

# Tips and Tricks for Working with a New Codebase

Charlie Fish



# About Me

- Built my first basic HTML website in 2006
- Started using the iOS SDK to build apps in 2009
- Attended my first 360iDev in 2010
- Started working at Wunderman Thompson Mobile (formerly POSSIBLE Mobile) in 2018

# Why?



SWIFT BY SUNDELL A WEEKLY BLOG ABOUT SWIFT DEVELOPMENT

# What makes a new codebase different?

- Prior work and conventions have already been defined
- Hundreds or thousands of files
- You didn't write it

# Open Source

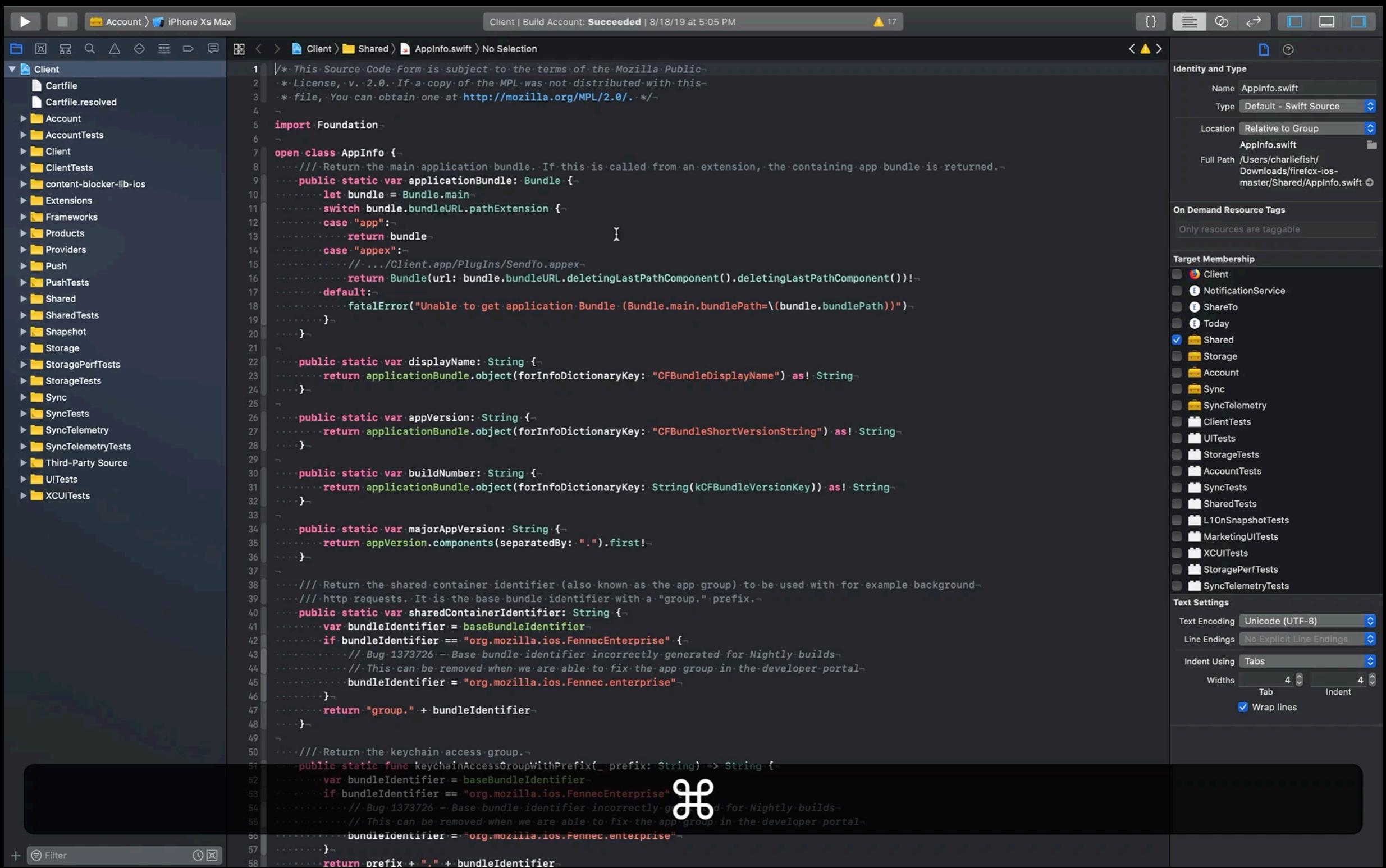
- Firefox
- Kickstarter
- GitHawk
- Wikipedia

# How to start?

- Jump right in
- Find a bug or something you want to change
- Ask questions!
- You always know more than you think you do!  
Trust your knowledge

# Xcode Tips

# Find (Command + F)



The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under the "Client" target, including files like Cartfile, Cartfile.resolved, Account, AccountTests, Client, ClientTests, content-blocker-lib-ios, Extensions, Frameworks, Products, Providers, Push, PushTests, Shared, SharedTests, Snapshot, Storage, StoragePerfTests, StorageTests, Sync, SyncTests, SyncTelemetry, SyncTelemetryTests, Third-Party Source, UITests, and XCUItests.
- Editor:** Displays the content of the `AppInfo.swift` file. The code is a Swift class that handles application bundle retrieval and shared container identifier generation.
- Identity and Type:** Shows the file's properties: Name (AppInfo.swift), Type (Default - Swift Source), Location (Relative to Group), Full Path (/Users/charliefish/Downloads/firefox-ios-master/Shared/AppInfo.swift).
- On Demand Resource Tags:** A section indicating "Only resources are taggable".
- Target Membership:** Shows the file is part of the "Shared" target, along with other targets like Client, NotificationService, ShareTo, Today, Storage, Account, Sync, SyncTelemetry, ClientTests, UITests, StorageTests, AccountTests, SyncTests, SharedTests, L10nSnapshotTests, MarketingUITests, XCUItests, StoragePerfTests, and SyncTelemetryTests.
- Text Settings:** Configuration for Text Encoding (Unicode (UTF-8)), Line Endings (No Explicit Line Endings), Indent Using (Tabs), and Widths (Tab width 4, Indent width 4). The "Wrap lines" checkbox is checked.

```
/* This Source Code Form is subject to the terms of the Mozilla Public
 * License, v. 2.0. If a copy of the MPL was not distributed with this
 * file, You can obtain one at http://mozilla.org/MPL/2.0/. */
import Foundation

open class AppInfo {
    /// Return the main application bundle. If this is called from an extension, the containing app bundle is returned.
    public static var applicationBundle: Bundle {
        let bundle = Bundle.main
        switch bundle.bundleURL.pathExtension {
            case "app":
                return bundle
            case "appex":
                // /Client.app/PlugIns/SendTo.appex
                return Bundle(url: bundle.bundleURL.deletingLastPathComponent().deletingLastPathComponent()!)
            default:
                fatalError("Unable to get application bundle (Bundle.main.bundlePath=\(bundle.bundlePath))")
        }
    }

    public static var displayName: String {
        return applicationBundle.object(forKey: "CFBundleDisplayName") as! String
    }

    public static var appVersion: String {
        return applicationBundle.object(forKey: "CFBundleShortVersionString") as! String
    }

    public static var buildNumber: String {
        return applicationBundle.object(forKey: String(kCFBundleVersionKey)) as! String
    }

    public static var majorAppVersion: String {
        return appVersion.components(separatedBy: ".").first!
    }

    /// Return the shared container identifier (also known as the app group) to be used with for example background
    /// http requests. It is the base bundle identifier with a "group." prefix.
    public static var sharedContainerIdentifier: String {
        var bundleIdentifier = baseBundleIdentifier
        if bundleIdentifier == "org.mozilla.ios.FennecEnterprise" {
            // Bug 1373726 - Base bundle identifier incorrectly generated for Nightly builds
            // This can be removed when we are able to fix the app group in the developer portal
            bundleIdentifier = "org.mozilla.ios.Fennec.enterprise"
        }
        return "group." + bundleIdentifier
    }

    /// Return the keychain access group.
    public static func keychainAccessGroupWithPrefix(_ prefix: String) -> String {
        var bundleIdentifier = baseBundleIdentifier
        if bundleIdentifier == "org.mozilla.ios.FennecEnterprise" {
            // Bug 1373726 - Base bundle identifier incorrectly generated for Nightly builds
            // This can be removed when we are able to fix the app group in the developer portal
            bundleIdentifier = "org.mozilla.ios.Fennec.enterprise"
        }
        return prefix + "." + bundleIdentifier
    }
}
```

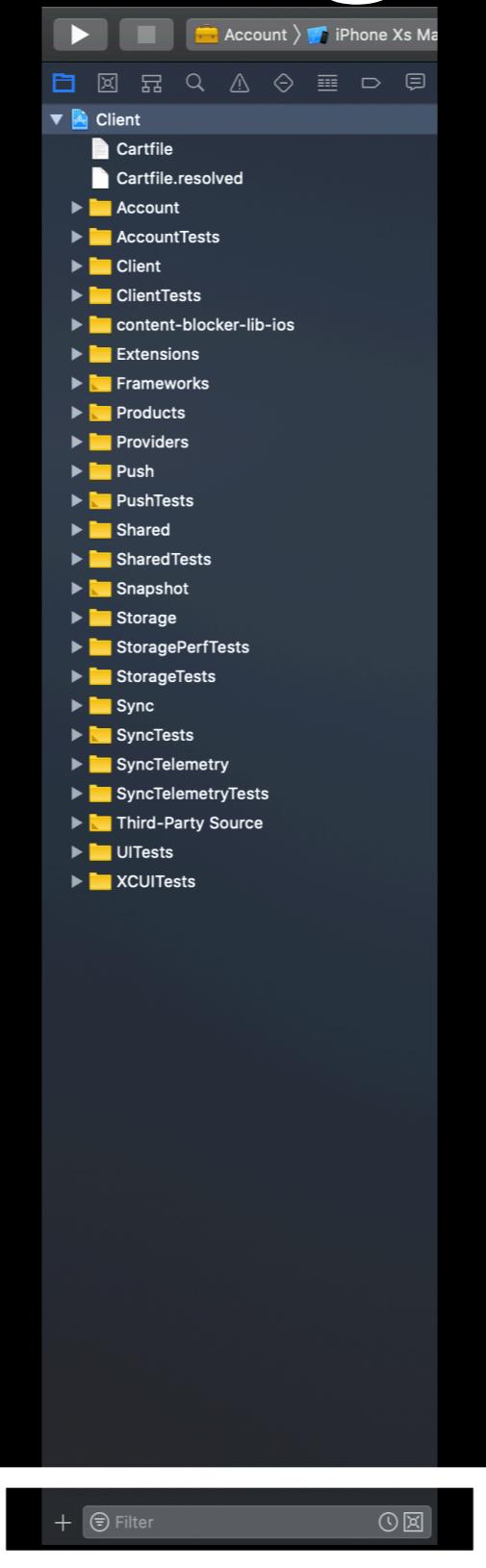
# Find All (Command + Shift + F)

The screenshot shows the Xcode interface with the 'Find All' search results for the method 'start()' in the file 'LeanplumIntegration.swift'. The code editor displays the implementation of 'start()' with numerous search results highlighted in yellow. The left sidebar shows the project structure under the 'Client' folder, and the top status bar indicates the device is 'Waiting for Device'.

```
164     LPAccountKey.fxaAccountVerified: profile?.hasSyncableAccount() ?? false
165     ...
166     ...
167     self.setupCustomTemplates()
168     ...
169     Leanplum.start(withUserId: nil, userAttributes: attributes, responseHandler: { _ in
170         self.track(event: .openedApp)
171     }
172     // We need to check if the app is a clean install to use for
173     // preventing the What's New URL from appearing.
174     if self.prefs?.intForKey(PrefsKeys.IntroSeen) == nil {
175         self.prefs?.setString(AppInfo.appVersion, forKey: LatestAppVersionProfileKey)
176         self.track(event: .firstRun)
177     } else if self.prefs?.boolForKey("SecondRun") == nil {
178         self.prefs?.setBool(true, forKey: "SecondRun")
179         self.track(event: .secondRun)
180     }
181
182     self.checkIfAppWasInstalled(key: PrefsKeys.HasFocusInstalled, isAppInstalled: self.focusInstalled(), lpEvent: .downloadedFocus)
183     self.checkIfAppWasInstalled(key: PrefsKeys.HasPocketInstalled, isAppInstalled: self.pocketInstalled(), lpEvent: .downloadedPocket)
184     self.recordSyncedClients(with: self.profile)
185 }
186 }
187
188 // Events
189 func track(event: LPEvent, withParameters parameters: [String: String]? = nil) {
190     guard isLPEnabled() else {
191         return
192     }
193     ensureMainThread {
194         guard !self.isPrivateMode() else {
195             return
196         }
197         if let params = parameters {
198             Leanplum.track(event.rawValue, withParameters: params)
199         } else {
200             Leanplum.track(event.rawValue)
201         }
202     }
203 }
204
205 func set(attributes: [AnyHashable: Any]) {
206     guard isLPEnabled() else {
207         return
208     }
209     ensureMainThread {
210         if !self.isPrivateMode() {
211             Leanplum.setUserAttributes(attributes)
212         }
213     }
214 }
215
216 func set(enabled: Bool) {
217     // Setting up Test Mode stops sending things to server.
218     if enabled { start() }
219     self.enabled = enabled
220     Leanplum.setTestModeEnabled(!enabled)
221 }
```

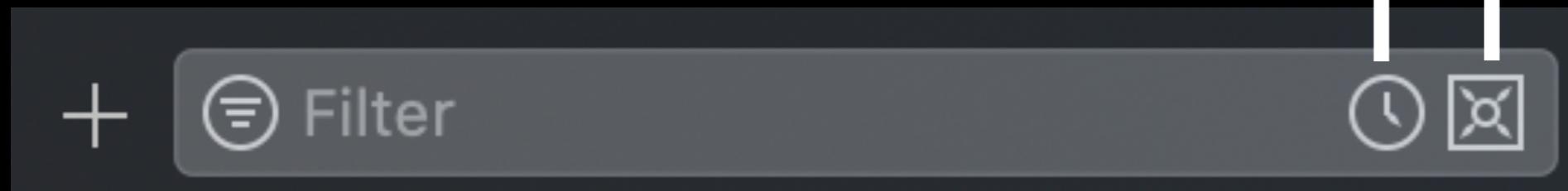
No Selection

# Project Navigator Filter

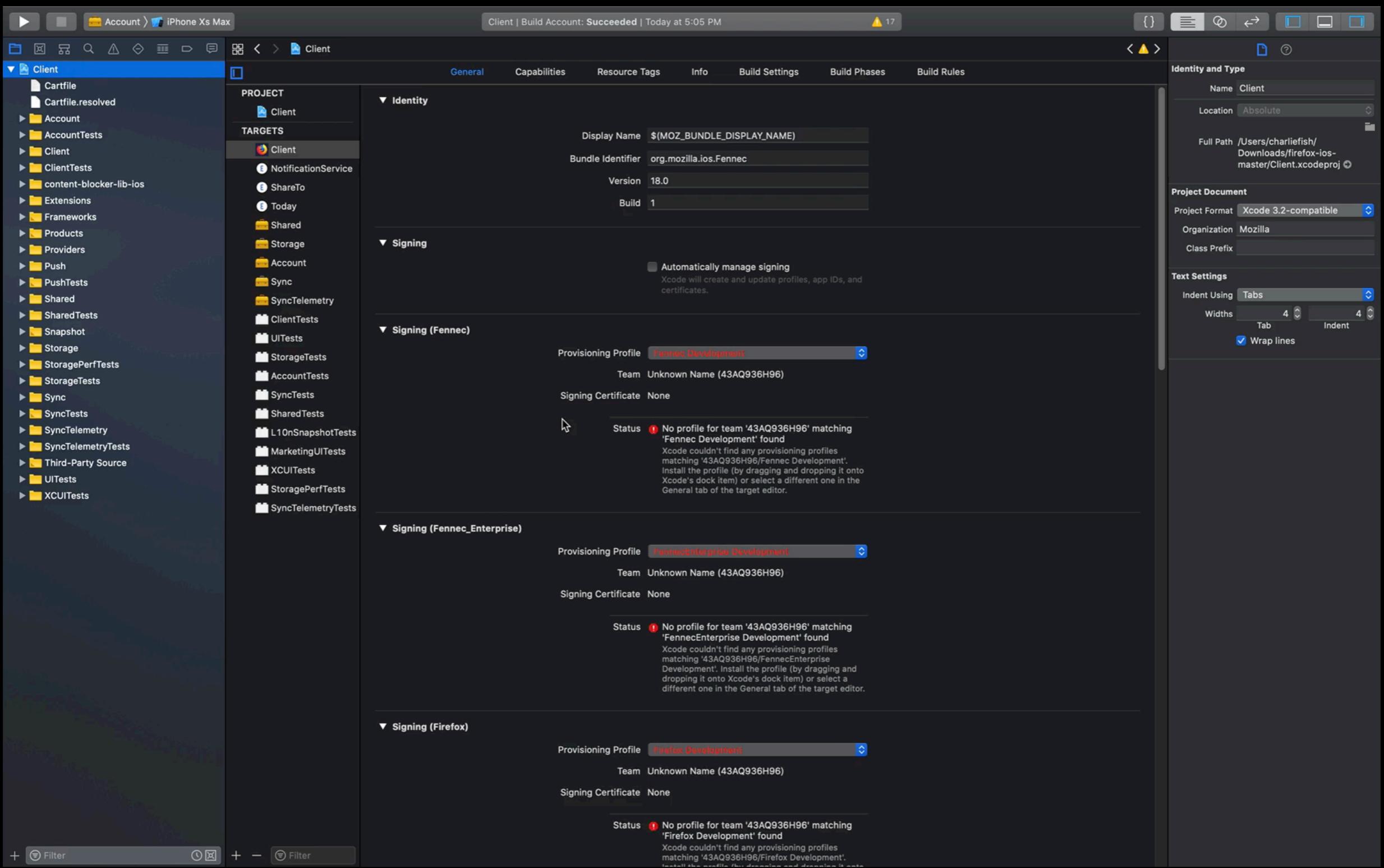


# Project Navigator Filter

Recent Files   Source Control



# Open Quickly (Command + Shift + O)



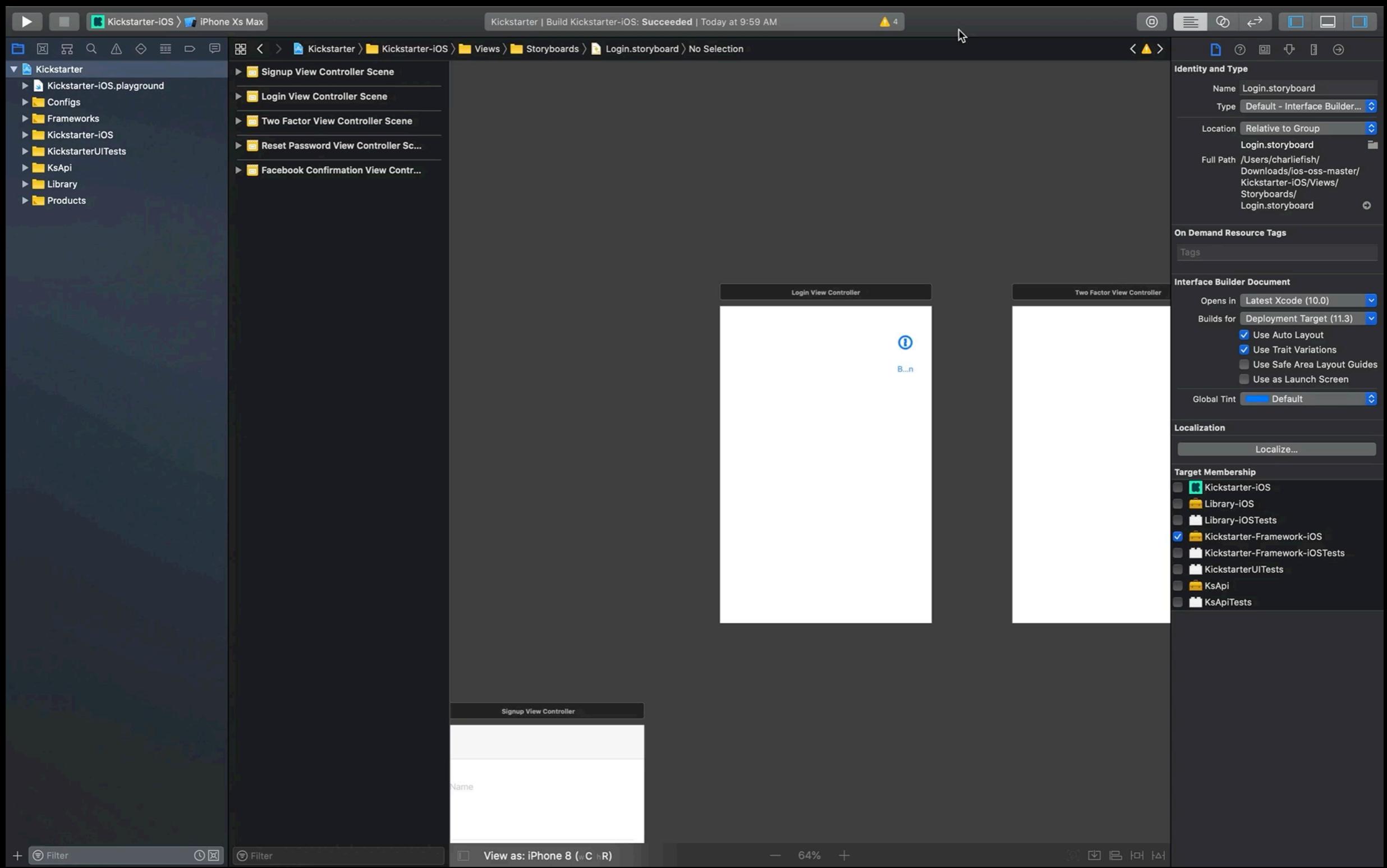
# Jump to Definition

The screenshot shows the Xcode interface with the following details:

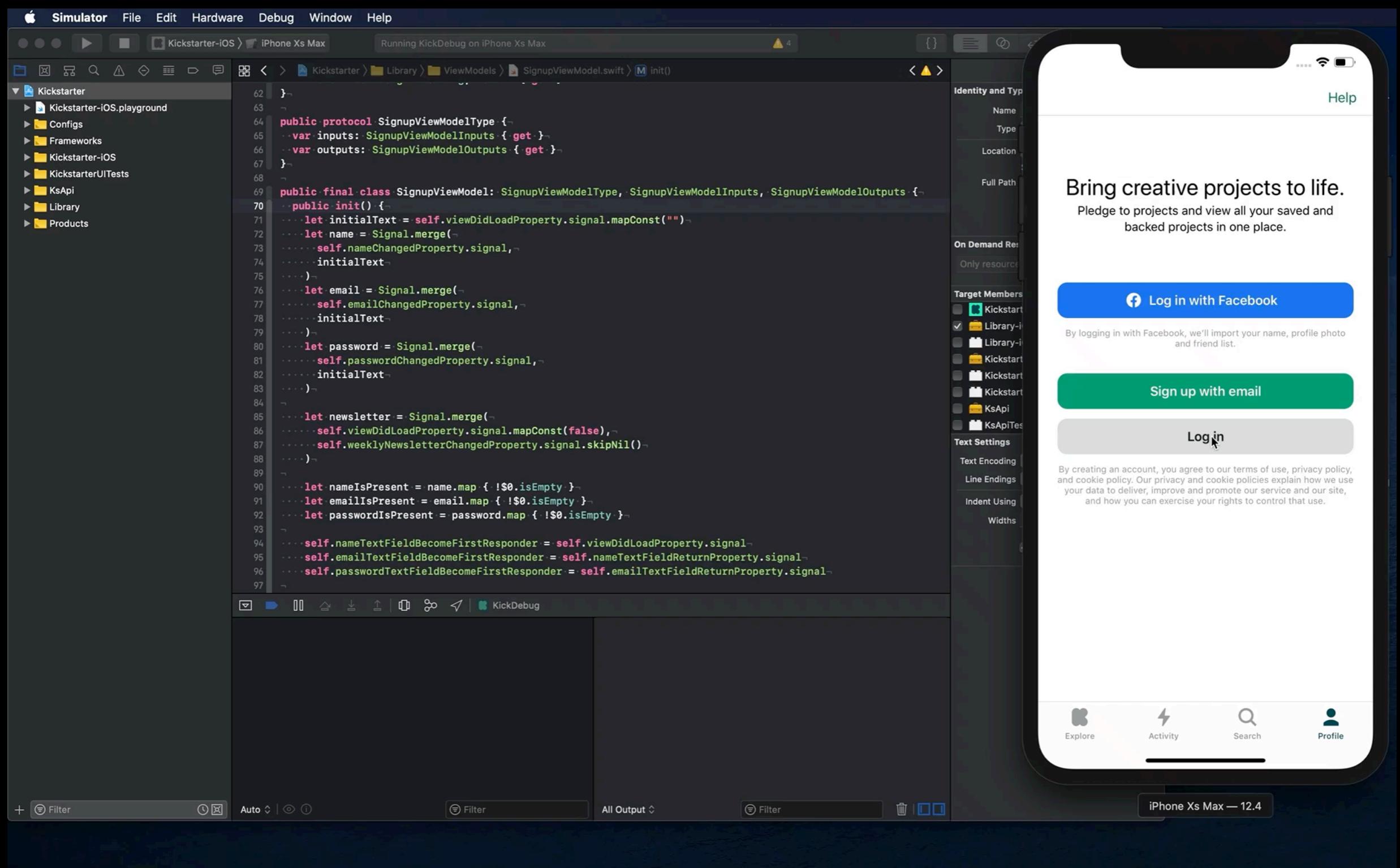
- Project Navigator:** Shows the project structure under "Kickstarter".
- Editor:** Displays the content of `SignupViewController.swift`. The code defines a `SignupViewController` class that conforms to `MFMailComposeViewControllerDelegate`. It includes properties for various UI components like scroll view, disclaimer button, email text field, etc., and implements `viewDidLoad` to add targets for text field delegates.
- Identity and Type:** Shows the file is a "Default - Swift Source" located relative to the group.
- On Demand Resource Tags:** Notes that only resources are taggable.
- Target Membership:** Lists the target as "Kickstarter-Framework-iOS".
- Text Settings:** Configures encoding as Unicode (UTF-8), line endings as No Explicit Line Endings, and indent using spaces with widths of 2 for tab and indent.

```
1 import Library
2 import MessageUI
3 import Prelude
4 import PreludeUIKit
5 import UIKit
6
7 internal final class SignupViewController: UIViewController, MFMailComposeViewControllerDelegate {
8     fileprivate let viewModel: SignupViewModelType = SignupViewModel()
9     fileprivate let helpViewModel = HelpViewModel()
10
11     @IBOutlet fileprivate var scrollView: UIScrollView!
12     @IBOutlet fileprivate var disclaimerButton: UIButton!
13     @IBOutlet fileprivate var emailTextField: UITextField!
14     @IBOutlet fileprivate var formBackgroundView: UIView!
15     @IBOutlet fileprivate var nameTextField: UITextField!
16     @IBOutlet fileprivate var newsletterLabel: UILabel!
17     @IBOutlet fileprivate var newsletterSwitch: UISwitch!
18     @IBOutlet fileprivate var passwordTextField: UITextField!
19     @IBOutlet fileprivate var rootStackView: UIStackView!
20     @IBOutlet fileprivate var signupButton: UIButton!
21
22     internal static func instantiate() -> SignupViewController {
23         let vc = Storyboard.Login.instantiate(SignupViewController.self)
24         vc.helpViewModel.inputs.configureWith(helpContext: .signup)
25         vc.helpViewModel.inputs.canSendEmail(MFMailComposeViewController.canSendMail())
26         return vc
27     }
28
29     internal override func viewDidLoad() {
30         super.viewDidLoad()
31
32         self.nameTextField.addTarget(
33             self,
34             action: #selector(self.nameTextFieldReturn),
35             for: .editingDidEndOnExit
36         )
37
38         self.nameTextField.addTarget(
39             self,
40             action: #selector(self.nameTextFieldChanged(_:)),
41             for: [.editingDidEndOnExit, .editingChanged]
42         )
43
44         self.emailTextField.addTarget(
45             self,
46             action: #selector(self.emailTextFieldReturn),
47             for: .editingDidEndOnExit
48         )
49
50         self.emailTextField.addTarget(
51             self,
52             action: #selector(self.emailTextFieldChanged(_:)),
53             for: [.editingDidEndOnExit, .editingChanged]
54         )
55
56         self.passwordTextField.addTarget(
57             self,
58             action: #selector(self.passwordTextFieldReturn)
59         )
60     }
61 }
```

# Assistant Editor (Command + Option + Enter)



# View Debugger



# Summary

- Open Source
- Xcode Tips
- Ask questions!

# Conclusion

- Slides available at <https://charlie.fish>
- Additional questions via Twitter (@char\_fish)
- Contact information available at <https://charlie.fish/contact>





