
POKKT SDK v2.0.8 Integration Guide for Unity3d 4.x (iOS)

Contents:

1. Introduction
2. Installation
3. PokktManagerSecurity.plist Setup
4. SDK Setup on Unity3d
5. Video Ad Functionalities
6. Debugging and Logging

1. Introduction:

Thank you for choosing Pokkt SDK for Unity3d. This document contains all the information that is needed by you to setup the SDK with your project. Kindly note that these instructions are for Unity3d Version 4.x and above, older versions of Unity3d are not supported at this moment.

There is a sample app provided with the SDK. We will be referencing this app during the course of explanation in this document. It is suggested that you should check that app to understand the following process in detail.

2. Installation:

All we need is the file provided: PokktUnityDemo_v2.X.X.unitypackage

Export this package into your unity app/game. The contents of “Script/Pokkt” directory and the Plugins/iOS deirectory are mandatory. You can choose to ignore all the other stuff, but for the sake of this document, it is recommended that you should export everything.

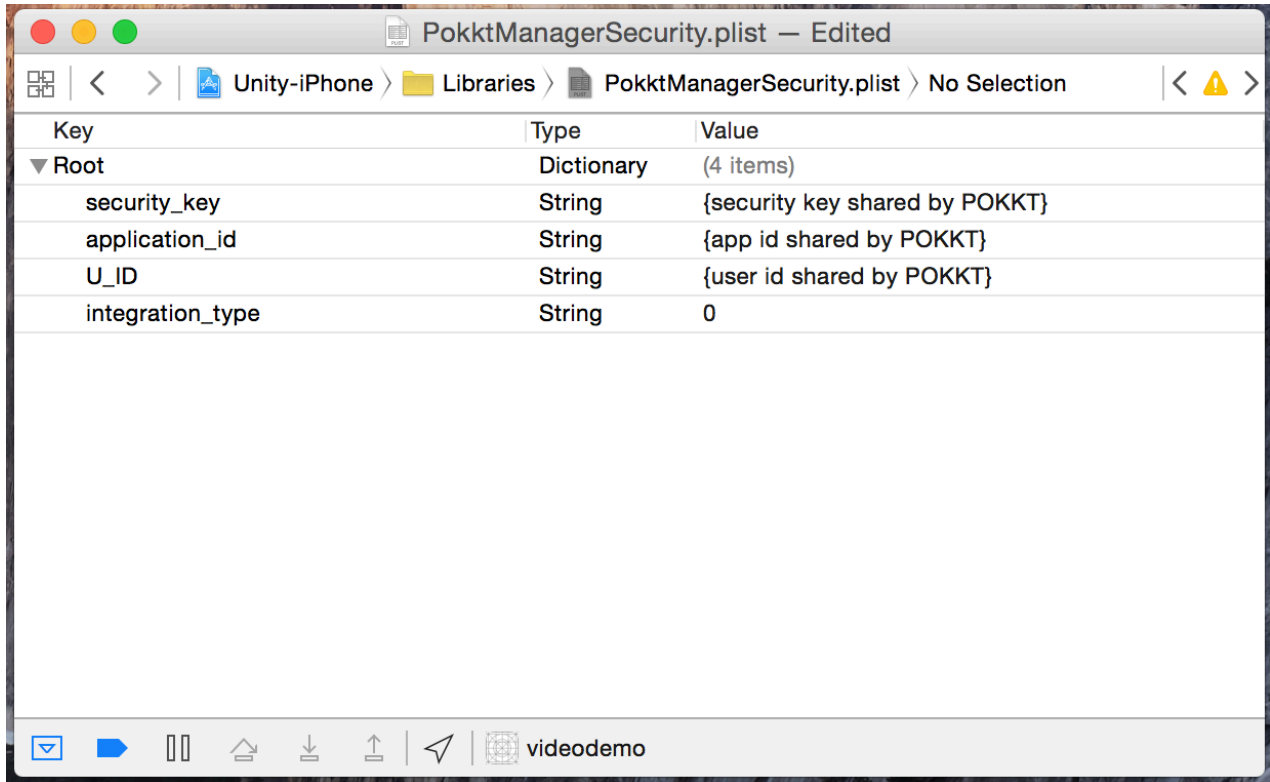
PokktResource.bundle

Once you export the iOS project, make sure to add the “PokktResource.bundle” to the exported Xcode project. You can find this inside Plugins -> iOS directory.

3. PokktManagerSecurity.plist Setup:

Step 4: Add meta-data information

Refer the following image for reference:



Note:

Leave the “integration_type” value to “0” for now.

You can skip this process completely and pass these values to the SDK via code, the process is explained further, under “SDK Setup on Unity3d” topic.

4. SDK Setup on Unity3d:

Initialize PokktManager

You can start with setting up the following values.

- Security Key
- Application Id
- User Id
- Integration Type
- Auto Cache Video

These values can also be set in “PokktManagerSecurity.plist”. In order to disable auto-caching of videos, you are required to enter these values in your code and NOT in PokktManagerSecurity.plist.

```
// SET THESE UP AS PER VALUES PROVIDED TO YOU
PokktManager.SecurityKey = "<your security key>";
PokktManager.ApplicationId = "<your application id>";
PokktManager.UserId = "<your user id>";
PokktManager.IntegrationType = "0";
```

After setting these values, make sure to set the auto caching option in the SDK ONLY AFTER you set the params. Ref.:

```
// make sure to set auto caching once the params are set
PokktManager.GetInstance().AutoCaching = <true/false>;
```

5. Video Ad Functionalities:

There are 7 events to manage the video caching and its playback, these are:

- VideoClosedEvent
- VideoDisplayedEvent
- VideoSkippedEvent
- VideoCompletedEvent
- VideoGratifiedEvent
- DownloadCompletedEvent
- DownloadFailedEvent

Add listeners to these events in the Awake()/Start() method of your MonoBehaviour or similar class. These are all of EventCustom type. Below are the reference on how to use them:

Reference on how to consume them:

```
// handle pokkt video ad events
PokktManager.GetInstance().Dispathcer.VideoClosedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoSkippedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoCompletedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoGratifiedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.VideoDisplayedEvent +=
(string message) =>
{
};

PokktManager.GetInstance().Dispathcer.DownloadCompletedEvent +=
(string message) =>
{
    string text = "Video VC is: " + message;
};
```

```
PokktManager.GetInstance().Dispathcer.DownloadFailedEvent +=  
(string message) =>  
{  
};
```

A video file is cached on user's device. You can set the auto-caching option in the beginning, as mentioned earlier in this document. In case of manual caching, call the following to start video caching:

```
PokktManager.GetInstance().StartVideoCaching();
```

Before playing video or showing button to play video, Application should check whether video is cached or not by calling the following:

```
PokktManager.GetInstance().IsVideoAvailable()
```

You should listen to "DownloadCompletedEvent" to check whether download is completed or not, you can show the play buttons once you receive this event.

Furthermore, Application can decide to play video as incent (user will be gratified after watching complete video) or non-incent (user will not be gratified after watching complete video). You must provide the screen-name parameter for it. Followings are the method calls to me made:

```
PokktManager.GetInstance().GetVideo("screen_name");  
PokktManager.GetInstance().GetVideoNonIncent("screen_name");
```

Next, you can listen to VideoGratifiedEvent to get the coins earned, if at all, by watching the last video.

6. Debugging and Logging

You can enable the SDK logs by setting the debugging option to true anytime. Ref.:

```
PokktManager.GetInstance().setDebug(<true/false>);
```

This can help you debug basic issues related to Pokkt SDK.

This concludes the integration documentation. It is highly suggested that you should check the sample app that is provided to you to understand it better.