Build a sample iOS app to evaluate the Connect SDK

Before you start integrating the Connect library into a native iOS app, we recommend installing our sample app with the library.

Requirements

- **Development environment**. To build and run the sample app, you need Xcode 15 with Command Line Tools and a recent CocoaPods version.
- **Mobile app compatibility**. You can run the sample app on simulators and real devices. iOS 14 or later is required.
- **Acoustic Connect**. If you have an Ultimate license for Connect and want to check how session replay works, you'll need an access key. For instructions, see Get an application key for the Connect library.

Setup instructions

1. Clone the sample app code from our GitHub repository.

Shell

```
git clone https://github.com/go-acoustic/SampleCode_Connect_iOS_SwiftUI.git
```

2. *(optional)* In the root project directory, find a Podfile and open it in a text editor. Note that use_frameworks is uncommented. If you don't want to use the debug version, change the pod value from AcousticConnectDebug to AcousticConnect. Never use the release and the debug version in the same Podfile.

Ruby

```ruby
source 'https://github.com/CocoaPods/Specs.git'
platform :ios, '14.0'
workspace 'SwiftUIMindBlowing'

target 'SwiftUIMindBlowing' do
  use_frameworks!
  pod 'AcousticConnectDebug'
end
```

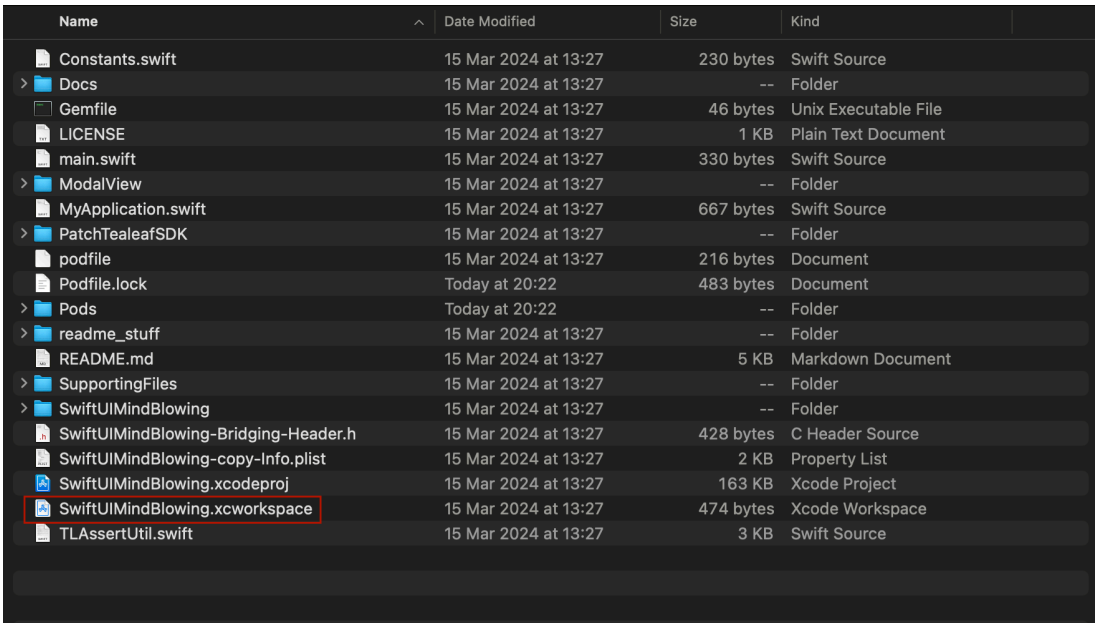
3. In a terminal, navigate to the root project directory and install the pods. Make sure the command is completed with no errors. If you get an error, run the same command with the `--verbose` option and share the error log with our services team.

Shell

```
cd SampleCode_Connect_iOS_SwiftUI
pod install
```

> **Note**
>
> Sometimes CocoaPods fails to load the latest version. So you may need to pull it using `pod update`.

4. In a file explorer, open the **SwiftUIMindBlowing.xcworkspace** file (not **SwiftUIMindBlowing.xcodeproj**).

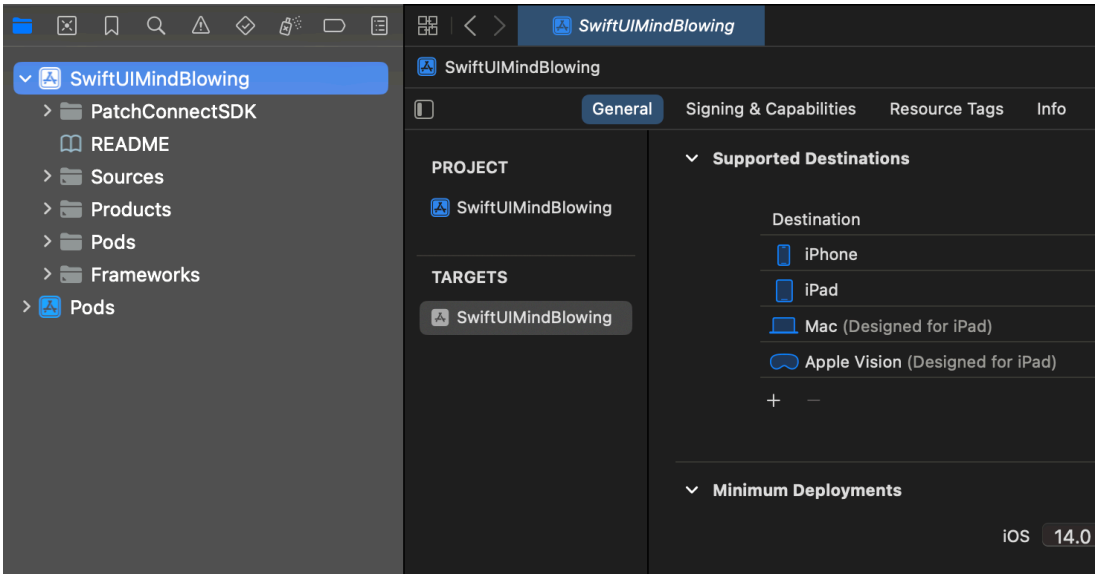


5. *(Connect Ultimate only)* Open the AppDelegate file and update the default application key and collector URL with the credentials associated with your Connect subscription. This is necessary for access to session replay in Connect (see the exercise below).

```
SwiftUIMindBlowing ⟩ Sources ⟩ AppDelegate ⟩ application(_:didFinishLaunchingWithOptions:)
```

```swift
//
//  Copyright © 2019 An Tran. All rights reserved.
//

import UIKit
import Tealeaf
import EOCore
import Connect
class AppDelegate: UIResponder, UIApplicationDelegate {
    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
        [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        let connectApplicationHelperObj =  ConnectApplicationHelper()

        // Enable library to load configuration settings
        let appKey:String = "b6c3709b7a4c479bb4b500000000xxxxx"
        let postMessageURL:String = "https://lib-us-2.brilliantcollector.com/collector/collectorPost"
        connectApplicationHelperObj.enableFramework(appKey, withPostMessageUrl: postMessageURL)

        // Read the new ConnectLayoutConfig settings
        let tlfAdvFilePath: String? = Bundle.main.path(forResource: "ConnectLayoutConfig", ofType: "json")
        var layoutConfigDict: [AnyHashable : Any] = [:]
        // read data into layoutConfigDict
        loadJson(filePath: tlfAdvFilePath!, jsonDict: &layoutConfigDict)

        // Update values in configuration for both json objects "AutoLayout" & "AppendMapIds"
        EOApplicationHelper.sharedInstance().setConfigItem("AutoLayout", value:layoutConfigDict["AutoLayout"],
            forModuleName:kTLFCoreModule)
        EOApplicationHelper.sharedInstance().setConfigItem("AppendMapIds",
            value:layoutConfigDict["AppendMapIds"], forModuleName:kTLFCoreModule)

        // Update values in configuration
        EOApplicationHelper.sharedInstance().setConfigItem("RemoveSwiftUIDuplicates", value:false,
            forModuleName:kTLFCoreModule)
        EOApplicationHelper.sharedInstance().setConfigItem("LogFullRequestResponsePayloads", value:true,
            forModuleName:kTLFCoreModule)

        return true
    }
```

6. Use the SwiftUIMindBlowing target to build and run the app.



Exercise (Connect Ultimate)

If you have an Ultimate license for Connect, you can check how user behavior data is captured.

1. Click around in the sample app.
2. In your Connect account, go to **Insights** > **Sessions** > **Session search**.
3. Find the session and play it back.

Notes:

- If a user is inactive for 30 minutes, their session times out.

- To learn more about the **Sessions** module, see Session replay in our marketing guide.

Updated about 2 months ago