wexer

iOS - Swift/ObjectiveC Content SDK Integration Guide

Date	2nd June 2020
Guide Version	1.2

Wexer provides iOS SDK that enables developers to show On Demand content in the app in the form of different types of collections, list of classes, filter/search content, and play the videos.

This guide will show you how to install the WexerContent SDK.

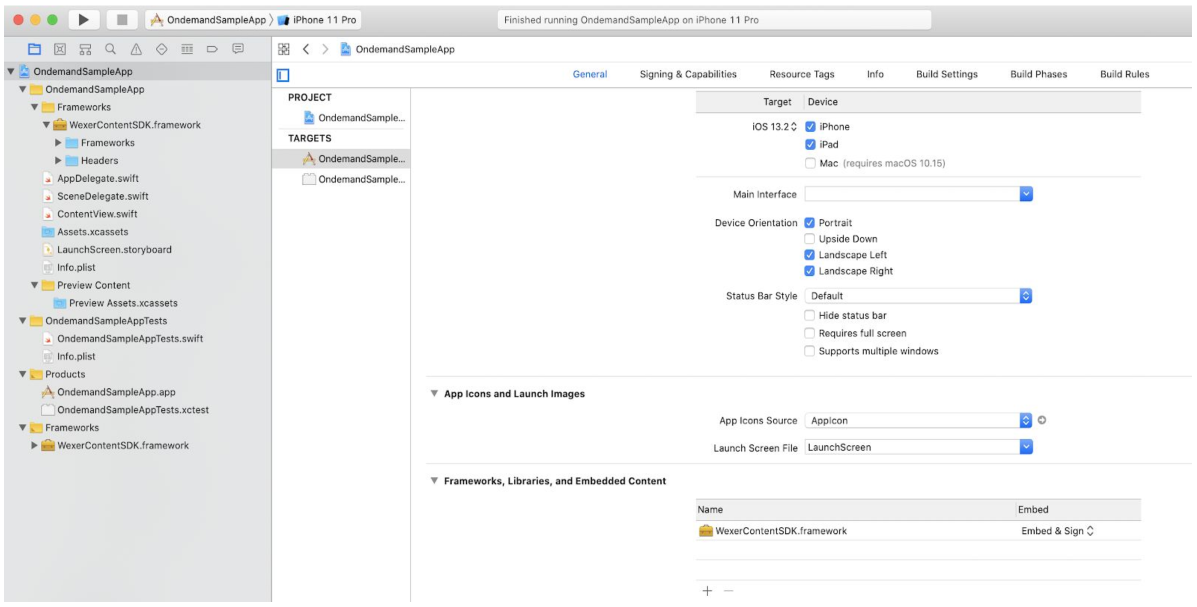
Step 1: Install SDK

Prerequisites:

- Xcode 9.0 or higher with at least iOS SDK 11.0 or higher
- Grab the following details to configure SDK
    1. API BaseURL
    2. Client Secret
    3. API Key
    4. TenantId
    5. LocalyticsKey

Option 1: Manually

1. Unzip shared WexerContent SDK folder.
2. Drag the **WexerContentSDK.framework** inside your project under the main project file.
3. Embed the framework. Select your app.xcodeproj file. Under "**General**", add the WexerContentSDK framework as an **embedded binary**.
4. Expand WexerContentSDK.framework. Drag & embed **Localytics.framework** as an embedded binary.



```
▼ OndemandSampleApp
  ▼ OndemandSampleApp
    ▼ Frameworks
      ▼ WexerContentSDK.framework
        ▶ Frameworks
        ▶ Headers
      AppDelegate.swift
      SceneDelegate.swift
      ContentView.swift
      Assets.xcassets
      LaunchScreen.storyboard
      Info.plist
    ▼ Preview Content
        Preview Assets.xcassets
  ▼ OndemandSampleAppTests
      OndemandSampleAppTests.swift
      Info.plist
  ▼ Products
      OndemandSampleApp.app
      OndemandSampleAppTests.xctest
  ▼ Frameworks
    ▶ WexerContentSDK.framework
```

Target	Device
iOS 13.2	✓ iPhone
	✓ iPad
	☐ Mac (requires macOS 10.15)

Main Interface:

Device Orientation: ✓ Portrait, ☐ Upside Down, ✓ Landscape Left, ✓ Landscape Right

Status Bar Style: Default

☐ Hide status bar
☐ Requires full screen
☐ Supports multiple windows

App Icons Source: AppIcon

Launch Screen File: LaunchScreen

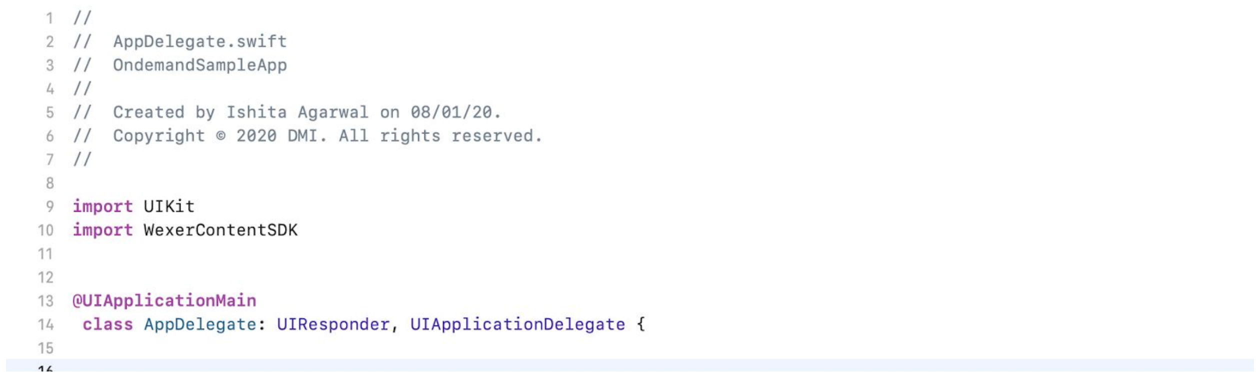
Name	Embed
WexerContentSDK.framework	Embed & Sign

Option 2: CocoaPods

Details will be available once we will publish the framework on CocoaPods.

Step 2: Initialize WexerContentSDK

1. Import WexerContentSDK in AppDelegate.



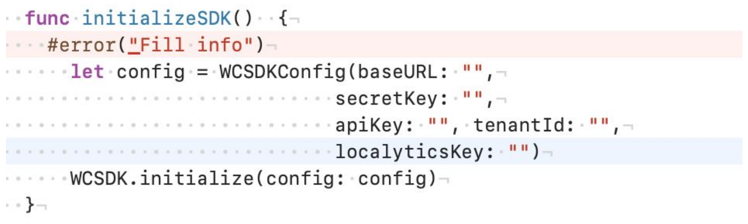
```swift
//
//  AppDelegate.swift
//  OndemandSampleApp
//
//  Created by Ishita Agarwal on 08/01/20.
//  Copyright © 2020 DMI. All rights reserved.
//

import UIKit
import WexerContentSDK


@UIApplicationMain
 class AppDelegate: UIResponder, UIApplicationDelegate {

```

2. Initialize configuration object for SDK in AppDelegate with shared API baseURL,client,secretKey,tenantId & localyticsKey details. Please contact the Wexer sales team to obtain test keys.



```swift
  func initializeSDK()  {
   #error("Fill info")
      let config = WCSDKConfig(baseURL: "",
                               secretKey: "",
                               apiKey: "", tenantId: "",
                               localyticsKey: "")
      WCSDK.initialize(config: config)
  }
```

Step 3: Call SDK methods

(Class Reference https://wexersdk.firebaseapp.com/Classes/WCSDK.html)

1. On Demand Collections -

```
WCSDK.getOndemandCollections(collectionId: nil,
                             maxResults: nil)
                            { (collections, error) in
                            }
```

2. On Demand Classes - It allows to fetch response with pagination which can be sorted on date/alphabet with “asc”/”dsc”

```
WCSDK.getOndemandClasses(page: 1,
                         pageSize: 10,
                         sort: "date",
                          order: "asc")
                          { (response, error) in
                          }
```

3. On Demand class detail - for single class having tag as “43794”

```
WCSDK.getOndemandClassDetails(for: "43794")
                             { (classobj, response) in
                              }
```

4. Show On Demand classes filter parameters

```
WCSDK.getOndemandMetaData { (response, error) in
                          }
```

5. Search & filter On Demand classes - sorting can be applied on date/alpha

```
var filterObj = WCSDKOndemandFilterRequest()
 filterObj.classLanguage = "en"
 filterObj.query = "yoga"  // provide search text in query parameters
 filterObj.provider = "cycling"
 WCSDK.getOndemandClassesForCriteria(with: filterObj)
                                    { (response, error) in
                                     }
```

6. Perform On Demand content: On Demand class can be performed only when the user has logged in with valid subscription.

   a. Create user session with providing user name which could be any alphanumeric unique value.

```
            WCSDK.startSession(userName: "test1")
                                    { (response, error) in
                                     }
```

   b. Activate subscription

```
let subscriptionObj = WCSDKUserSubscriptionRequest(startDate: Date(),
                                                       subscriptionId:"monthly")
 WCSDK.activateSubscription(subscription: subscriptionObj)
                                    { (response, error) in
                                     }
```

   c. Play On Demand - pass viewcontroller reference on which player shall be presented.

```
WCSDK.performOndemandContent(for: "43794",
            controller: self.window.rootViewController!)
            {(error) in
            }
```

`

   It will work if a user has logged in with a valid subscription.. It will throw an error as NSError having code & userInfo with details .
   Error - 403 - Not Subscribed
   Step 1: Call startSession method for login
   Step 2: call activateSubscription method

   d. Define custom behaviours on play/pause/exit events.

```
WCSDK.shared.playerExitCallBack = { value in
                                     self.playedDuration = value // custom behaviour   }
```