

#WWDC19

Getting Started with Xcode

Prachi Pai Asnodkar, Xcode Engineer

Holly Borla, Xcode Engineer

Honza Dvorsky, Xcode Engineer

Creating a New Project

Writing and Navigating Source Code

Running and Debugging

Working with Packages and Frameworks

Testing and Distribution

#WWDC19

Creating a New Project

Prachi Pai Asnodkar, Xcode Engineer

Mind | Build Mind: **Succeeded** | Today at 3:31 PM

Mind > Mind > AppDelegate.swift > AppDelegate

Mind

- Mind
 - AppDelegate.swift
 - SceneDelegate.swift
 - ContentView.swift
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - Preview Content
 - MindUITests
 - MindUITests.swift
 - Info.plist
 - Products

```

1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
            
```

UISceneSession Lifecycle

Identity and Type

Name: AppDelegate.swift

Type: Default - Swift Source

Location: Relative to Group
AppDelegate.swift

Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind

MindUITests

Text Settings

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: Tab: 4, Indent: 4

Wrap lines

Mind | Build Mind: **Succeeded** | Today at 3:31 PM

Mind > Mind > AppDelegate.swift > AppDelegate

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

Name: AppDelegate.swift
Type: Default - Swift Source
Location: Relative to Group
AppDelegate.swift
Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind
 MindUITests

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: Tab: 4, Indent: 4
 Wrap lines

Source Editor

The screenshot displays the Xcode Source Editor interface. The main editor window shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application(_:application, didFinishLaunchingWithOptions:) and applicationWillTerminate(_:application:). A third method, application(_:application, configurationForConnectingConnectingSceneSession:options:), is partially visible at the bottom. The right-hand side of the interface shows the 'Identity and Type' inspector, which provides details about the selected file, including its name, type, location, and full path. Below this, the 'Target Membership' section shows that the file is associated with the 'Mind' target. The 'Text Settings' section at the bottom right allows for configuring text encoding, line endings, and indentation.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

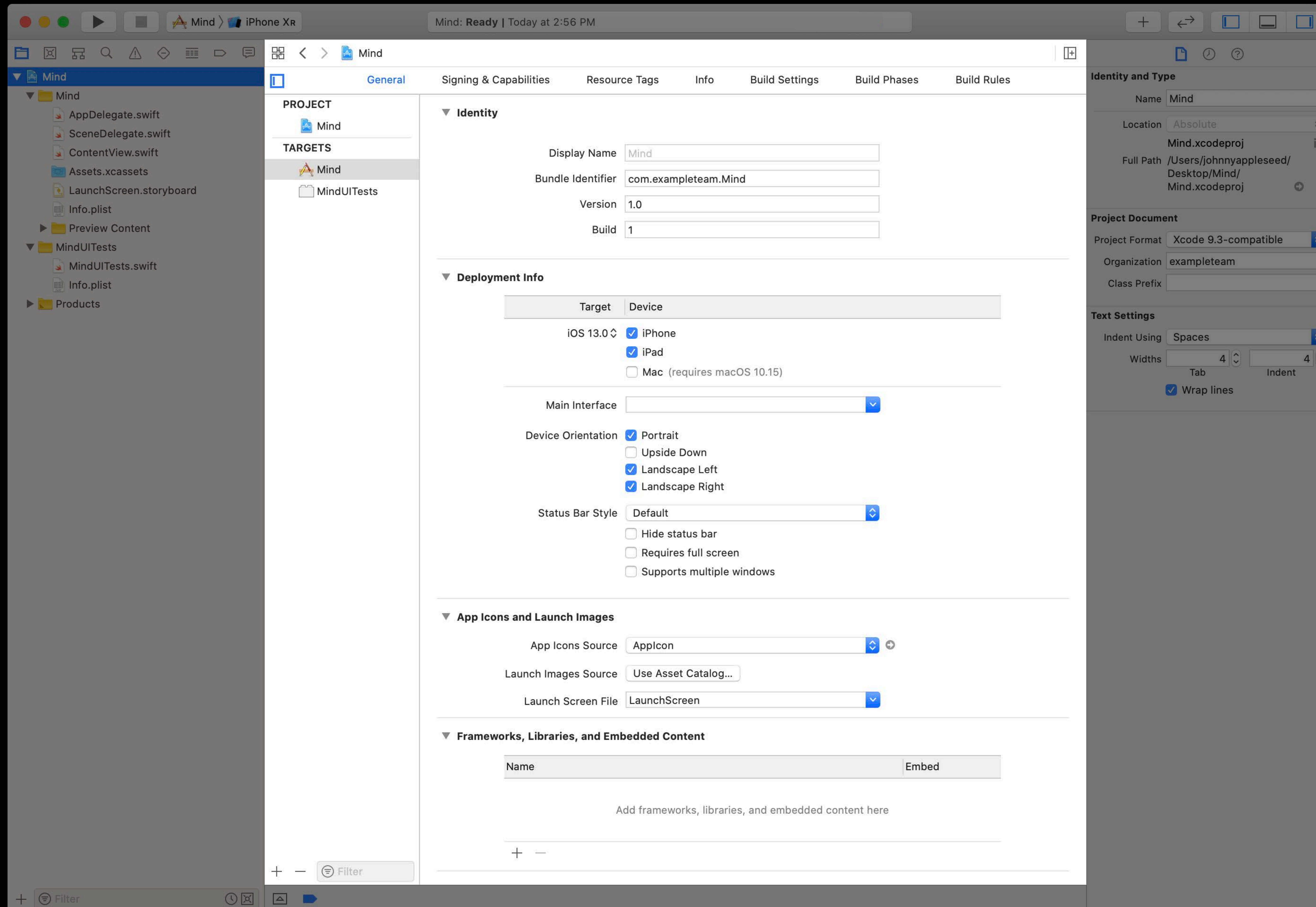
Target Membership

- Mind
- MindUITests

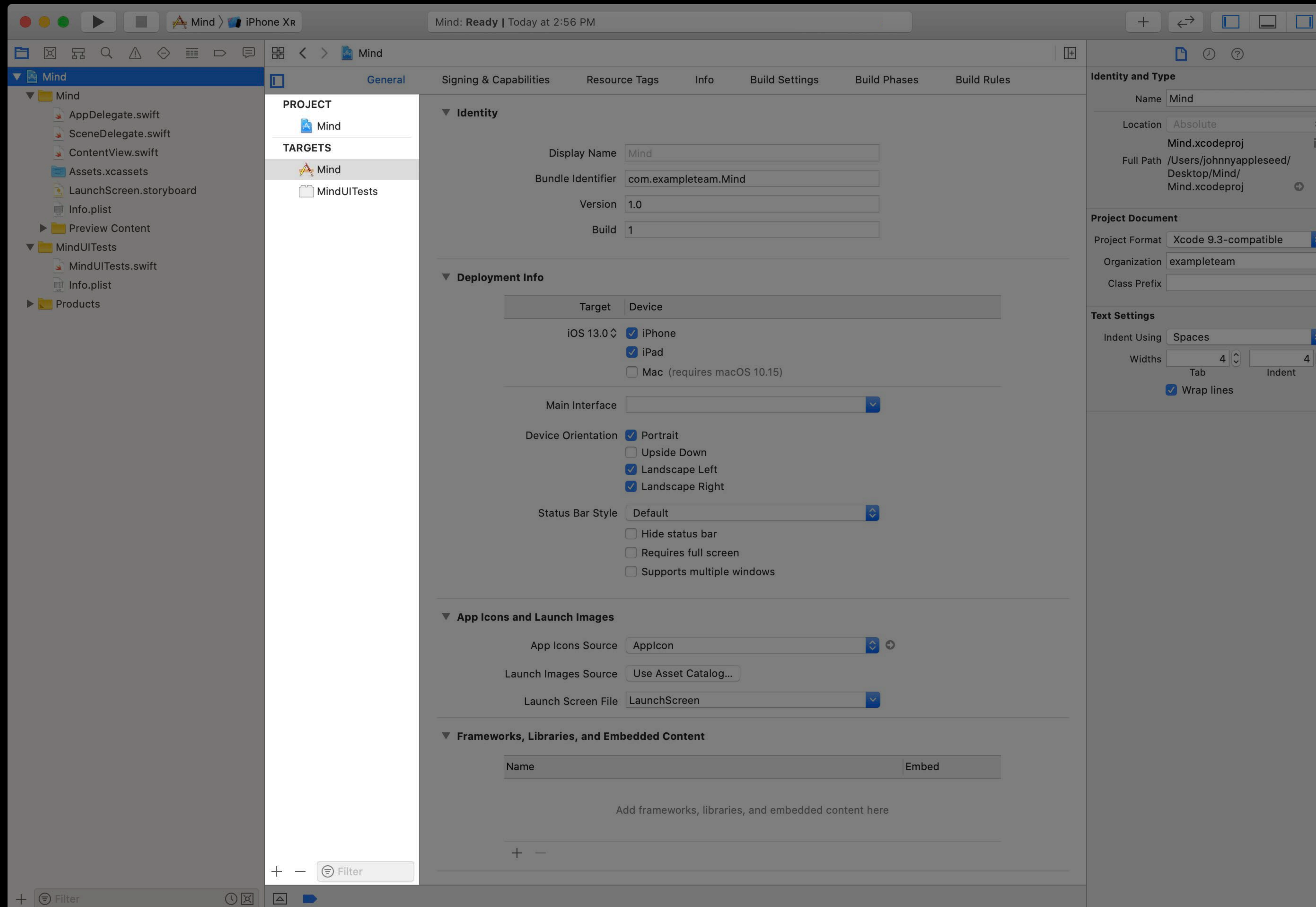
Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Project Editor



Project Editor



Navigator

The screenshot displays the Xcode IDE interface. On the left is the Navigator pane showing a project tree with 'AppDelegate.swift' selected. The central Editor pane shows the Swift code for AppDelegate.swift, including comments and function definitions. On the right is the Inspector pane, which is currently showing the 'Identity and Type' section for the selected file.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

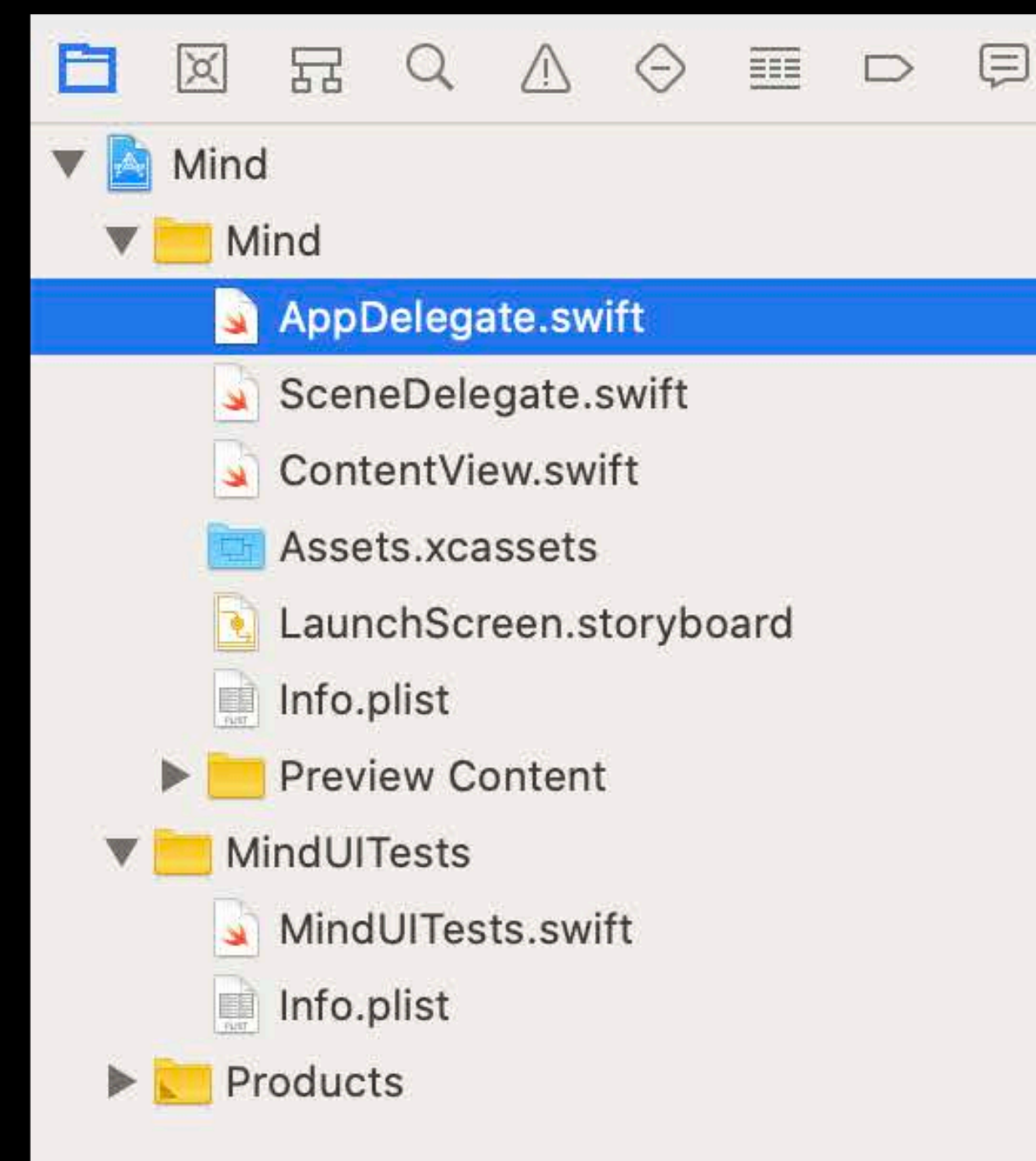
Target Membership

- Mind
- MindUITests

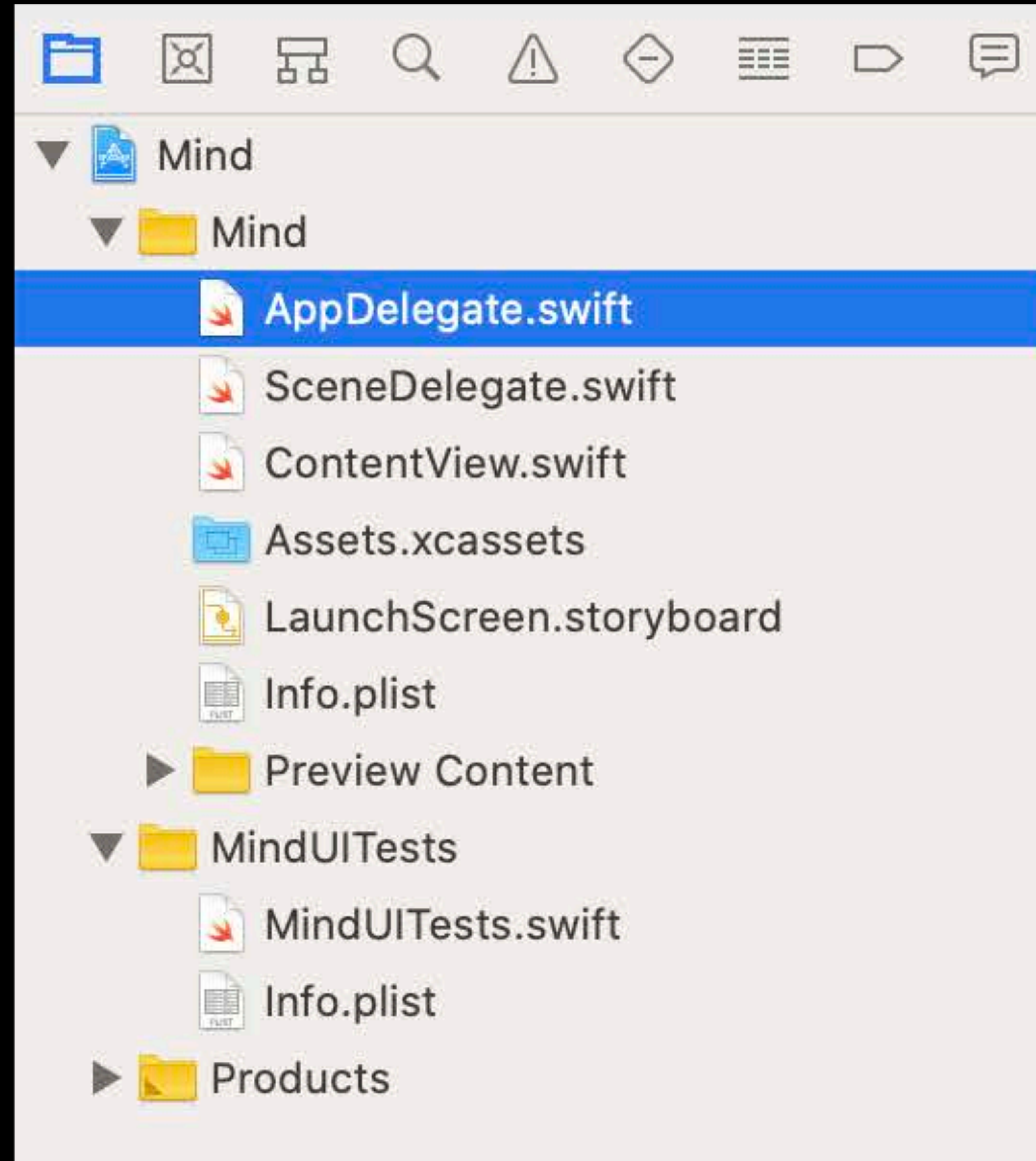
Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

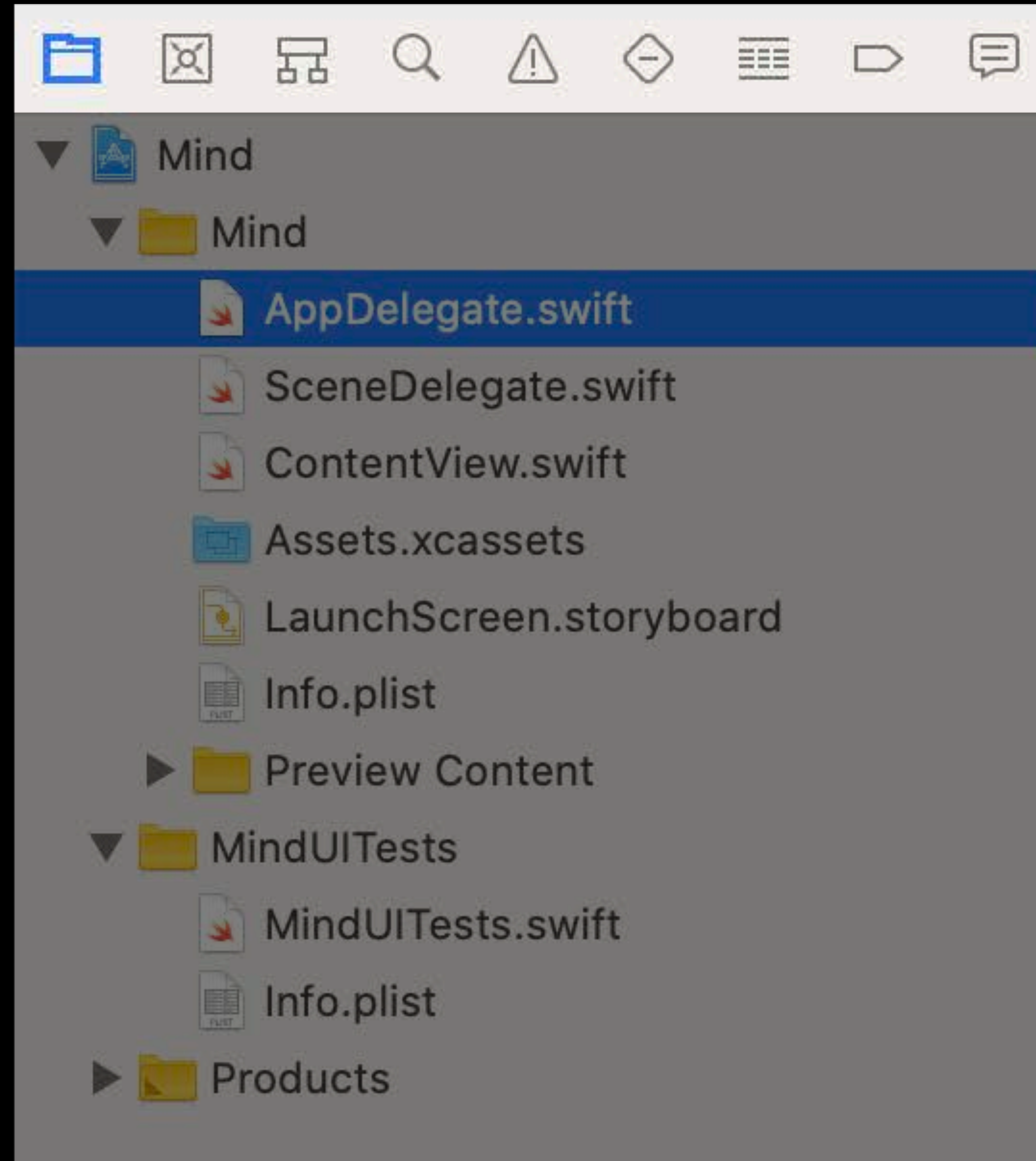
Navigator



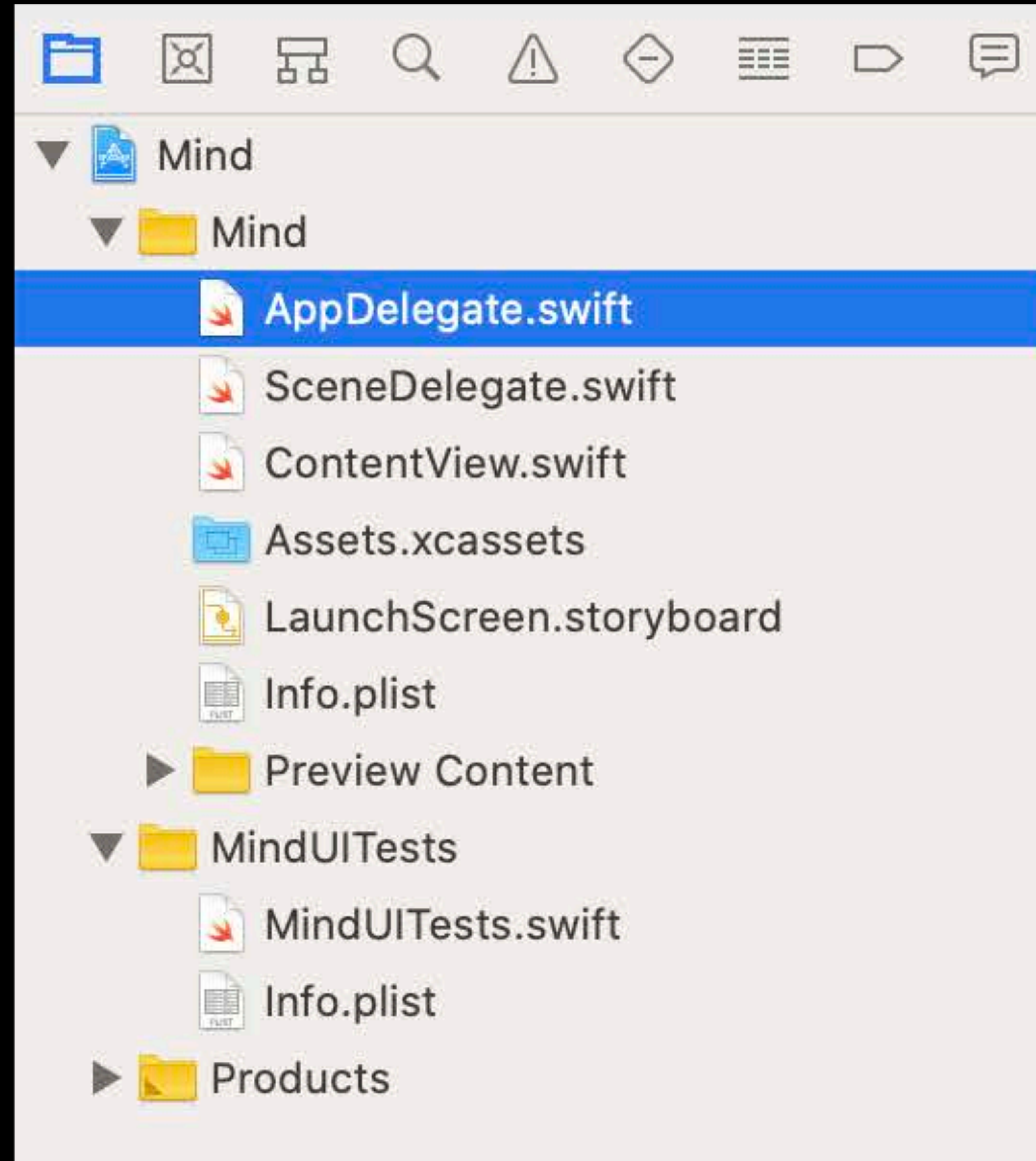
Navigator



Navigator



Navigator



Inspector

The screenshot displays the Xcode IDE interface. The main editor shows the Swift source code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application(_:application, didFinishLaunchingWithOptions:) and applicationWillTerminate(_:application:). A comment marks the UISceneSession Lifecycle section, followed by the application(_:application, configurationForConnecting:connectingSceneSession:options:) method.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

The Inspector panel on the right shows the following settings:

- Identity and Type:** Name: AppDelegate.swift, Type: Default - Swift Source, Location: Relative to Group, Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift
- On Demand Resource Tags:** Only resources are taggable
- Target Membership:** Mind (checked), MindUITests (unchecked)
- Text Settings:** Text Encoding: No Explicit Encoding, Line Endings: No Explicit Line Endings, Indent Using: Spaces, Widths: Tab (4), Indent (4), Wrap lines (checked)

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application(_:application, didFinishLaunchingWithOptions:) and applicationWillTerminate(_:application:). A second application method is partially visible at the bottom. The right-hand side of the interface features the 'Identity and Type' inspector, which shows the file's name, type, location, and full path. Below this, the 'Target Membership' section shows the 'Mind' target selected. The 'Text Settings' section at the bottom right shows options for text encoding, line endings, and indentation.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application(_:application, didFinishLaunchingWithOptions:) and applicationWillTerminate(_:application:). A second application method is partially visible at the bottom. The right-hand side of the interface shows the 'Identity and Type' inspector, which provides details about the selected file, including its name, type, location, and full path. Below this, the 'Target Membership' section shows that the file is associated with the 'Mind' target. The 'Text Settings' section at the bottom right allows for configuring text encoding, line endings, and indentation settings.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

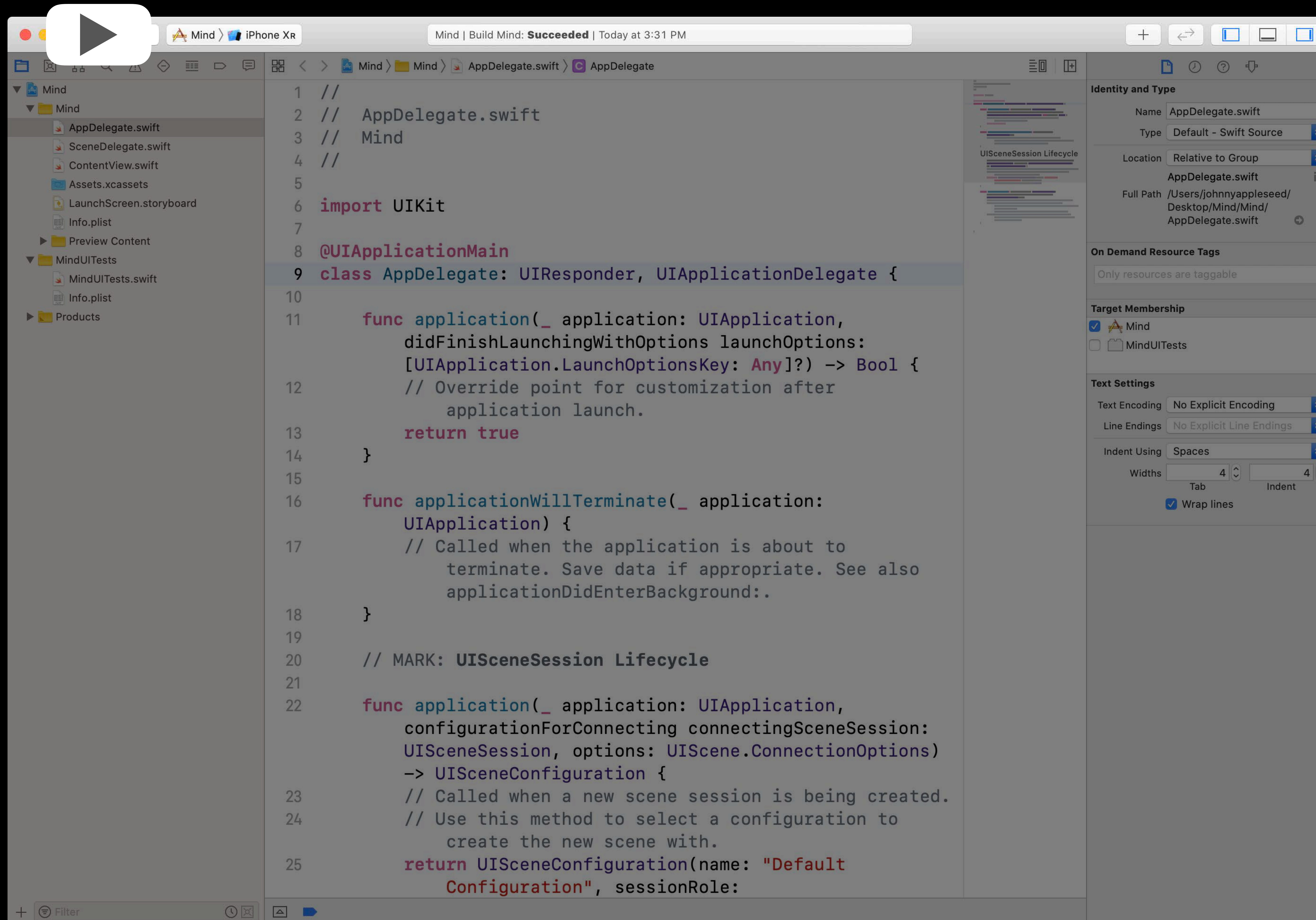
Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar



The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application:didFinishLaunchingWithOptions and applicationWillTerminate. A comment marks the UISceneSession Lifecycle section, followed by the application:configurationForConnecting method.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

Name: AppDelegate.swift
Type: Default - Swift Source
Location: Relative to Group
AppDelegate.swift
Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind
 MindUITests

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: Tab: 4, Indent: 4
 Wrap lines

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application(_:application, didFinishLaunchingWithOptions:) and applicationWillTerminate(_:application). A second application method is partially visible at the bottom. The right-hand side of the interface shows the 'Identity and Type' inspector, which is currently displaying settings for the selected AppDelegate.swift file. The 'Target Membership' section shows that the file is associated with the 'Mind' target. The 'Text Settings' section shows various formatting options like text encoding, line endings, and indentation.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

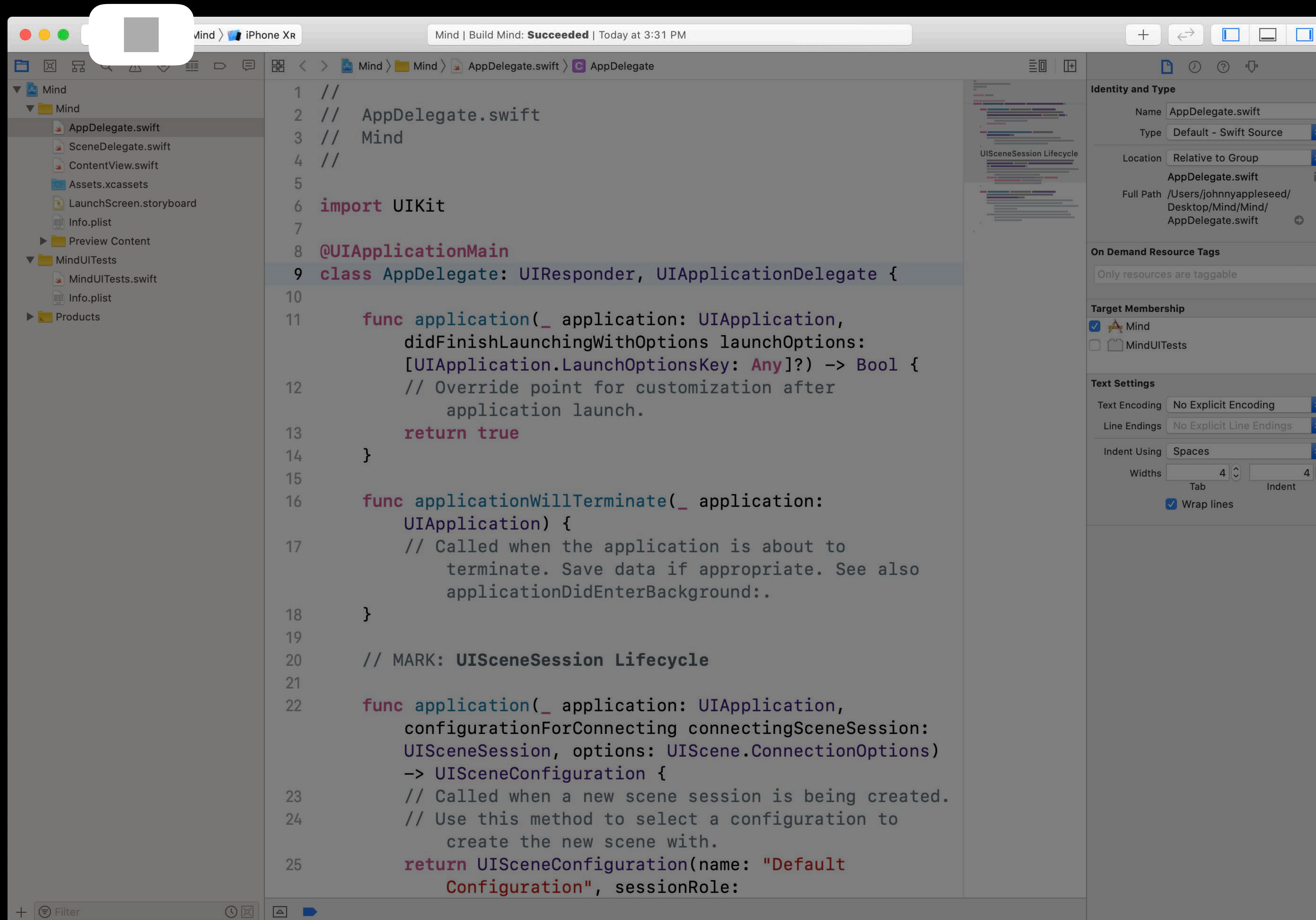
Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar



The image shows a screenshot of the Xcode IDE. At the top, a toolbar is highlighted with a white rounded rectangle. The toolbar contains several icons: a plus sign, a left-pointing arrow, a right-pointing arrow, a magnifying glass, a document icon, a trash can, and a refresh icon. Below the toolbar, the main editor area displays the Swift code for AppDelegate.swift. The code is as follows:

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

On the right side of the editor, the 'Identity and Type' panel is visible, showing the file name 'AppDelegate.swift', its type 'Default - Swift Source', and its location 'Relative to Group'. Below this, the 'Text Settings' panel is also visible, showing options for 'Text Encoding', 'Line Endings', 'Indent Using', and 'Wrap Lines'.

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application:didFinishLaunchingWithOptions and applicationWillTerminate. A comment marks the UISceneSession Lifecycle section, which contains the application:configurationForConnecting method. The right-hand side of the interface shows the Identity and Type inspector, which is currently displaying settings for AppDelegate.swift, including its name, type, location, and target membership.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application:didFinishLaunchingWithOptions and applicationWillTerminate. A comment marks the UISceneSession Lifecycle section, followed by the application:configurationForConnecting method.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

The right-hand side of the interface shows the 'Identity and Type' inspector. It displays the file name 'AppDelegate.swift', its type 'Default - Swift Source', and its location 'Relative to Group'. Below this, the 'On Demand Resource Tags' section is empty. The 'Target Membership' section shows that the target 'Mind' is selected. The 'Text Settings' section shows 'Text Encoding' as 'No Explicit Encoding', 'Line Endings' as 'No Explicit Line Endings', and 'Indent Using' as 'Spaces' with a width of 4. The 'Wrap lines' checkbox is checked.

Toolbar

Mind | Build Mind: **Succeeded** | Today at 3:31 PM

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                               Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application(_:application, didFinishLaunchingWithOptions:) and applicationWillTerminate(_:application:). A second application method is partially visible at the bottom. The right-hand side of the interface features the 'Identity and Type' inspector, which shows the file's name, type, location, and full path. Below this, the 'Target Membership' section shows that the file is associated with the 'Mind' target. The 'Text Settings' section at the bottom right shows options for text encoding, line endings, and indentation.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The left sidebar shows the project structure for 'Mind'. The right sidebar shows the 'Identity and Type' inspector for the selected file. A toolbar with three window icons is overlaid on the top right of the IDE.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

- Name: AppDelegate.swift
- Type: Default - Swift Source
- Location: Relative to Group
- App Delegate: AppDelegate.swift
- Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

- Text Encoding: No Explicit Encoding
- Line Endings: No Explicit Line Endings
- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Toolbar

The screenshot displays the Xcode IDE interface. The main editor shows the Swift code for AppDelegate.swift. The code includes comments, an import statement for UIKit, and two application lifecycle methods: application:didFinishLaunchingWithOptions and applicationWillTerminate. A comment marks the UISceneSession Lifecycle section, followed by the application:configurationForConnecting method.

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths Tab 4 Indent 4
 Wrap lines

Mind | Build Mind: **Succeeded** | Today at 3:31 PM

Mind > Mind > AppDelegate.swift > AppDelegate

```
1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:
```

Identity and Type

Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind
 MindUITests

Text Settings

Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths Tab: 4 Indent: 4
 Wrap lines

Mind | Build Mind: **Succeeded** | Today at 3:31 PM
+ ↩ 📄 📄 📄

Mind

- Mind
 - AppDelegate.swift
 - SceneDelegate.swift
 - ContentView.swift
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - Preview Content
 - MindUITests
 - MindUITests.swift
 - Info.plist
 - Products

```

1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                               Configuration", sessionRole:

```

UISceneSession Lifecycle

Identity and Type

Name: AppDelegate.swift

Type: Default - Swift Source

Location: Relative to Group

AppDelegate.swift

Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind

MindUITests

Text Settings

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: Tab: 4 Indent: 4

Wrap lines


```

1 //
2 // AppDelegate.swift
3 // Mind
4 //
5
6 import UIKit
7
8 @UIApplicationMain
9 class AppDelegate: UIResponder, UIApplicationDelegate {
10
11     func application(_ application: UIApplication,
12                     didFinishLaunchingWithOptions launchOptions:
13                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
14         // Override point for customization after
15         // application launch.
16         return true
17     }
18 }
19
20 // MARK: UISceneSession Lifecycle
21
22 func application(_ application: UIApplication,
23                 configurationForConnecting connectingSceneSession:
24                 UISceneSession, options: UIScene.ConnectionOptions)
25                 -> UISceneConfiguration {
26     // Called when a new scene session is being created.
27     // Use this method to select a configuration to
28     // create the new scene with.
29     return UISceneConfiguration(name: "Default
30                                 Configuration", sessionRole:

```

Search

Developer Documentation ⌘0

Xcode Help

- What's New in Xcode
- Release Notes
- Developer Account Help
- App Store Connect Help
- Report an Issue

Show Quick Help for Selected Item ^⌘?

Search Documentation for Selected Text ^⌘/

Identity and Type

Name AppDelegate.swift

Type Default - Swift Source

Location Relative to Group

AppDelegate.swift

Full Path /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

Text Encoding No Explicit Encoding

Line Endings No Explicit Line Endings

Indent Using Spaces

Widths Tab: 4 Indent: 4

Wrap lines


Xcode Help

Xcode Help

Q Search

- Welcome
- > Get started in Xcode
- > Xcode basics
- > Customize Xcode
- > Create playgrounds
- > Run your app on a device
- > Edit source code
- > Lay out user interfaces
- > Work with assets
- > Debug your app
- > Test code
- > Configure targets and builds
- > Work with Swift packages
- > Configure capabilities
- > Localize your app
- > Manage devices
- > Maintain signing assets
- > Manage source control
- > Archive, distribute, and test
- > Submit apps to the App Store

[More help for your Mac >](#)



Xcode Help

[Welcome to Xcode](#) | [Hide topics](#)

Demo

Summary

Summary

Creating a project using Xcode

Summary

Creating a project using Xcode

Choosing a template

Summary

Creating a project using Xcode

Choosing a template

Adding an app icon

Summary

Creating a project using Xcode

Choosing a template

Adding an app icon

Using source control

Summary

Creating a project using Xcode

Choosing a template

Adding an app icon

Using source control

#WWDC19

Writing and Navigating Source Code

Holly Borla, Xcode Engineer

Mind | Preview Mind: **Succeeded** | Today at 2:42 PM

Mind > iPhone XR

Mind > Mind > MeditationController.swift > formattedDuration

Mind

- README.md
- Mind
 - Mind.entitlements
 - AppDelegate.swift
 - MeditationController.swift
 - HealthStore.swift
 - HealthStoreFactory.swift
 - ContentView.swift
 - BackgroundView.swift
 - Icons.swift
 - Preview Assets.xcassets
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - Mind.xctestplan
- MindUITests
 - MindUITests.swift
 - Info.plist
- Products
- Frameworks

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21     var formattedDuration: String {
22         return
23             MeditationController.timeFormatter
24             .string(from: rawValue)!
25     }
26 }
27
28 // MARK: - Initialization
29
30 let healthStore: HealthStore
31 var duration: Duration
32
33 init(healthStore: HealthStore =
34     HealthStoreFactory.makeHealthStore()) {
35     self.healthStore = healthStore
36     self.duration = .threeMinutes
37 }
```

Identity and Type

Name: MeditationController.swift

Type: Default - Swift Source

Location: Relative to Group

MeditationController.swift

Full Path: /Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/MeditationController.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: 4 Tab 4 Indent

Wrap lines

Mind | iPhone XR Mind | Preview Mind: Succeeded | Today at 2:42 PM

Mind > Mind > MeditationController.swift > formattedDuration

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

Identity and Type

Name: MeditationController.swift
Type: Default - Swift Source
Location: Relative to Group
MeditationController.swift
Full Path: /Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/Mind/MeditationController.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind
 MindUITests

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: 4 (Tab) 4 (Indent)
 Wrap lines

Line Numbers

The image shows a screenshot of the Xcode IDE. The main editor window displays Swift code for a file named `MeditationController.swift`. The code is formatted with line numbers on the left side, ranging from 1 to 33. The code includes imports for `Foundation`, `SwiftUI`, and `Combine`. It defines a `MeditationController` class that inherits from `BindableObject`. Inside the class, there is an `enum Duration` with cases for `fiveSeconds`, `fifteenSeconds`, `thirtySeconds`, `oneMinute`, `threeMinutes`, `fiveMinutes`, `tenMinutes`, and `fifteenMinutes`. A `formattedDuration` property is defined as a `String` that returns the result of `MeditationController.timeFormatter.string(from: rawValue)!`. The class also has an `init` method that initializes `healthStore` and `duration`.

On the right side of the editor, the 'Identity and Type' panel is visible. It shows the file name `MeditationController.swift` and its type `Default - Swift Source`. The location is set to `Relative to Group`. The full path is `/Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/Mind/MeditationController.swift`. The 'On Demand Resource Tags' section is empty. The 'Target Membership' section shows that the file is targeted to the `Mind` target and not to `MindUITests`. The 'Text Settings' section shows that the text encoding is `No Explicit Encoding`, line endings are `No Explicit Line Endings`, and the indent is set to `Spaces` with a width of `4`. The `Wrap lines` option is checked.

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```


Minimap

The image shows a screenshot of the Xcode IDE interface. The main editor displays a Swift file named `MeditationController.swift`. The code is as follows:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

On the right side of the editor, there is a vertical minimap showing a color-coded overview of the code structure. Below the editor, the 'Identity and Type' panel is visible, showing the file name `MeditationController.swift` and its location. The 'Text Settings' panel is also visible, showing options for text encoding, line endings, and indentation.

Jump Bar

The screenshot shows the Xcode IDE with a Swift file named `MeditationController.swift` open. The Jump Bar is visible at the top of the editor, showing the current file and the selected line. The code in the editor is as follows:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right-hand pane shows the Identity and Type settings for the selected file, including Name, Type, Location, Full Path, On Demand Resource Tags, Target Membership, and Text Settings.

Jump Bar

The screenshot shows the Xcode IDE interface. The top toolbar includes a Jump Bar with a magnifying glass icon and a search field. The main editor displays the following Swift code:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right sidebar shows the Identity and Type panel for the selected file, MeditationController.swift. The panel includes fields for Name, Type, Location, Full Path, On Demand Resource Tags, Target Membership, and Text Settings.

Jump Bar

The image shows a screenshot of the Xcode IDE. The main editor displays a Swift file named `MeditationController.swift`. A white Jump Bar is positioned over the code, showing a blue 'P' icon and the text `formattedDuration`. The code in the background includes:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right-hand side of the Xcode window shows the 'Identity and Type' inspector for the selected file, displaying details such as Name, Type, Location, and Full Path. The 'Text Settings' section is also visible, showing options for Text Encoding, Line Endings, and Indent Using (Spaces).

Jump Bar

The screenshot shows the Xcode IDE interface. The top toolbar includes a Jump Bar with a magnifying glass icon and a search field. The main editor displays the following Swift code:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right sidebar shows the 'Identity and Type' panel for the selected file, 'MeditationController.swift', with settings for Name, Type, Location, Full Path, On Demand Resource Tags, Target Membership, and Text Settings.

Jump Bar

The screenshot shows the Xcode IDE interface. The main editor window displays the Swift file `MeditationController.swift`. A Jump Bar is visible at the top of the editor, showing the current file and the selected line (`formattedDuration`).

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right-hand pane shows the Identity and Type settings for the selected file, including Name, Type, Location, Full Path, and Text Settings.

Jump Bar

The image shows a screenshot of the Xcode IDE. The main editor displays the Swift source code for `MeditationController.swift`. A white Jump Bar is overlaid on the code, highlighting the `formattedDuration` property. The Jump Bar contains a hamburger menu icon on the left and a plus sign icon on the right. The code in the background is as follows:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right-hand pane shows the Properties Inspector for the selected `formattedDuration` property. The Name is `MeditationController.swift` and the Type is `Default - Swift Source`. The Location is `Relative to Group` and the Full Path is `/Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/MeditationController.swift`. The Target Membership section shows that the property is associated with the `Mind` target. The Text Settings section shows that the property is indented using 4 spaces and that the `Wrap lines` option is checked.

Jump Bar

The screenshot shows the Xcode IDE with a Swift file named `MeditationController.swift` open. The Jump Bar is visible at the top of the editor, showing the current file and the selected line. The code in the file is as follows:

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21         var formattedDuration: String {
22             return
23                 MeditationController.timeFormatter
24                 .string(from: rawValue)!
25         }
26     }
27 }
28
29 // MARK: - Initialization
30
31 let healthStore: HealthStore
32 var duration: Duration
33
34 init(healthStore: HealthStore =
35     HealthStoreFactory.makeHealthStore()) {
36     self.healthStore = healthStore
37     self.duration = .threeMinutes
38 }
```

The right-hand pane shows the Identity and Type settings for the selected file, including Name, Type, Location, Full Path, On Demand Resource Tags, Target Membership, and Text Settings.

Mind | iPhone XR Mind | Preview Mind: Succeeded | Today at 2:42 PM

Mind > Mind > MeditationController.swift > formattedDuration

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21     var formattedDuration: String {
22         return
23             MeditationController.timeFormatter
24             .string(from: rawValue)!
25     }
26
27     // MARK: - Initialization
28
29     let healthStore: HealthStore
30     var duration: Duration
31
32     init(healthStore: HealthStore =
33         HealthStoreFactory.makeHealthStore()) {
34         self.healthStore = healthStore
35         self.duration = .threeMinutes
36     }
37 }
```

Identity and Type

Name: MeditationController.swift
Type: Default - Swift Source
Location: Relative to Group
MeditationController.swift
Full Path: /Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/Mind/MeditationController.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

Mind
 MindUITests

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: 4 Tab 4 Indent
 Wrap lines

Mind | Preview Mind: **Succeeded** | Today at 2:42 PM

Mind > iPhone XR

Mind > Mind > MeditationController.swift > formattedDuration

Mind

- README.md
- Mind
 - Mind.entitlements
 - AppDelegate.swift
 - MeditationController.swift
 - HealthStore.swift
 - HealthStoreFactory.swift
 - ContentView.swift
 - BackgroundView.swift
 - Icons.swift
 - Preview Assets.xcassets
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - Mind.xctestplan
- MindUITests
 - MindUITests.swift
 - Info.plist
- Products
- Frameworks

```
1 //
2 // MeditationController.swift
3 // Mind
4 //
5
6 import Foundation
7 import SwiftUI
8 import Combine
9
10 class MeditationController: BindableObject {
11     enum Duration: TimeInterval, CaseIterable, Hashable {
12         case fiveSeconds = 5
13         case fifteenSeconds = 15
14         case thirtySeconds = 30
15         case oneMinute = 60
16         case threeMinutes = 180
17         case fiveMinutes = 300
18         case tenMinutes = 600
19         case fifteenMinutes = 900
20
21     var formattedDuration: String {
22         return
23             MeditationController.timeFormatter
24             .string(from: rawValue)!
25     }
26 }
27
28 // MARK: - Initialization
29
30 let healthStore: HealthStore
31 var duration: Duration
32
33 init(healthStore: HealthStore =
34     HealthStoreFactory.makeHealthStore()) {
35     self.healthStore = healthStore
36     self.duration = .threeMinutes
37 }
```

Identity and Type

Name: MeditationController.swift

Type: Default - Swift Source

Location: Relative to Group

MeditationController.swift

Full Path: /Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/MeditationController.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

- Mind
- MindUITests

Text Settings

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: 4 Tab 4 Indent

Wrap lines

9:41



Welcome to Mind!



- 00:00
- 00:15
- 00:30
- 01:00
- 03:00
- 05:00
- 10:00
- 15:00

3
Sessions

25
Total Sec

8.3
Average Sec



THE SWIFT PROGRAMMING LANGUAGE
SWIFT 5

WELCOME TO SWIFT

About Swift
Version Compatibility

A Swift Tour

LANGUAGE GUIDE

LANGUAGE REFERENCE

REVISION HISTORY

← RETURN TO SWIFT.ORG

ON THIS PAGE

A Swift Tour

Tradition suggests that the first program in a new language should print the words “Hello, world!” on the screen. In Swift, this can be done in a single line:

```
1 print("Hello, world!")  
2 // Prints "Hello, world!"
```

If you have written code in C or Objective-C, this syntax looks familiar to you—in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like input/output or string handling. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you enough information to start writing code in Swift by showing you how to accomplish a variety of programming tasks. Don’t worry if you don’t understand something—everything introduced in this tour is explained in detail in the rest of this book.

NOTE

For the best experience, open this chapter as a playground in Xcode. Playgrounds allow you to edit the code listings and see the result immediately.

[Download Playground](#)

Simple Values

Use `let` to make a constant and `var` to make a variable. The value of a constant doesn’t need to be known at compile time, but you must assign it a value exactly once. This means you can use constants to name a value that you determine once but use in many places.

```
1 var myVariable = 42
```




THE SWIFT PROGRAMMING LANGUAGE

SWIFT 5

WELCOME TO SWIFT

About Swift

Version Compatibility

A Swift Tour

LANGUAGE GUIDE

LANGUAGE REFERENCE

REVISION HISTORY

[← RETURN TO SWIFT.ORG](#)

A Swift Tour

Tradition suggests that the first program in a new language should print the words “Hello, world!” on the screen. In Swift, this can be done in a single line:

```
1 print("Hello, world!")
2 // Prints "Hello, world!"
```

If you have written code in C or Objective-C, this syntax looks familiar to you—in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like input/output or string handling. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you enough information to start writing code in Swift by showing you how to accomplish a variety of programming tasks. Don’t worry if you don’t understand something—everything introduced in this tour is explained in detail in the rest of this book.

NOTE

For the best experience, open this chapter as a playground in Xcode. Playgrounds allow you to edit the code listings and see the result immediately.

[Download Playground](#)


Simple Values

Use `let` to make a constant and `var` to make a variable. The value of a constant doesn’t need to be known at compile time, but you must assign it a value exactly once. This means you can use constants to name a value that you determine once but use in many places.

```
1 var myVariable = 42
```


docs.swift.org

ON THIS PAGE



A Swift Tour

Tradition suggests that the first program in a new language should print the words “Hello, world!” on the screen. In Swift, this can be done in a single line:

```
1 print("Hello, world!")
2 // Prints "Hello, world!"
```

If you have written code in C or Objective-C, this syntax looks familiar to you—in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like input/output or string handling. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you enough information to start writing code in Swift by showing you how to accomplish a variety of programming tasks. Don’t worry if you don’t understand something—everything introduced in this tour is explained in detail in the rest of this book.

← RETURN TO SWIFT.ORG

THE SWIFT PROGRAMMING LANGUAGE
SWIFT 5

WELCOME TO SWIFT

- About Swift
- Version Compatibility
- A Swift Tour**

LANGUAGE GUIDE

LANGUAGE REFERENCE

REVISION HISTORY

NOTE

For the best experience, open this chapter as a playground in Xcode. Playgrounds allow you to edit the code listings and see the result immediately.

[Download Playground](#)

Simple Values

Use `let` to make a constant and `var` to make a variable. The value of a constant doesn’t need to be known at compile time, but you must assign it a value exactly once. This means you can use constants to name a value that you determine once but use in many places.

```
1 var myVariable = 42
```




THE SWIFT PROGRAMMING LANGUAGE

SWIFT 5

WELCOME TO SWIFT

About Swift

Version Compatibility

A Swift Tour

LANGUAGE GUIDE

LANGUAGE REFERENCE

REVISION HISTORY

[← RETURN TO SWIFT.ORG](#)

A Swift Tour

Tradition suggests that the first program in a new language should print the words “Hello, world!” on the screen. In Swift, this can be done in a single line:

```
1 print("Hello, world!")
2 // Prints "Hello, world!"
```

If you have written code in C or Objective-C, this syntax looks familiar to you—in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like input/output or string handling. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you enough information to start writing code in Swift by showing you how to accomplish a variety of programming tasks. Don’t worry if you don’t understand something—everything introduced in this tour is explained in detail in the rest of this book.

NOTE

For the best experience, open this chapter as a playground in Xcode. Playgrounds allow you to edit the code listings and see the result immediately.

[Download Playground](#)


Simple Values

Use `let` to make a constant and `var` to make a variable. The value of a constant doesn’t need to be known at compile time, but you must assign it a value exactly once. This means you can use constants to name a value that you determine once but use in many places.

```
1 var myVariable = 42
```


docs.swift.org

ON THIS PAGE



Swift

THE SWIFT PROGRAMMING LANGUAGE
SWIFT 5

WELCOME TO SWIFT

- About Swift
- Version Compatibility
- A Swift Tour**

LANGUAGE GUIDE

LANGUAGE REFERENCE

REVISION HISTORY

[← RETURN TO SWIFT.ORG](#)

A Swift Tour

Tradition suggests that the first program in a new language should print the words “Hello, world!” on the screen. In Swift, this can be done in a single line:

```
1 print("Hello, world!")
2 // Prints "Hello, world!"
```

If you have written code in C or Objective-C, this syntax looks familiar to you—in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like input/output or string handling. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you an easy information to start writing code in Swift by showing you how to accomplish a variety of programming tasks. Don't worry if you don't understand something—everything introduced in this tour is explained in detail in the rest of this book.

NOTE

For the best experience, open this chapter as a playground in Xcode. Playgrounds allow you to edit the code listings and see the result immediately.

[Download Playground](#)

Simple Values

Use `let` to make a constant and `var` to make a variable. The value of a constant doesn’t need to be known at compile time, but you must assign it a value exactly once. This means you can use constants to name a value that you determine once but use in many places.

```
1 var myVariable = 42
```

docs.swift.org/swift-book/GettingStarted/GettingStarted.html

Demo

Summary

Summary

Interacting with the Canvas

Summary

Interacting with the Canvas

Using code completion

Summary

Interacting with the Canvas

Using code completion

Editing structures from the Action Menu

Summary

Interacting with the Canvas

Using code completion

Editing structures from the Action Menu

Using Fix-Its

Summary

Interacting with the Canvas

Using code completion

Editing structures from the Action Menu

Using Fix-Its

What's New in Swift

WWDC 2019

Introducing SwiftUI: Building Your First App

WWDC 2019

#WWDC19

Running and Debugging

Honza Dvorsky, Xcode Engineer

Demo

Summary

Summary

Running and debugging with the Simulator

Summary

Running and debugging with the Simulator

Running on a device

Summary

Running and debugging with the Simulator

Running on a device

Adding a Swift Package

Summary

Running and debugging with the Simulator

Running on a device

Adding a Swift Package

Advanced Debugging with Xcode and LLDB

WWDC 2018

Adopting Swift Packages in Xcode

WWDC 2019

#WWDC19

Working with Packages and Frameworks

Holly Borla, Xcode Engineer

Demo



- ABOUT SWIFT
- BLOG
- DOWNLOAD
- GETTING STARTED
- DOCUMENTATION
- SOURCE CODE
- COMMUNITY
- CONTRIBUTING
- CONTINUOUS INTEGRATION
- SOURCE COMPATIBILITY
- FOCUS AREAS**
- GSOC 2019
- SERVER WORK GROUP
- PROJECTS**
- COMPILER AND STANDARD LIBRARY
- PACKAGE MANAGER
- CORE LIBRARIES
- REPL, DEBUGGER & PLAYGROUNDS

API Design Guidelines

To facilitate use as a quick reference, the details of many guidelines can be expanded individually. Details are never hidden when this page is printed. [Expand all details now](#)

Table of Contents

- [Fundamentals](#)
- [Naming](#)
 - [Promote Clear Usage](#)
 - [Strive for Fluent Usage](#)
 - [Use Terminology Well](#)
- [Conventions](#)
 - [General Conventions](#)
 - [Parameters](#)
 - [Argument Labels](#)
- [Special Instructions](#)

Fundamentals

- **Clarity at the point of use** is your most important goal. Entities such as methods and properties are declared only once but *used* repeatedly. Design APIs to make those uses clear and concise. When evaluating a design, reading a declaration is seldom sufficient; always examine a use case to make sure it looks clear in context.



API Design Guidelines

- ABOUT SWIFT
- BLOG
- DOWNLOAD
- GETTING STARTED
- DOCUMENTATION
- SOURCE CODE
- COMMUNITY
- CONTRIBUTING
- CONTINUOUS INTEGRATION
- SOURCE COMPATIBILITY
- FOCUS AREAS**
- GSOC 2019
- SERVER WORK GROUP
- PROJECTS**
- COMPILER AND STANDARD LIBRARY
- PACKAGE MANAGER
- CORE LIBRARIES
- REPL, DEBUGGER & PLAYGROUNDS

To facilitate use as a quick reference, the details of many guidelines can be expanded individually. Details are never hidden when this page is printed. [Expand all details now](#)

Table of Contents

- [Fundamentals](#)
- [Naming](#)
 - [Strive for Fluent Usage](#)
 - [Use Terminology Well](#)
- [Conventions](#)
 - [General Conventions](#)
 - [Parameters](#)
 - [Argument Labels](#)
- [Special Instructions](#)

swift.org/documentation/api-design-guidelines/

Fundamentals

- **Clarity at the point of use** is your most important goal. Entities such as methods and properties are declared only once but *used* repeatedly. Design APIs to make those uses clear and concise. When evaluating a design, reading a declaration is seldom sufficient; always examine a use case to make sure it looks clear in context.

Summary

Summary

Importing a Swift Package

Summary

Importing a Swift Package

Creating a framework

Summary

Importing a Swift Package

Creating a framework

Reading Developer Documentation

Summary

Importing a Swift Package

Creating a framework

Reading Developer Documentation

Writing documentation comments

#WWDC19

Testing and Distribution

Honza Dvorsky, Xcode Engineer

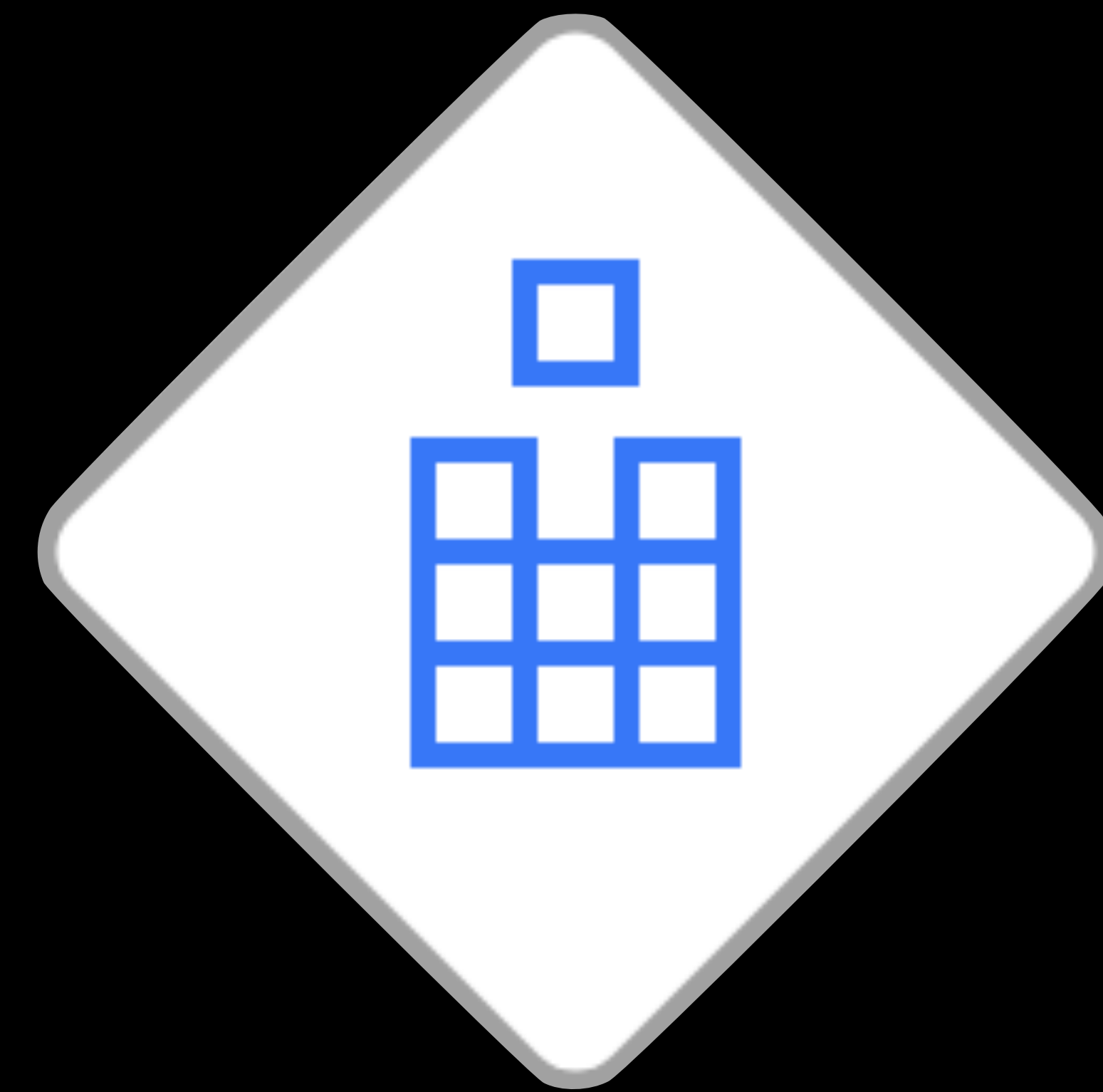
Tests



Tests

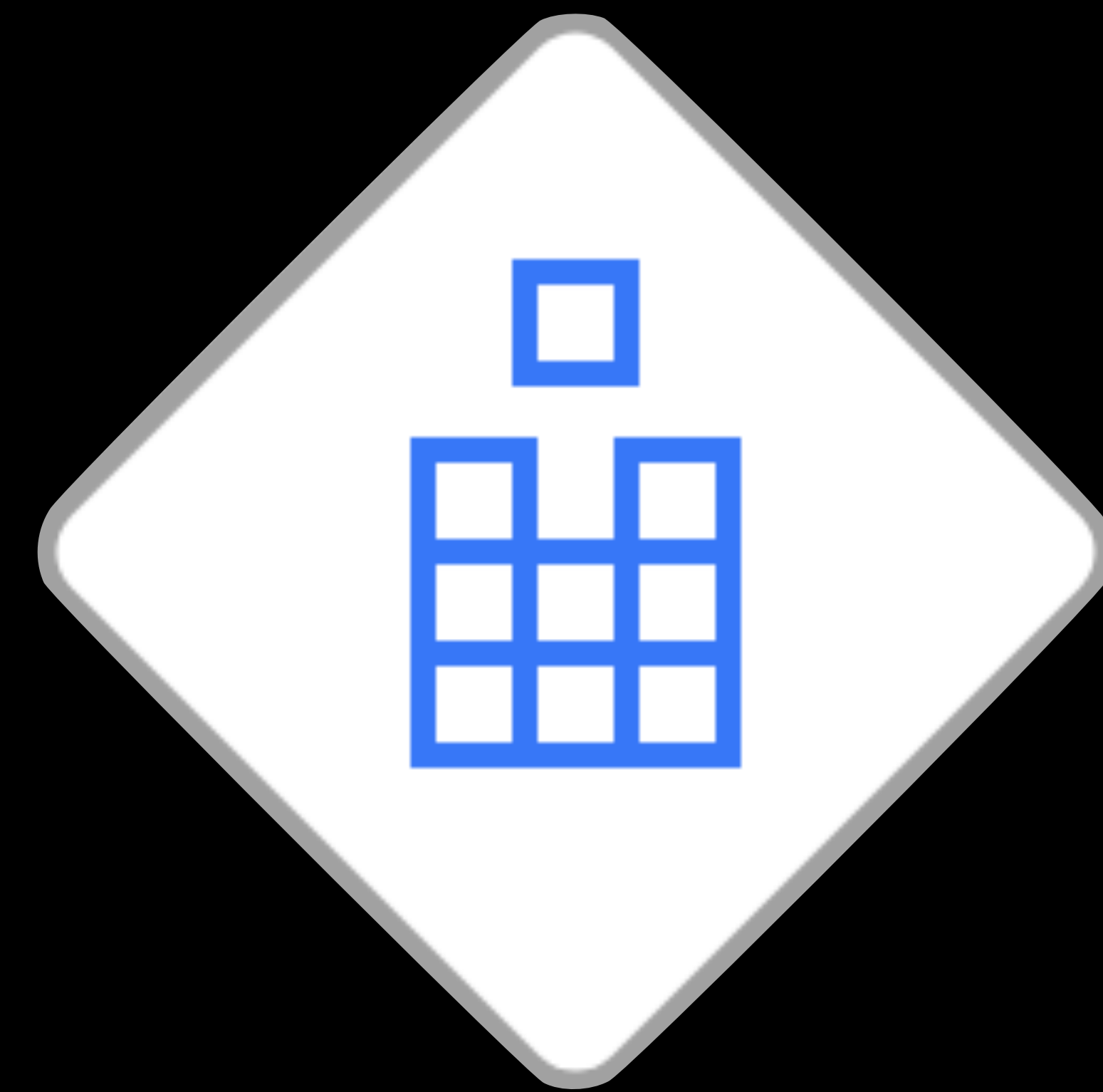


Tests

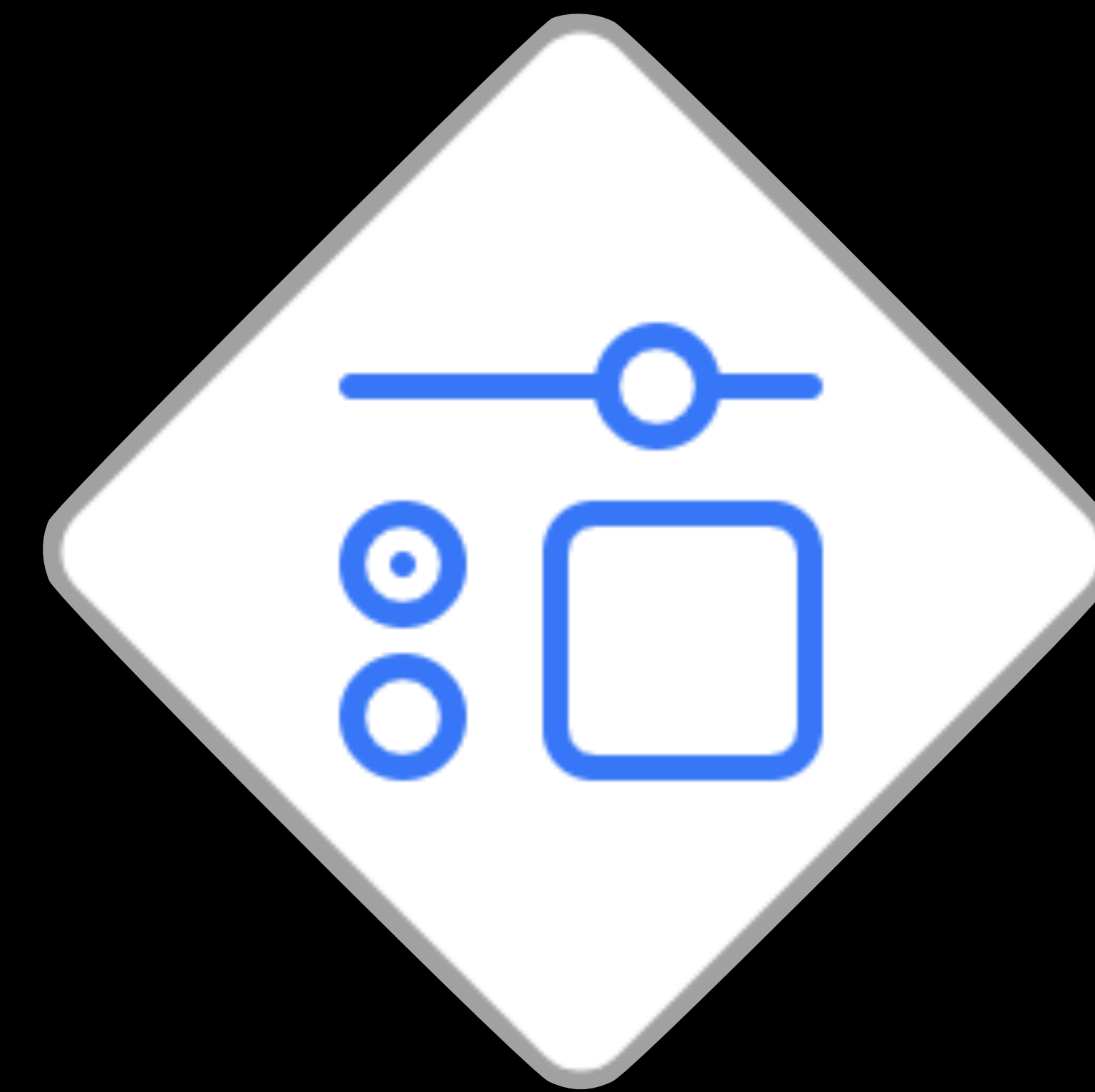


Unit

Tests



Unit



UI

Demo

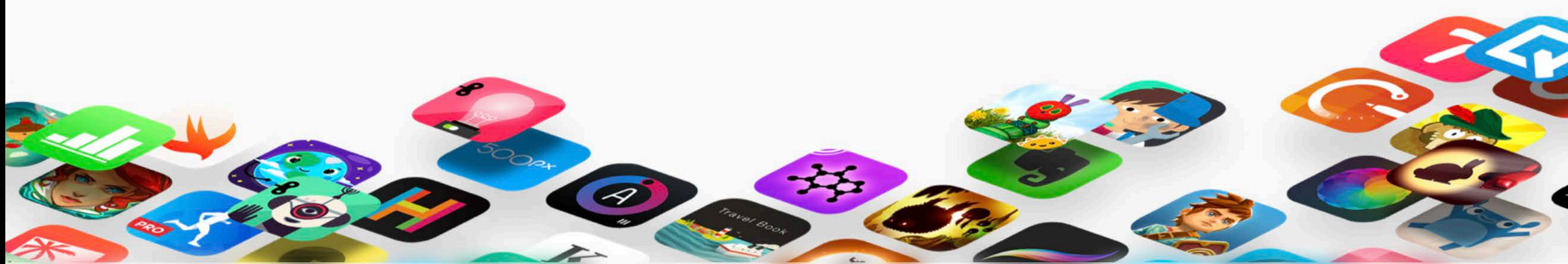


App Store Connect

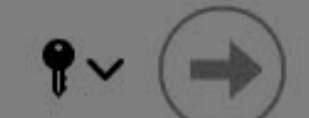
  

Remember me

[Forgot Apple ID or password?](#)



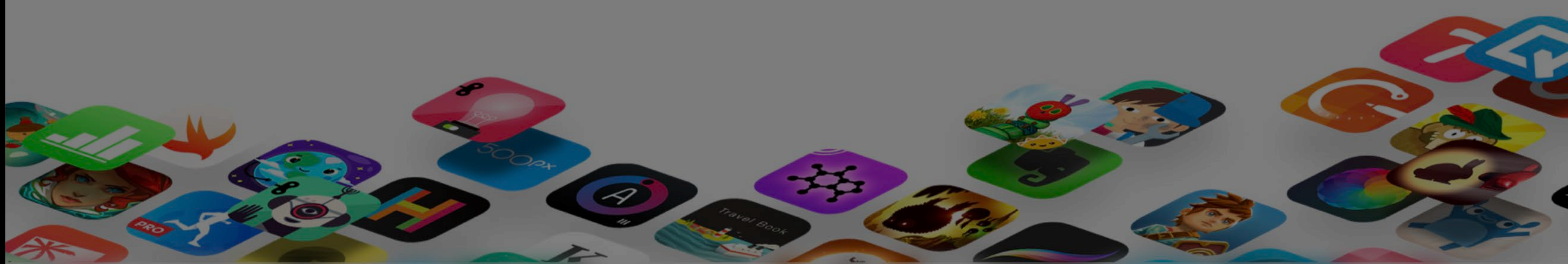
App Store Connect

Remember me

[Forgot Apple ID or password?](#)

appstoreconnect.apple.com



Summary

Summary

Writing unit and UI tests

Summary

Writing unit and UI tests

Editing Test Plans

Summary

Writing unit and UI tests

Editing Test Plans

Viewing the Test Report

Summary

Writing unit and UI tests

Editing Test Plans

Viewing the Test Report

Archiving and uploading using the Organizer

Summary

Writing unit and UI tests

Editing Test Plans

Viewing the Test Report

Archiving and uploading using the Organizer

Testing in Xcode

Thursday, 11:00AM

What's New in Signing for Xcode and Xcode Server

WWDC 2017

Creating a New Project

Writing and Navigating Source Code

Running and Debugging

Working with Packages and Frameworks

Testing and Distribution

More Information

developer.apple.com/wwdc19/404

Getting Started with Xcode Lab

Tuesday, 12:00PM

