

# Pokkt SDK for tvOS (v1.0) Integration Guide

---

Please follow these steps as per your integration requirement

## Configuration Steps

1. Extract the **downloaded** file. The contents of downloaded folder should look like following  
tvOS\_sdk\_v1.0
  - | --- Pokkt\_SDK\_tvOS\_Integration\_Guide\_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
  - | --- PokktSDK.framework
  - | --- PokktSDK.bundle
3. Go to your project's settings's "**Build Phases -> Link Binary with Libraries** and add the **PokktSDK.framework**
4. Make sure to add **PokktSDK.bundle** file to application
5. 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>
    <key>cloudfront.net</key>
    <dict>
        <key>NSIncludesSubdomains</key>
        <true/>
        <key>NSEnvironmentAllowsInsecureHTTPLoads</key>
        <true/>
        <key>NSEnvironmentRequiresForwardSecrecy</key>
        <false/>
        <key>NSEnvironmentMinimumTLSVersion</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>

```

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

- Foundation.framework
- MediaPlayer.framework
- SystemConfiguration.framework
- UIKit.framework
- AdSupport.framework
- CoreGraphics.framework
- AVFoundation.framework
- libc++.tbd
- CloudKit.Framework
- AVKit.Framework

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

8. 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)

9. Need to enable iCloud in Xcode for send log on iCloud.

**Project Header -> Targets -> Capabilities -> iCloud Modes -> Enable iCloud**

10. You will have to call the `notifyAppInstall` method, When application launch first.

+ (void) *notifyAppInstall*

## 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 ***PokktInitDelegate*** protocol in your class to know PokktSDK initialisation has been succeed or failed.
3. You will have to implement the ***AdDelegate*** protocol in your class to listen for all ad (**Video, Interstitial**) related events.
4. Before Calling PokktManager init method it's recommended to **set PokktInitDelegate** to know PokktInit Succeeded or failed
5. Before calling any other methods from the ***PokktManager*** please make sure that you have called the ***initPokkt*** already
6. For ***initPokkt*** method call ***PokktConfig*** instance. ***PokktConfig*** is plain NSObject object which will hold all the values required by the SDK which you need to assign.
7. In ***PokktConfig*** you can assign ***applicationId*** and ***securityKey*** which are must for all type of integrations.
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. To export log you need to enable iCloud from capabilities in general setting.

12. Please call [\[PokktManager.trackIAP: InAppPurchaseDetails\]](#) to send any in-app purchase information to Pokkt.
13. 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.
14. 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.

- [AdConfig](#)

1. In [AdConfig](#) you can assign [screenName](#), [isRewarded](#), [shouldAllowSkip](#), [adFormat](#), [defaultSkipTime](#), [skipConfirmMessage](#), [shouldAllowMute](#), [shouldSkipConfirm](#), [skipConfirmYesLabel](#), [skipConfirmNoLabel](#), [skipTimerMessage](#) and [incentiveMessage](#) . These values can be used to configure the behaviour of ad.
2. For getting specific type of ad set adFormat in adConfig. AdFormat is a type for ad such as 0:video , 3: interstitial , default is 0
3. 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.
4. 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](#)
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 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 ad with or without incentive to user by setting [isRewarded](#) as true or false. Video gratification will only happen for incentivised playback.

7. You can configure the ad skip dialog yes/no labels by setting *skipConfirmYesLabel* and *skipConfirmNoLabel*.
8. You can configure the ad incentive message by setting *incentiveMessage*.
9. 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.

- Ad (Video and Interstitial)

1. Before calling show method you need to create *AdDelegate* implementation class as mentioned in [step 3](#) in [implementation steps](#). After that set adDelegate by calling *[PokktManager setAdDelegate:];*
2. You can call *[PokktManager checkAdAvailability: (AdConfig \*)adConfig]* to check if the campaign are available for a particular adConfig before you try to show ad.
3. You can call *[PokktManager showAd:(AdConfig \*)adConfig viewController: (UIViewController \*)viewController];* to show ad.
4. You will get different callbacks as given in *AdDelegate* implementation class for ad display.
5. Please reward user only from the *onAdGratified* method in *AdDelegate* 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)

### 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.

