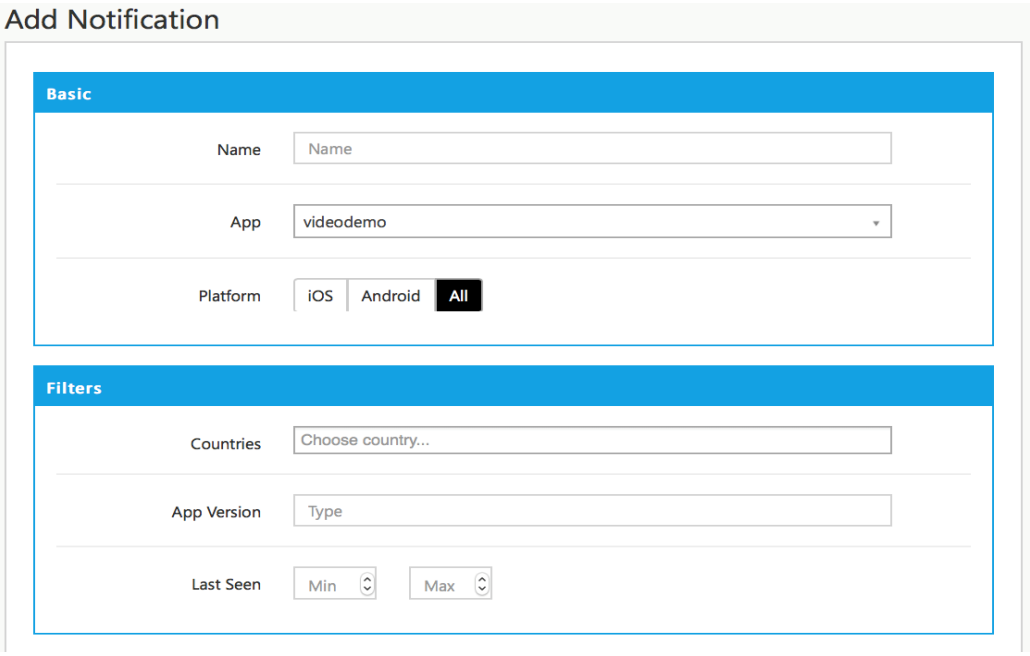
```
 PokktAnalyticsDetails *analyticsDetail = [[PokktAnalyticsDetails alloc] init];
analyticsDetail.eventType = FABRIC_ANALYTICS;
analyticsDetail.fabricTrackerID = @"xyz";
[PokktAds setPokktAnalyticsDetail:analyticsDetail];
```

In App Notification

Open developer dashboard -> Manage App -> Notifications.

Create Notification

1. Basic notification information required.



- Name
  A friendly name for notification. It will help you to distinguish between different notifications
- App
  Select your app for which you want to assign notifications
- Countries
  Select countries where this notifications will be shown to users. Let's say if you have users in multiple countries, you can selectively target notifications to them.
- App version
  Enter your app version for which you want to show notifications. Let's say you have multiple version installed among users and you want to send different notifications to different users based on their versions
- Last seen
  Set minimum and maximum limit in days for which user can remain away from the app. Let's say if min = 2 and max = 4, and user hasn't open your app for atleast 2 days, you can remind by showing notification, but if 4 days have passed app will not show any more notifications to user.
- Message
  Message you want to show in notifications bar
- Title
  Title for the notification to be shown in notification bar

2. Schedule Notification
   You can schedule notification for daily, weekly and monthly.

Other Settings

- Need to enable background fetch mode in Xcode at
  Project Header **->** Targets **->** Capabilities **->** Background Modes **->** Enable Background fetch.
  Now add below mentioned mentioned script at *DidFinishLaunchWithOption* **AppDelegate** method.

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

- Enable the local notification for *InApp Notifications* in **AppDelegate** class.
  Add below mentioned script at *DidFinishLaunchingWithOptions* delegate method **AppDelegate** class.

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

- You will have to call the inAppNotificationEvent Method, When user tap on local notification

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

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

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