

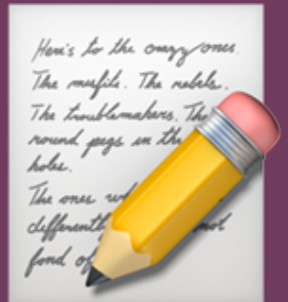
Swift Package all the things!



Federico Zanetello

★★★★★ fivestars.blog • @zntfdr


fivestars.blog
WWDCNotes.com
Bangkok Metro



What's a Swift Package?

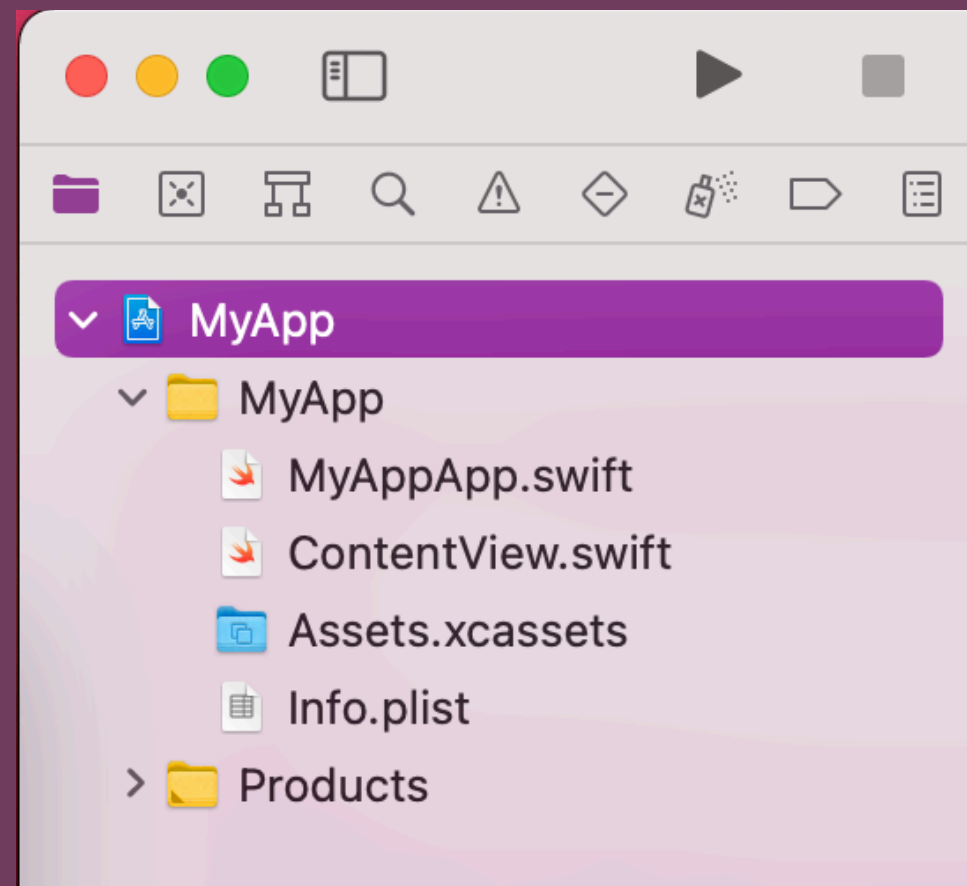
- New way to build frameworks
- Swift project since 2015
- Xcode 11 ❤️ iOS Apps
- Swift, Objective-C, Objective-C++, C, or C++
- Can bundle code, binaries, assets, ...

Why should we use one?

- modularization
- single-responsibility
- testability
- share between different projects/platforms
- blessed by 
- fun
- easy to use
- ...much more

Getting
Started

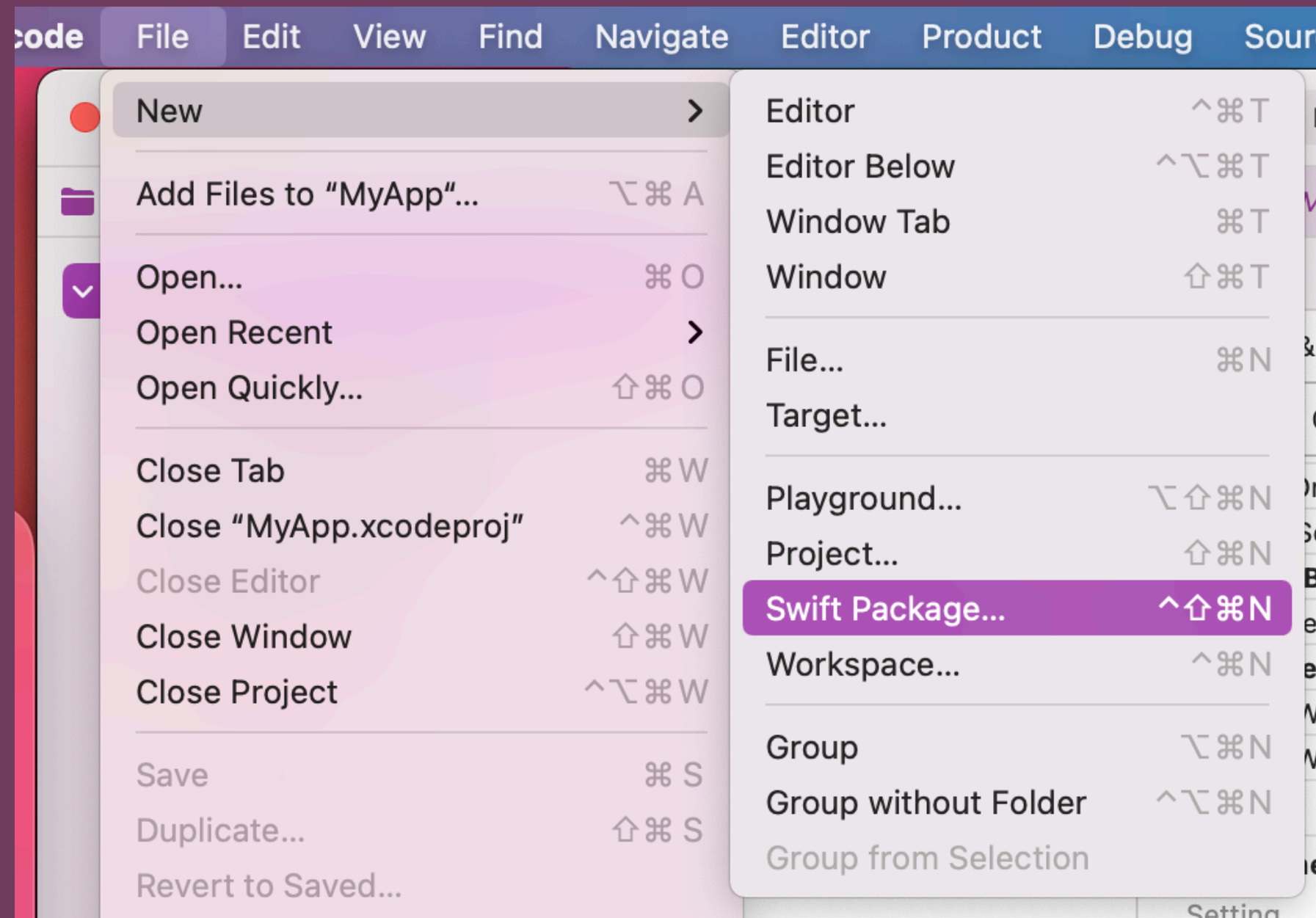
MyApp



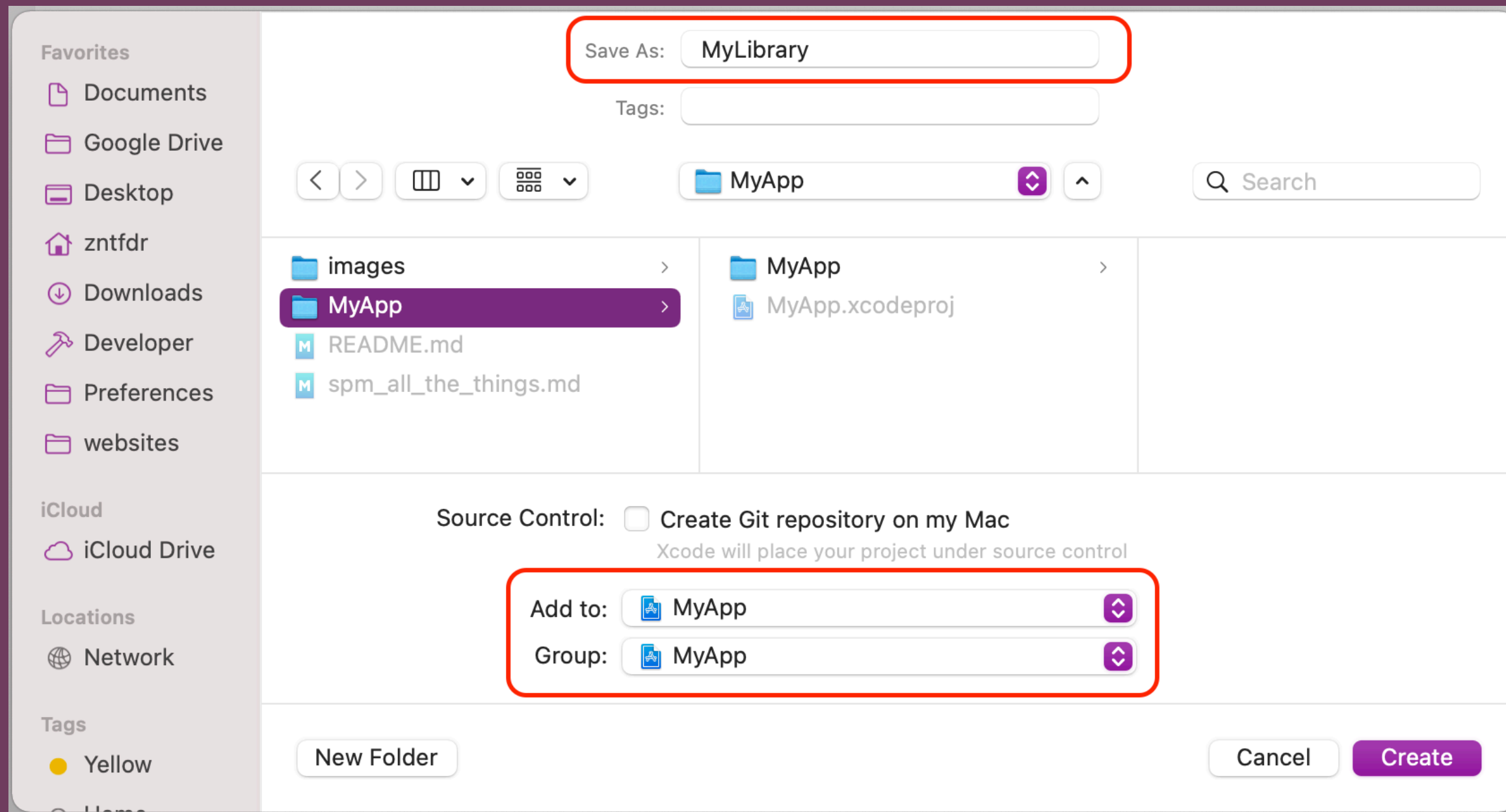
Create a Swift Package

1. Open your project
2. File > New > Swift Package...
3. Name it and add it to your project

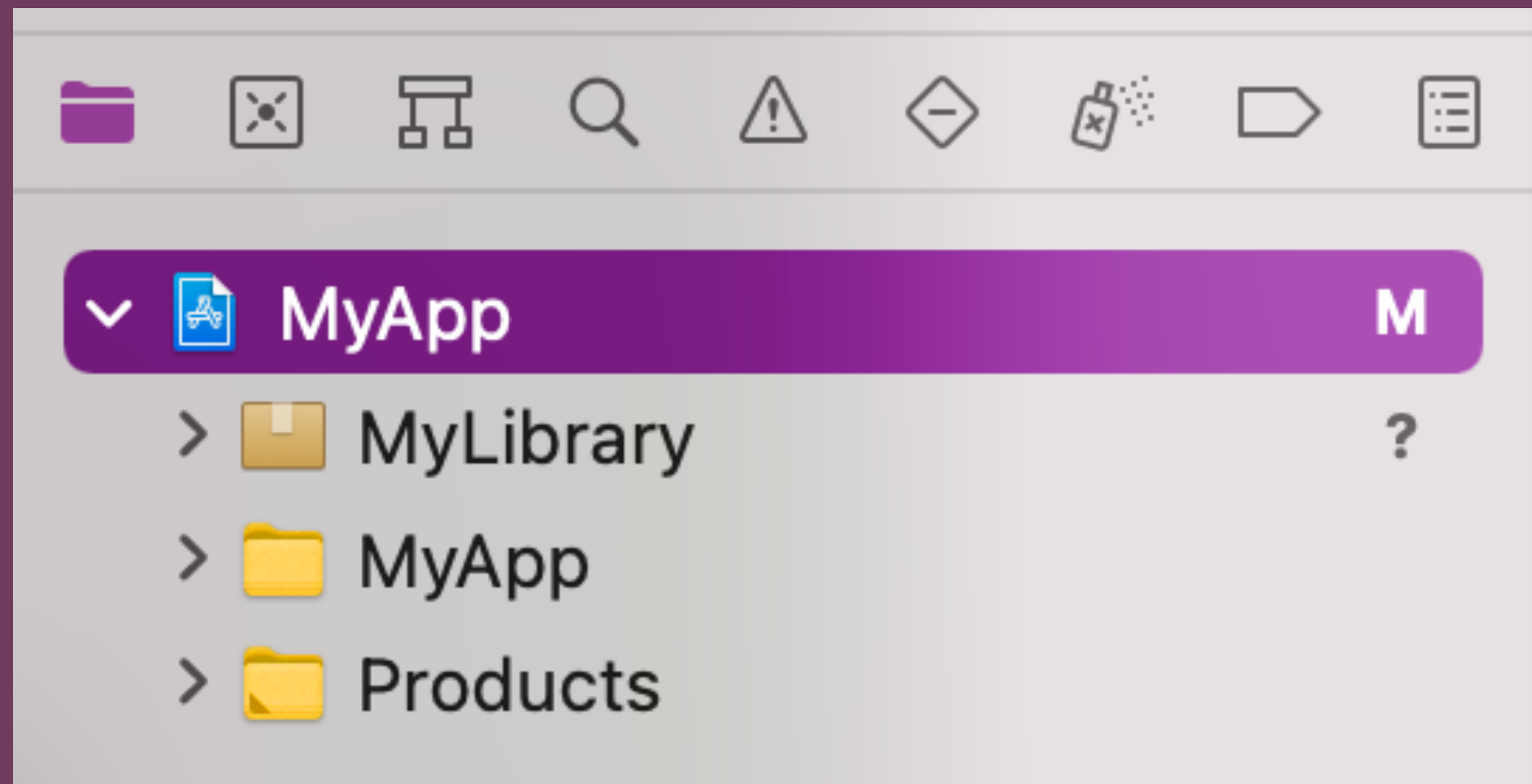
Create a Swift Package (1/2)



Create a Swift Package (2/2)



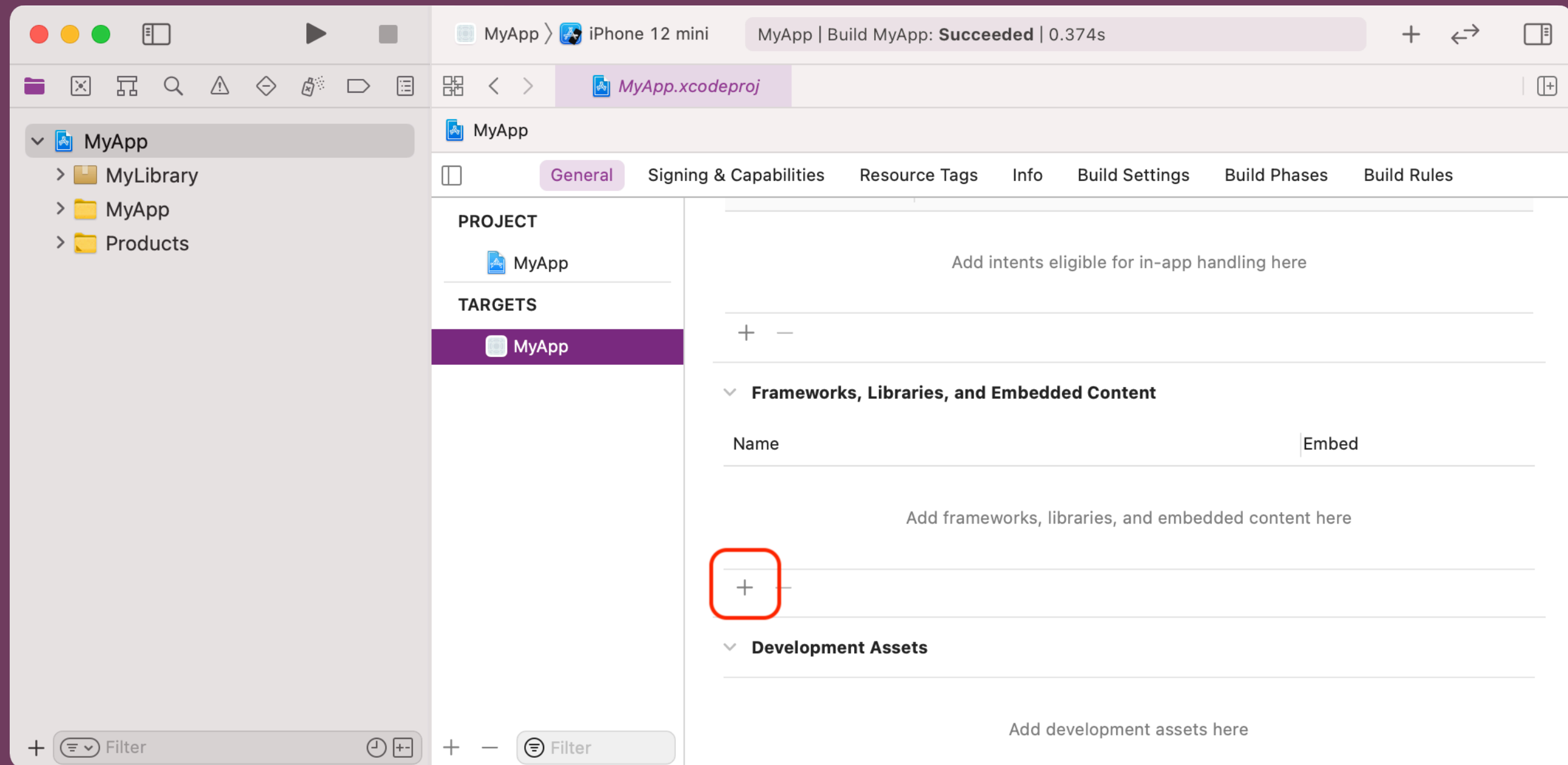
Done!



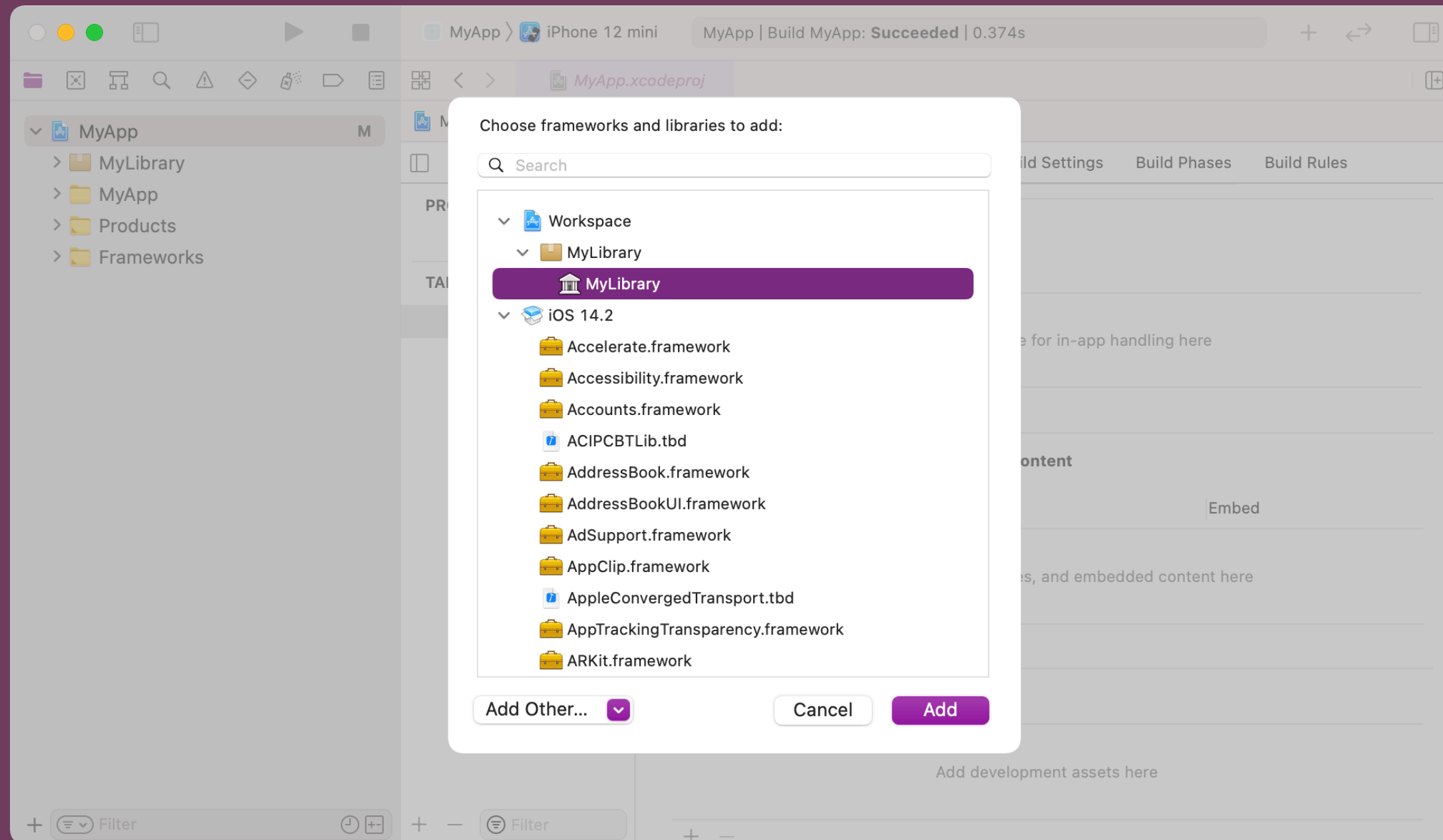
Add Package to your App

1. Open app Project Editor
2. Add the package under Frameworks, Libraries, and Embedded Content

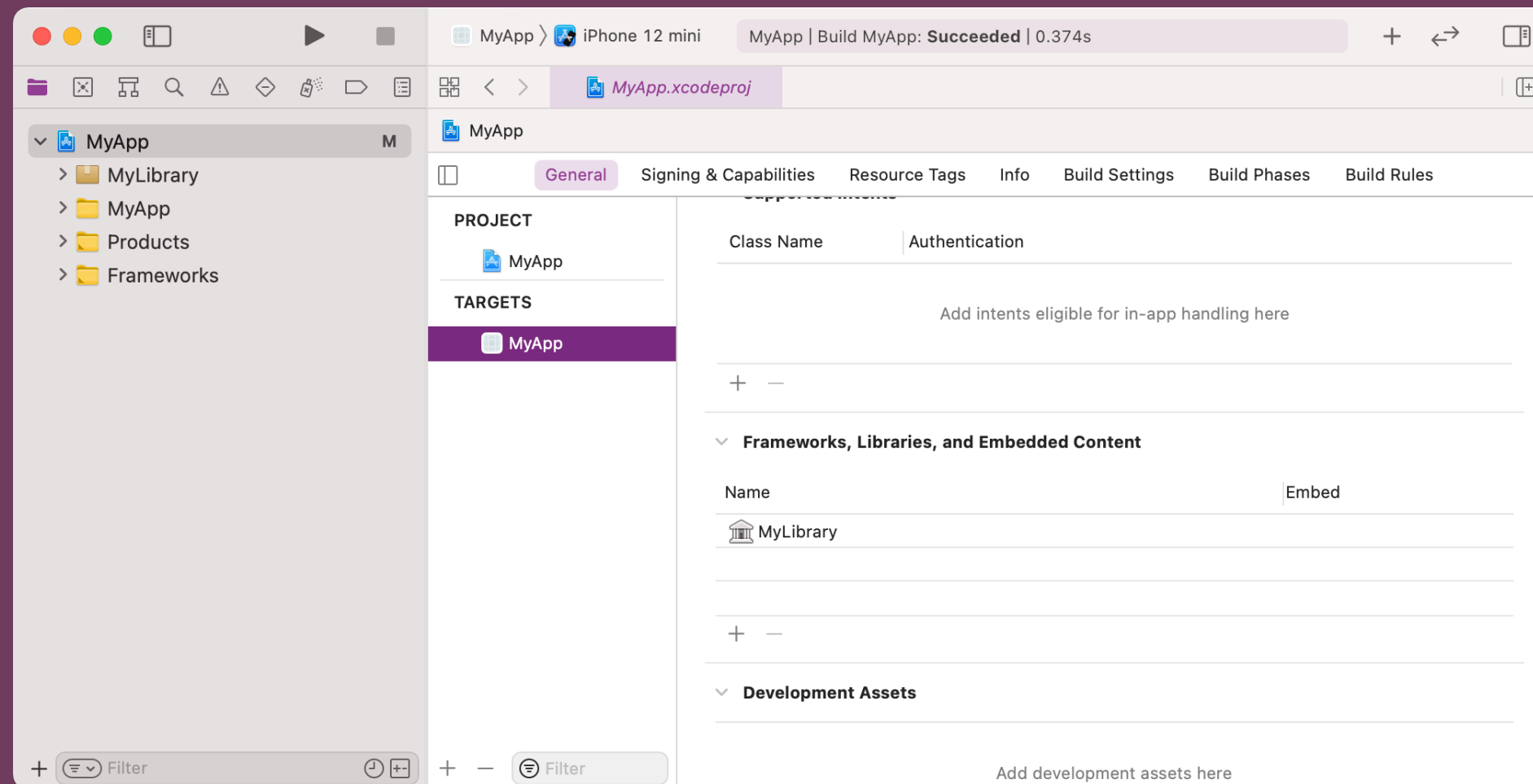
Add Package to your App (1/2)



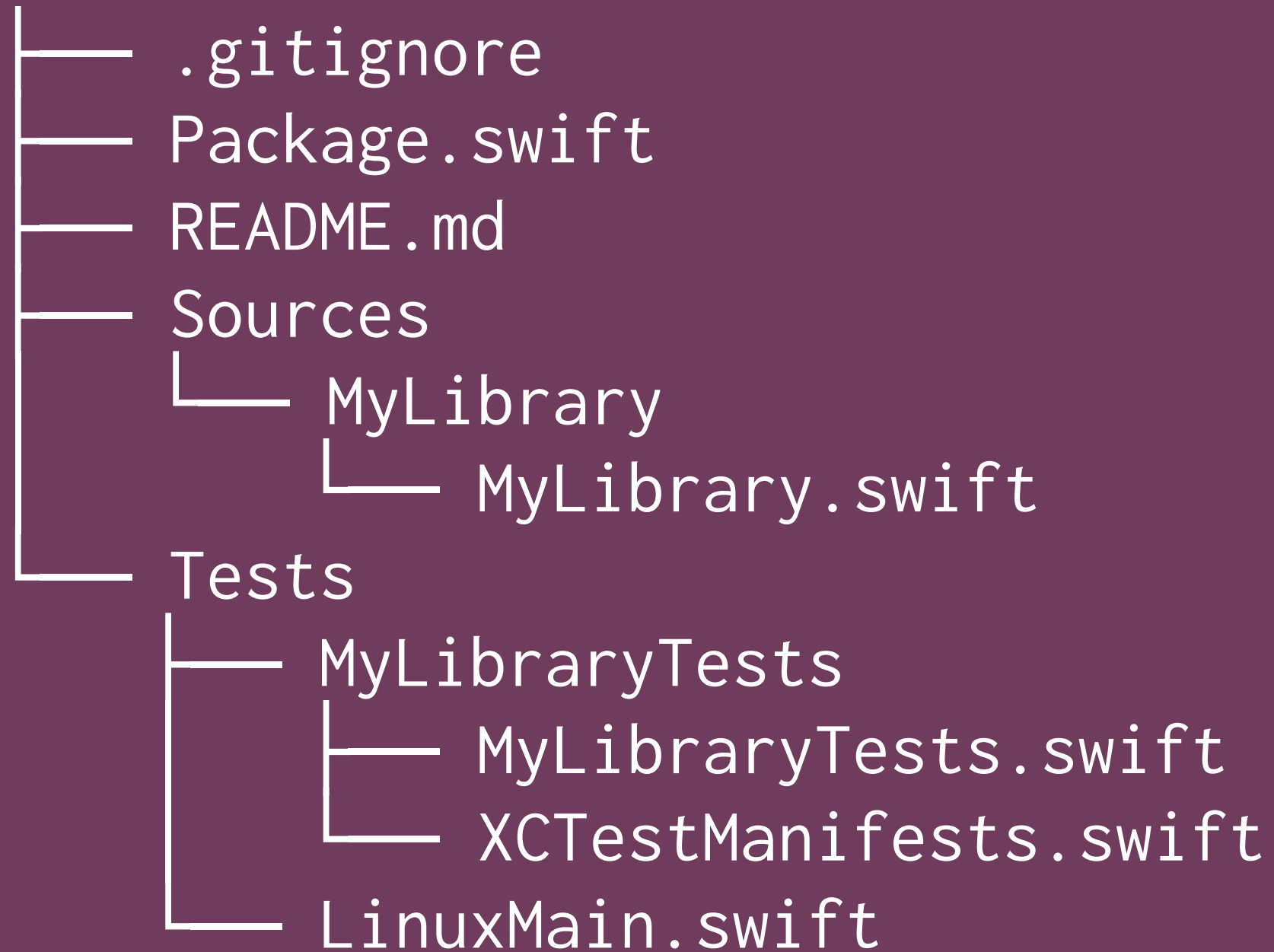
Add Package to your App (2/2)



Done!



The Package Structure



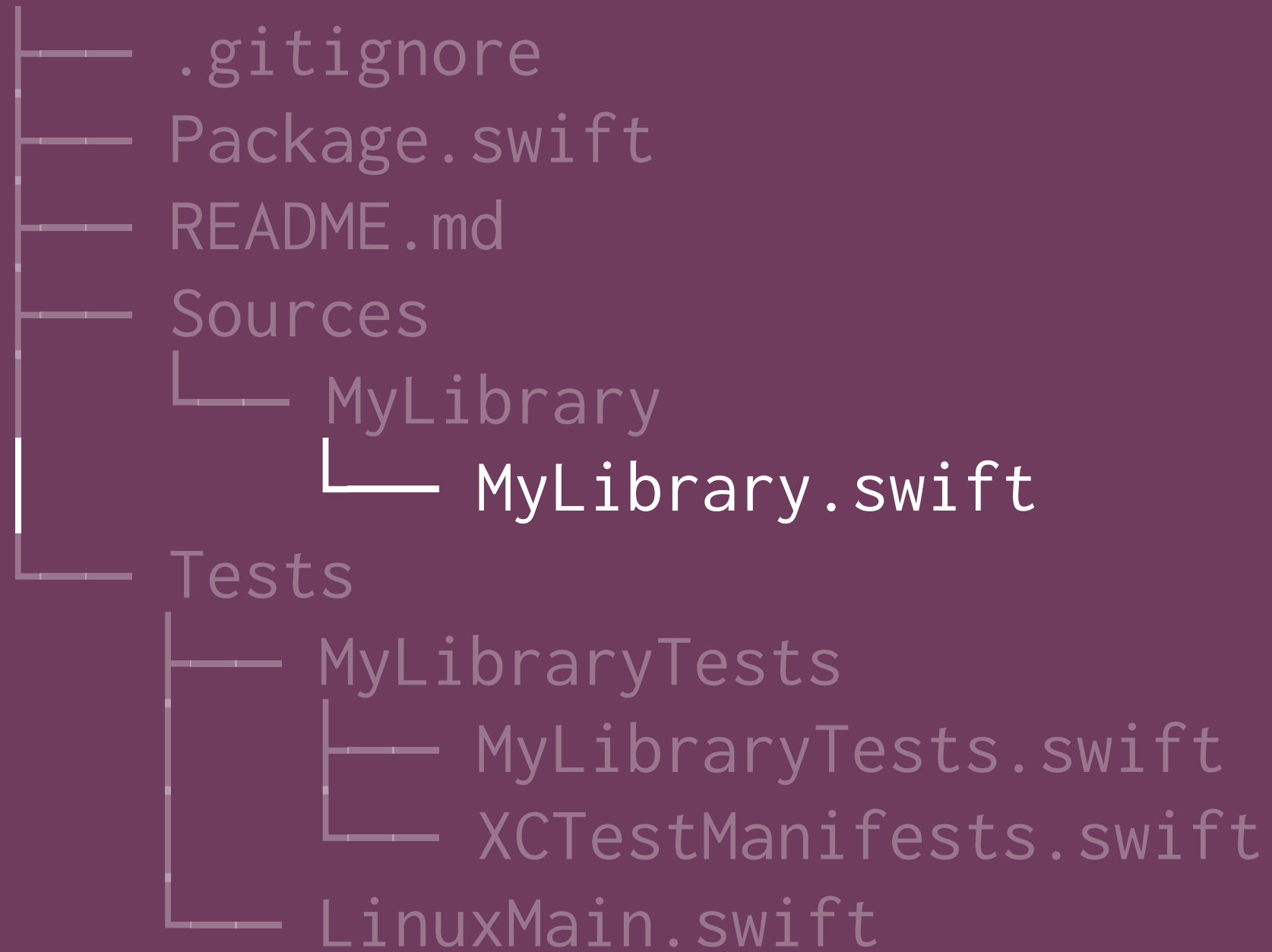
Package.swift

```
// swift-tools-version:5.3

import PackageDescription

let package = Package(
    name: "MyLibrary",
    products: [
        .library(name: "MyLibrary", targets: ["MyLibrary"]),
    ],
    dependencies: [],
    targets: [
        .target(name: "MyLibrary", dependencies: []),
        .testTarget(name: "MyLibraryTests", dependencies: ["MyLibrary"]),
    ]
)
```


The Package Structure



MyLibrary.swift

```
struct MyLibrary {  
    var text = "Hello, World!"  
}
```

MyLibrary.swift

```
import SwiftUI

public struct MyCoolView: View {

    public init() {}

    public var body: some View {
        Text("This view is totally cool")
    }
}
```

Package.swift

```
// swift-tools-version:5.3

import PackageDescription

let package = Package(
    name: "MyLibrary",
    platforms: [.iOS(.v13)],
    products: [
        .library(name: "MyLibrary", targets: ["MyLibrary"]),
    ],
    dependencies: [],
    targets: [
        .target(name: "MyLibrary", dependencies: []),
        .testTarget(name: "MyLibraryTests", dependencies: ["MyLibrary"]),
    ]
)
```

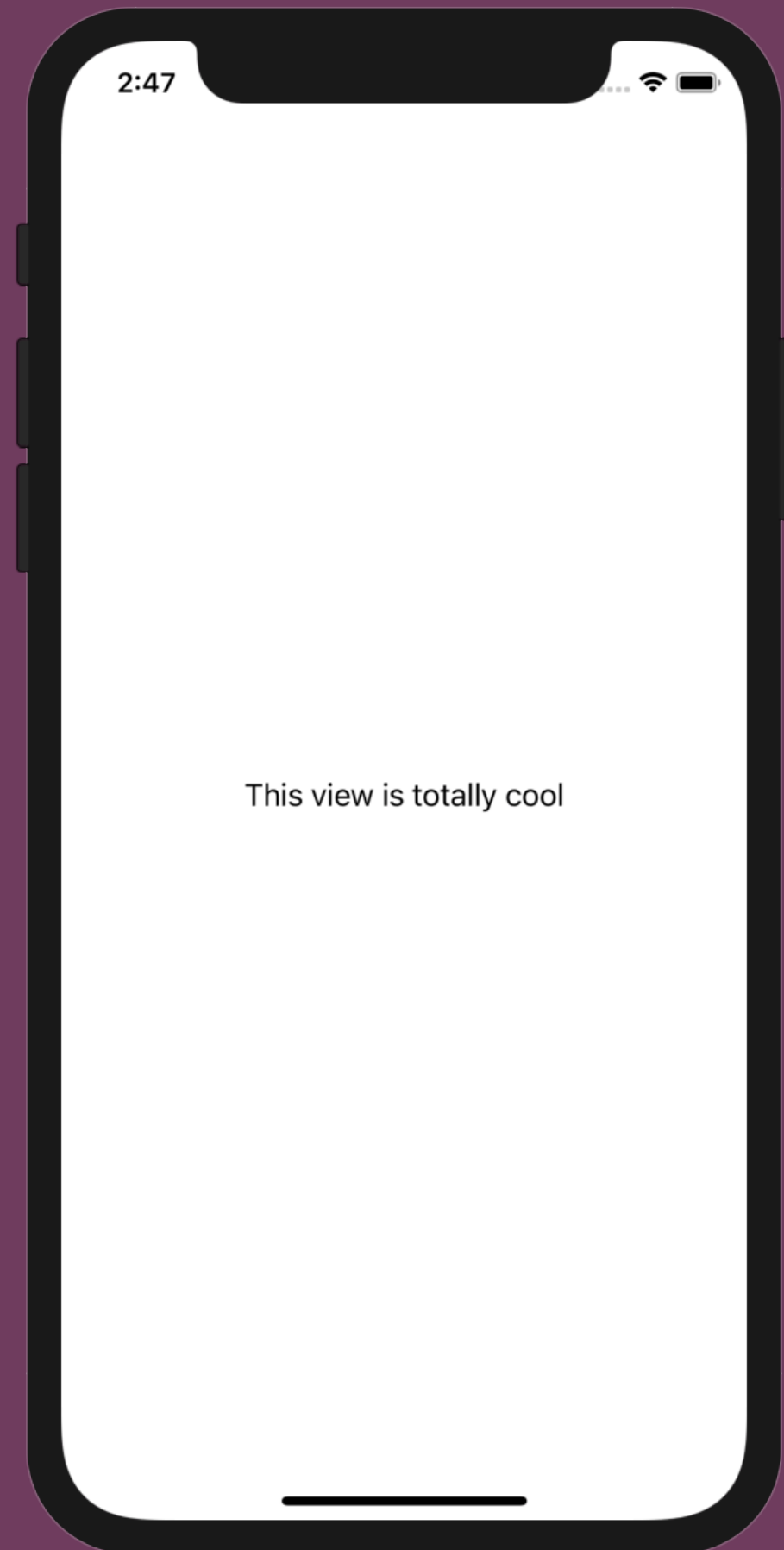
MyLibrary.swift

```
import SwiftUI

public struct MyCoolView: View {

    public init() {}

    public var body: some View {
        Text("This view is totally cool")
    }
}
```



Add images

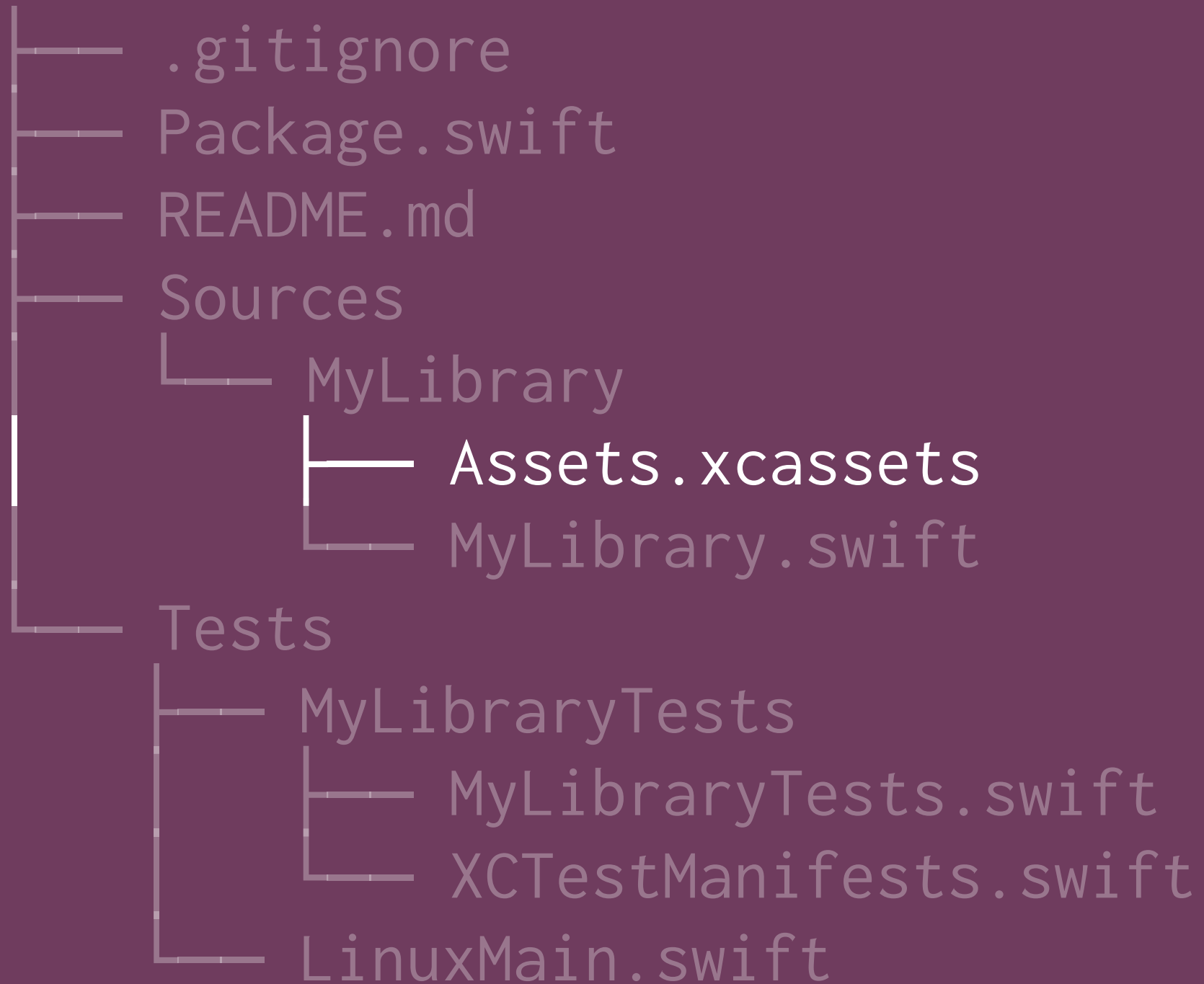
```
import SwiftUI

public struct MyCoolView: View {

    public init() {}

    public var body: some View {
        VStack {
            Image("coolImage", bundle: .module)
            Text("This view is totally cool")
        }
    }
}
```

The Package Structure

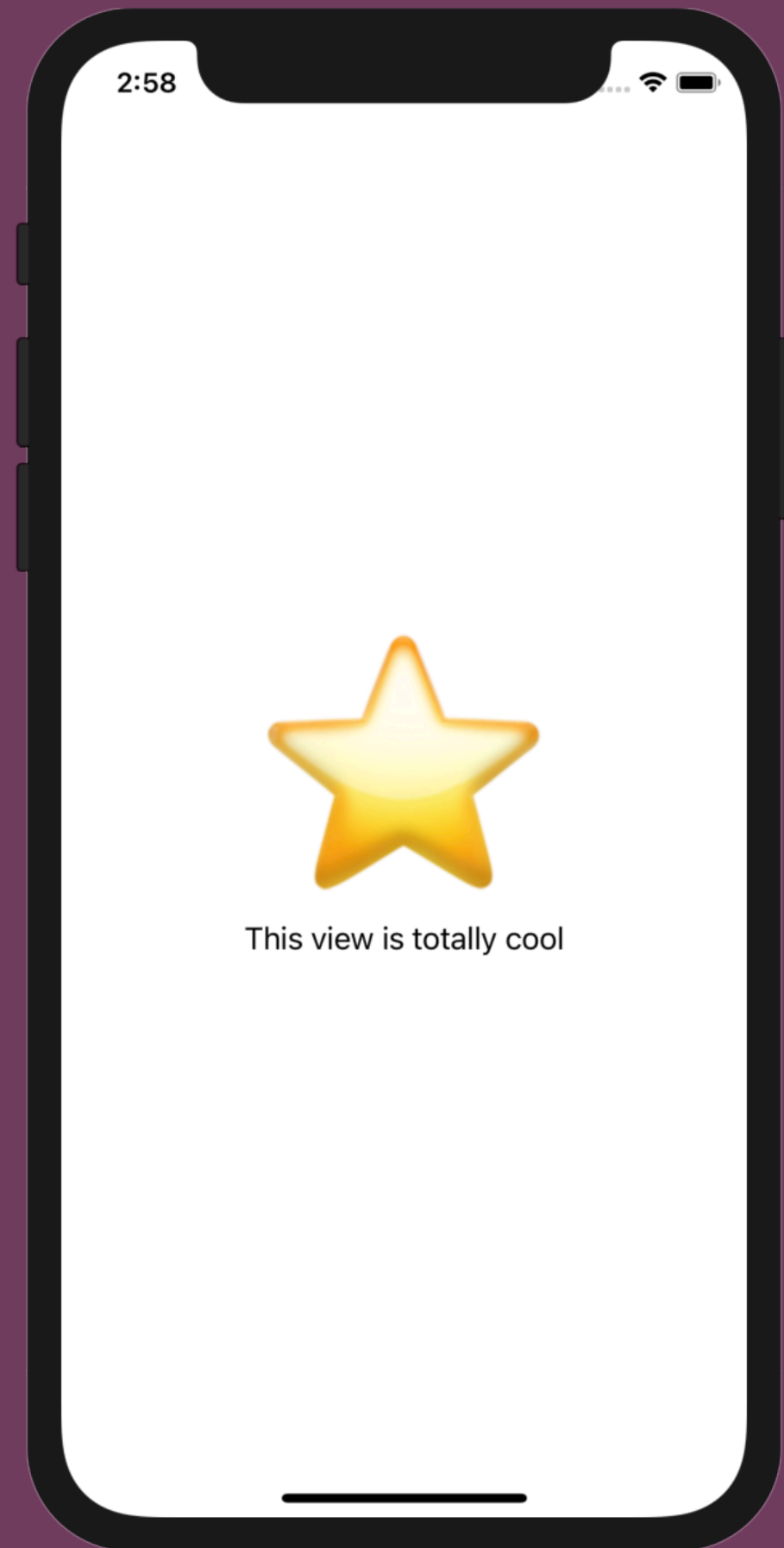


```
import SwiftUI

public struct MyCoolView: View {

    public init() {}

    public var body: some View {
        VStack {
            Image("coolImage", bundle: .module)
            Text("This view is totally cool")
        }
    }
}
```



Add localization

```
import SwiftUI

public struct MyCoolView: View {

    public init() {}

    public var body: some View {
        VStack {
            Image("coolImage", bundle: .module)
            Text("cool_view_text", bundle: .module)
        }
    }
}
```

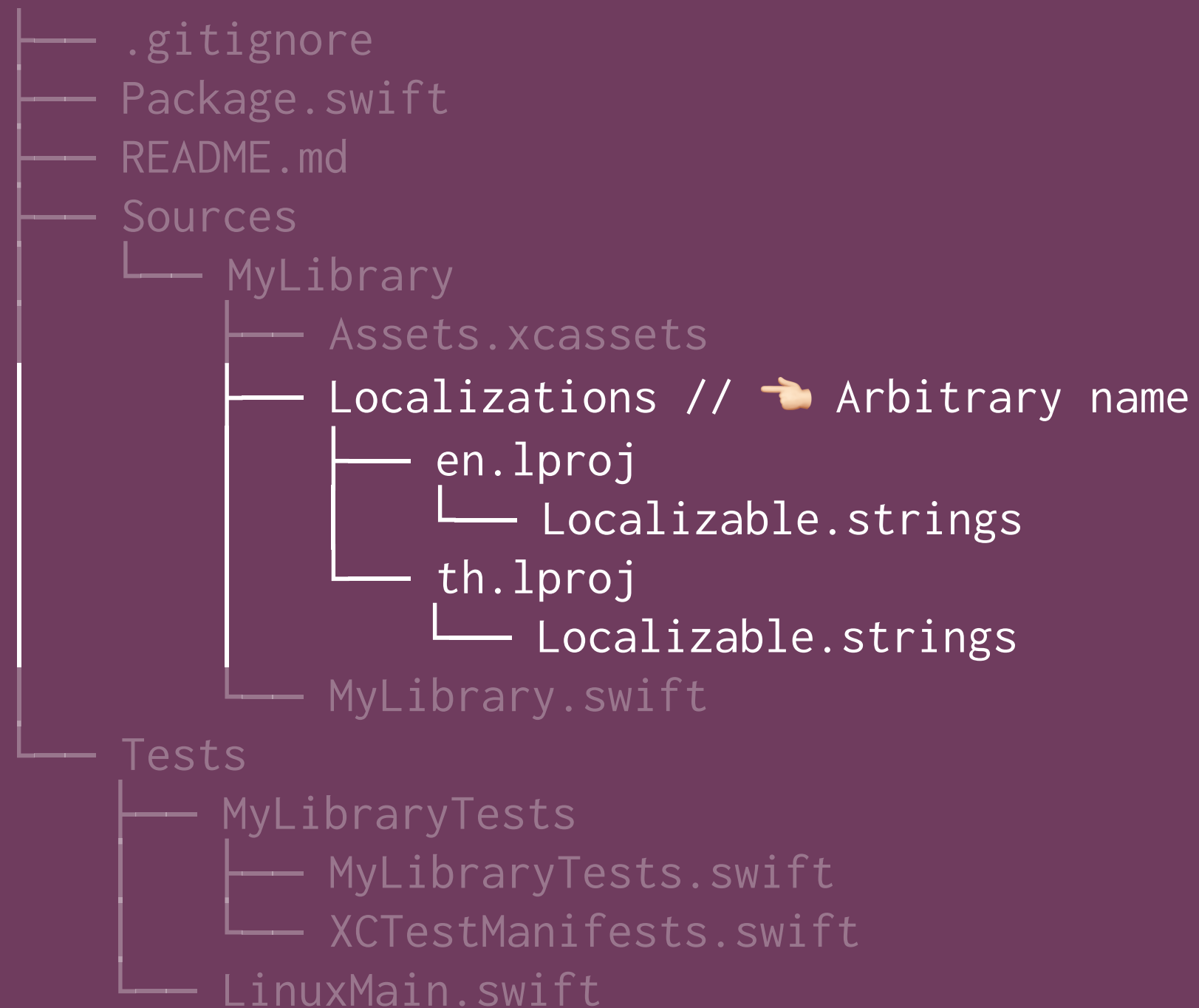
Package.swift

```
// swift-tools-version:5.3

import PackageDescription

let package = Package(
    name: "MyLibrary",
    defaultLocalization: LanguageTag("en"),
    platforms: [.iOS(.v13)],
    products: [
        .library(name: "MyLibrary", targets: ["MyLibrary"]),
    ],
    dependencies: [],
    targets: [
        .target(name: "MyLibrary", dependencies: []),
        .testTarget(name: "MyLibraryTests", dependencies: ["MyLibrary"]),
    ]
)
```

The Package Structure



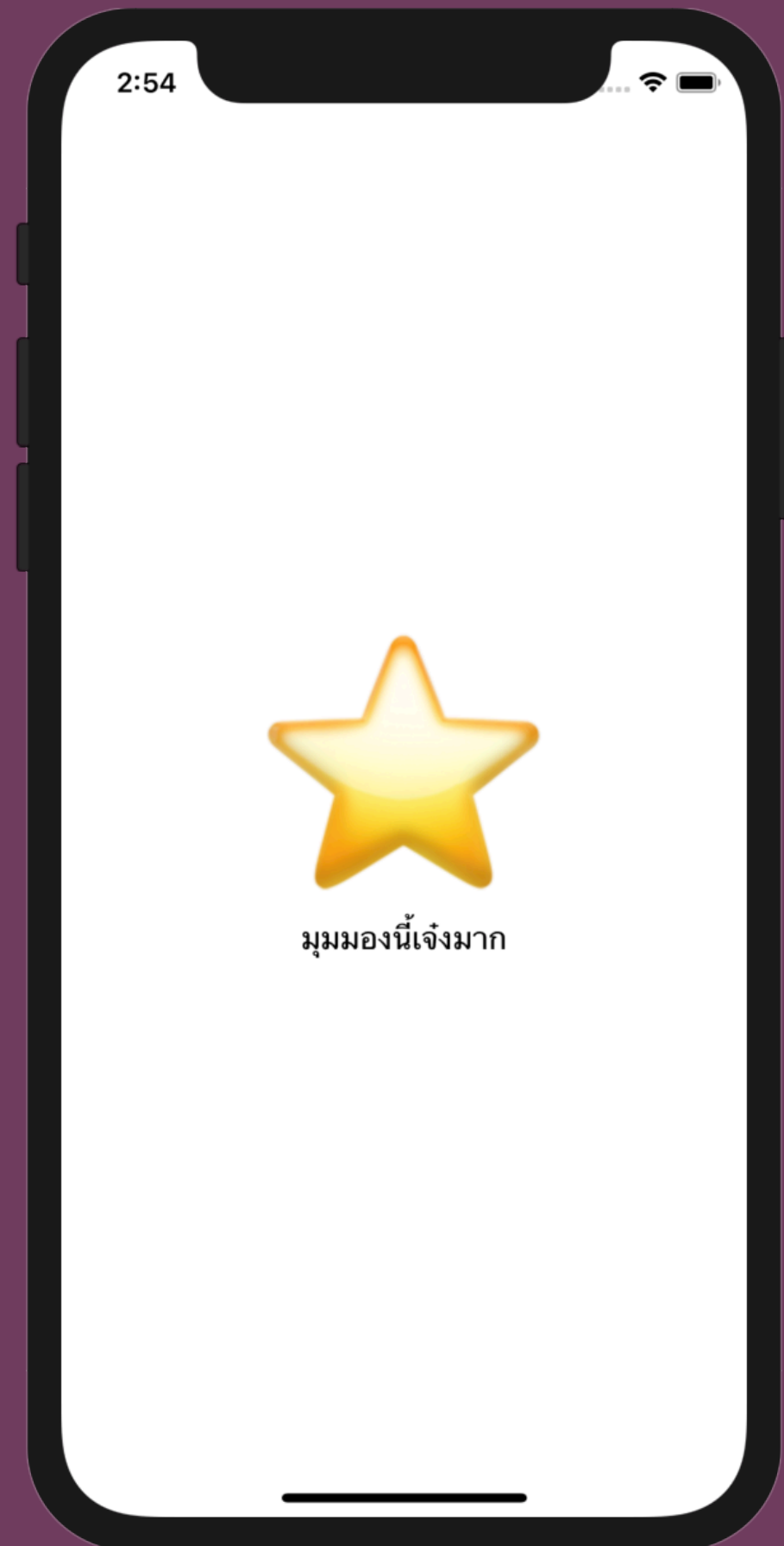
Add localization

```
import SwiftUI

public struct MyCoolView: View {

    public init() {}

    public var body: some View {
        VStack {
            Image("coolImage", bundle: .module)
            Text("cool_view_text", bundle: .module)
        }
    }
}
```



Testing our
package

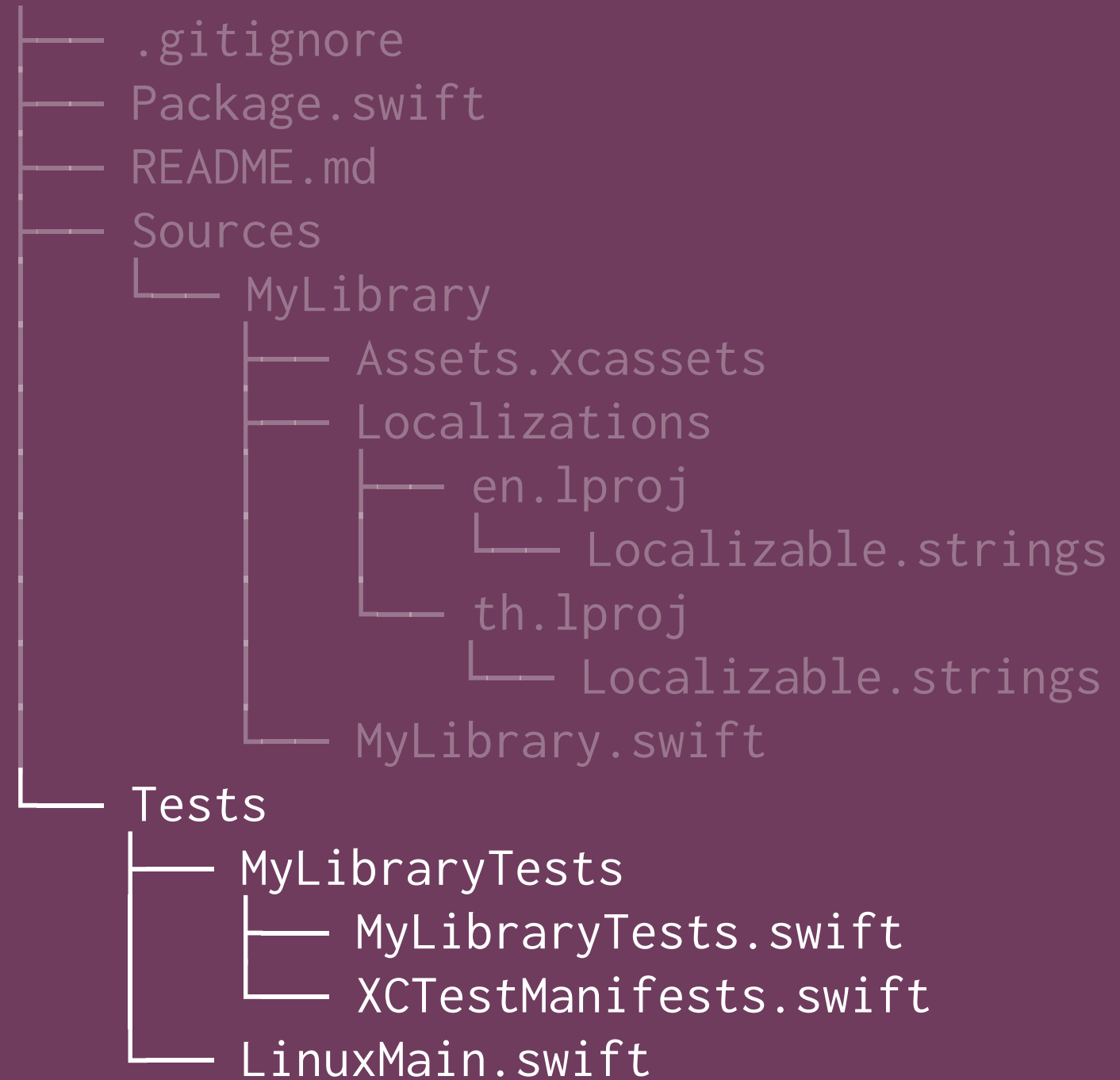
Testing a package

```
// swift-tools-version:5.3
```

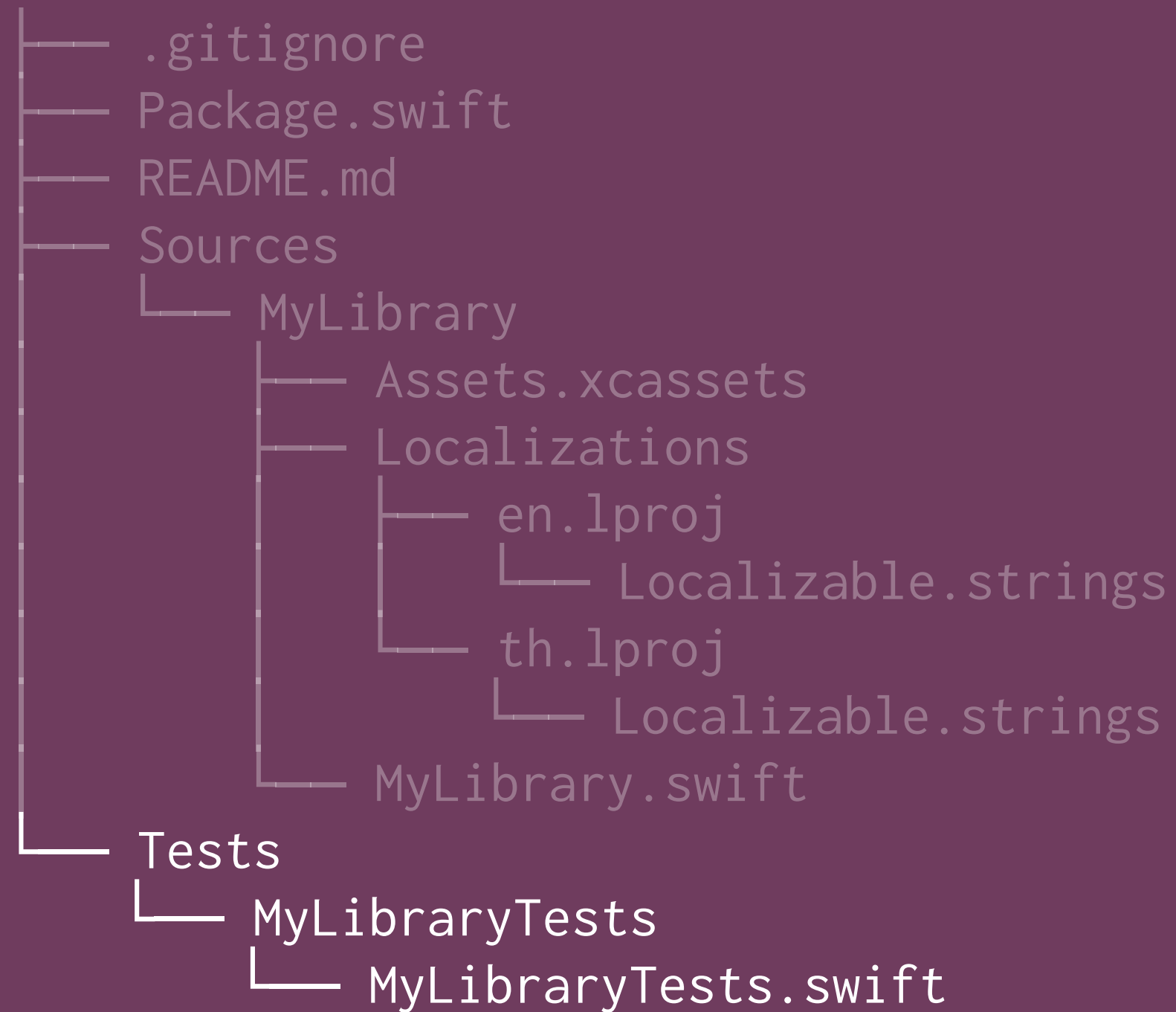
```
import PackageDescription
```

```
let package = Package(  
    name: "MyLibrary",  
    defaultLocalization: LanguageTag("en"),  
    platforms: [.iOS(.v13)],  
    products: [  
        .library(name: "MyLibrary", targets: ["MyLibrary"]),  
    ],  
    dependencies: [],  
    targets: [  
        .target(name: "MyLibrary", dependencies: []),  
        .testTarget(name: "MyLibraryTests", dependencies: ["MyLibrary"]),  
    ],  
)
```

Testing a package



Testing a package



Testing a package

```
import XCTest
@testable import MyLibrary

final class MyLibraryTests: XCTestCase {
    func testExample() {
        XCTAssertEqual(..., "Hello, World!")
    }
}
```

Using our
package

Using our package

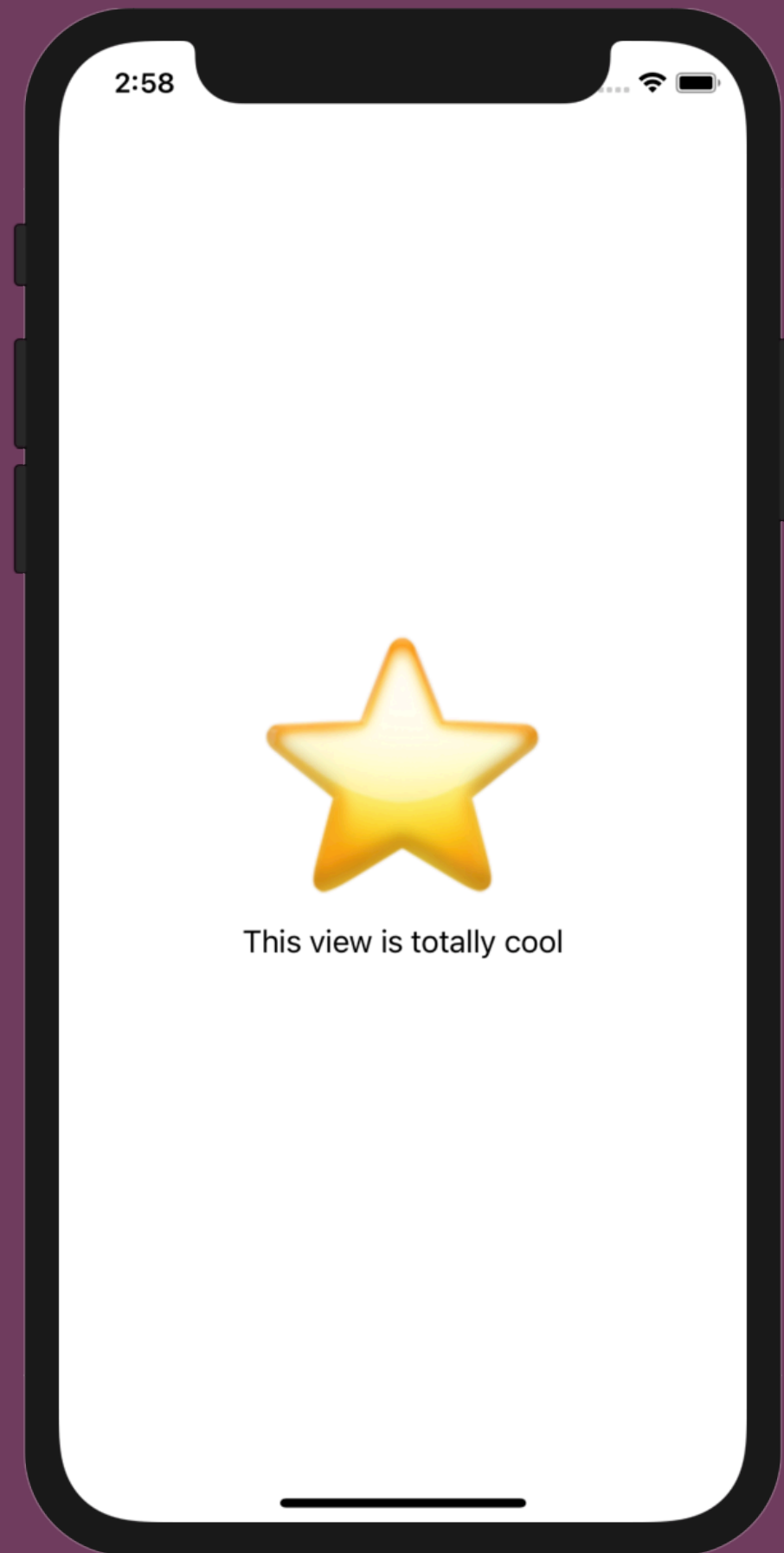
```
import MyLibrary
import SwiftUI

struct ContentView: View {
    var body: some View {
        MyCoolView()
    }
}
```

Done! 🌟😄

```
import MyLibrary
import SwiftUI

struct ContentView: View {
    var body: some View {
        MyCoolView()
    }
}
```

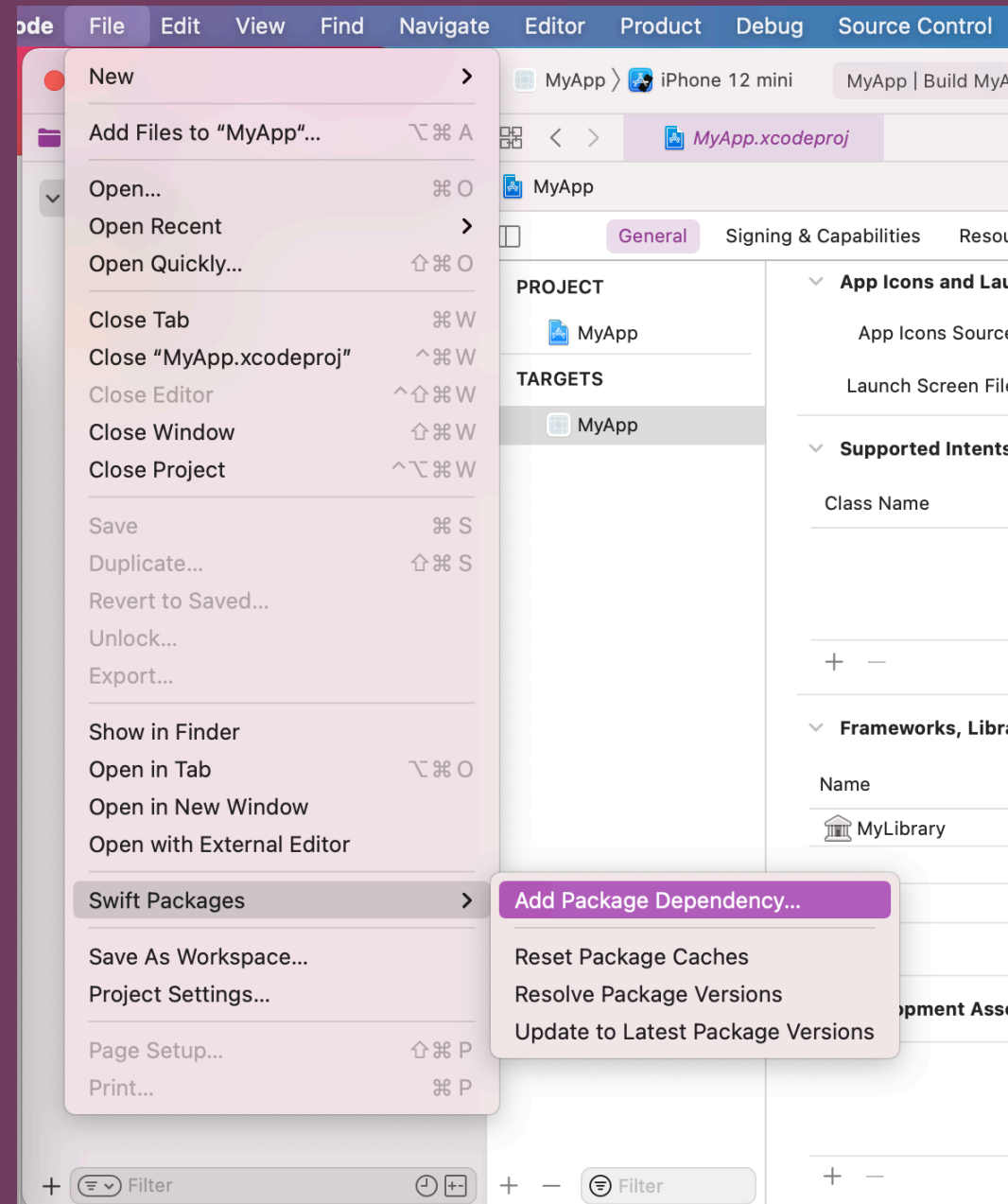


External
packages

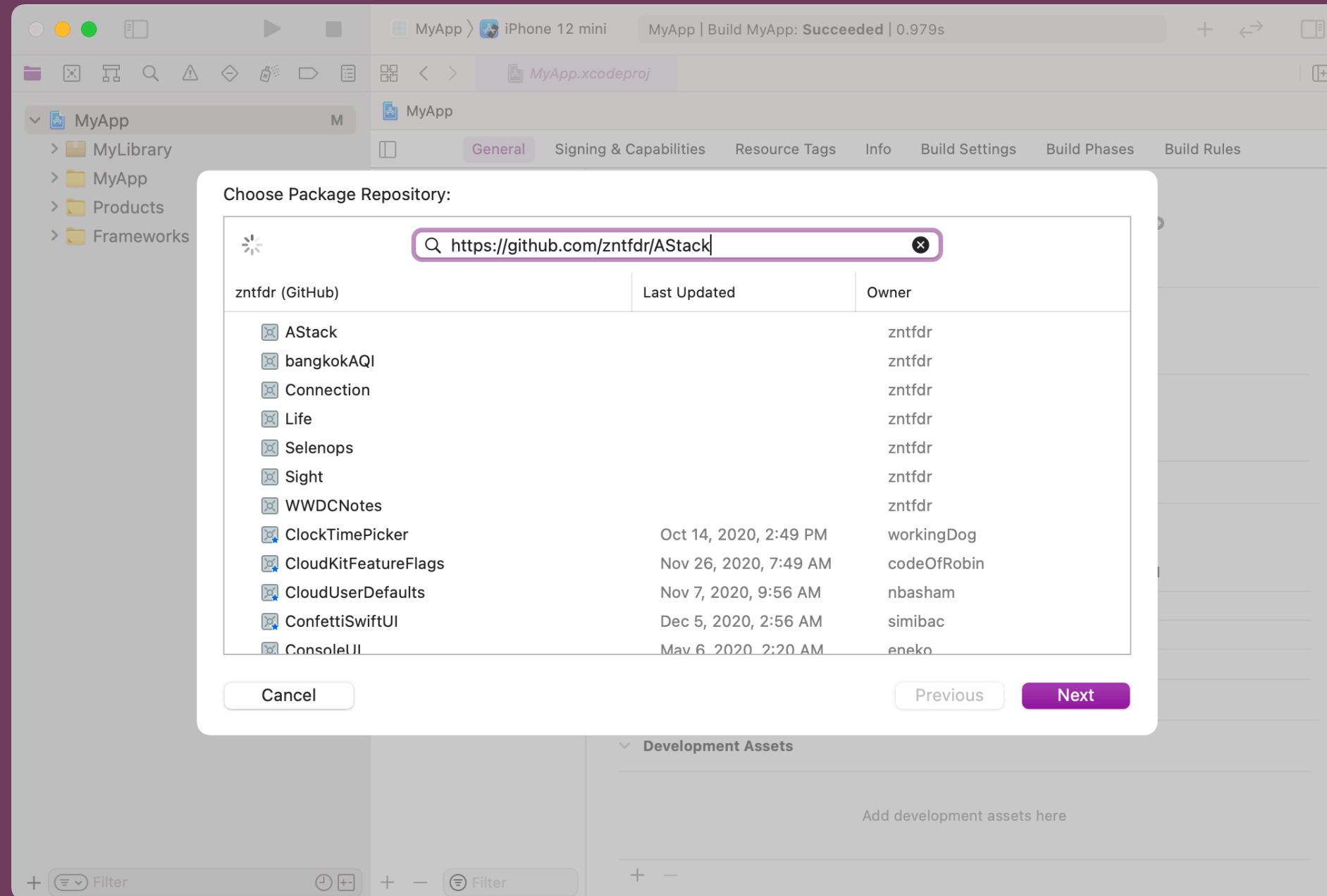
Adding an External Package

1. File > Swift Packages > Add Package Dependency
2. Insert dependency .git URL and follow wizard

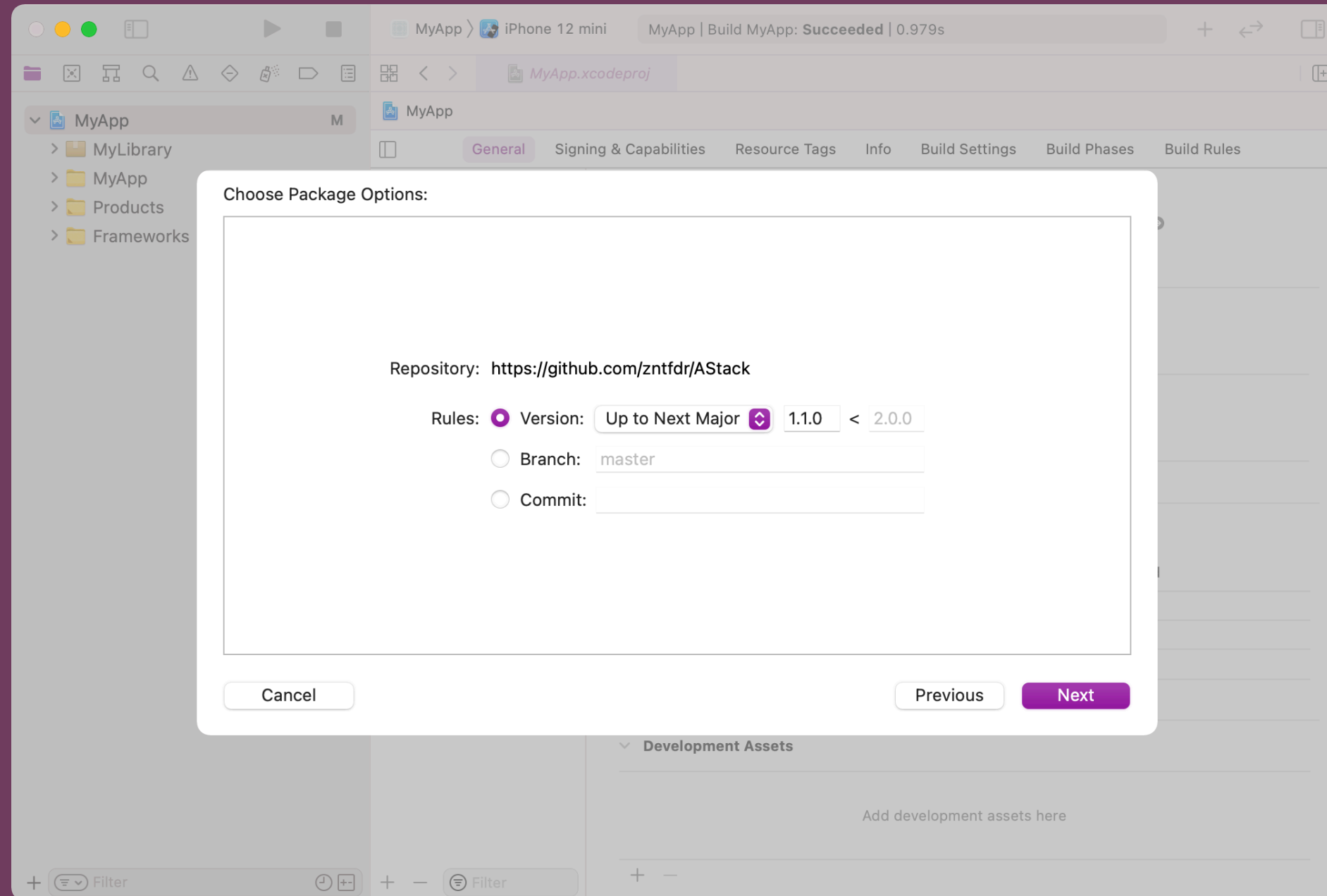
Adding an External Package (1/2)



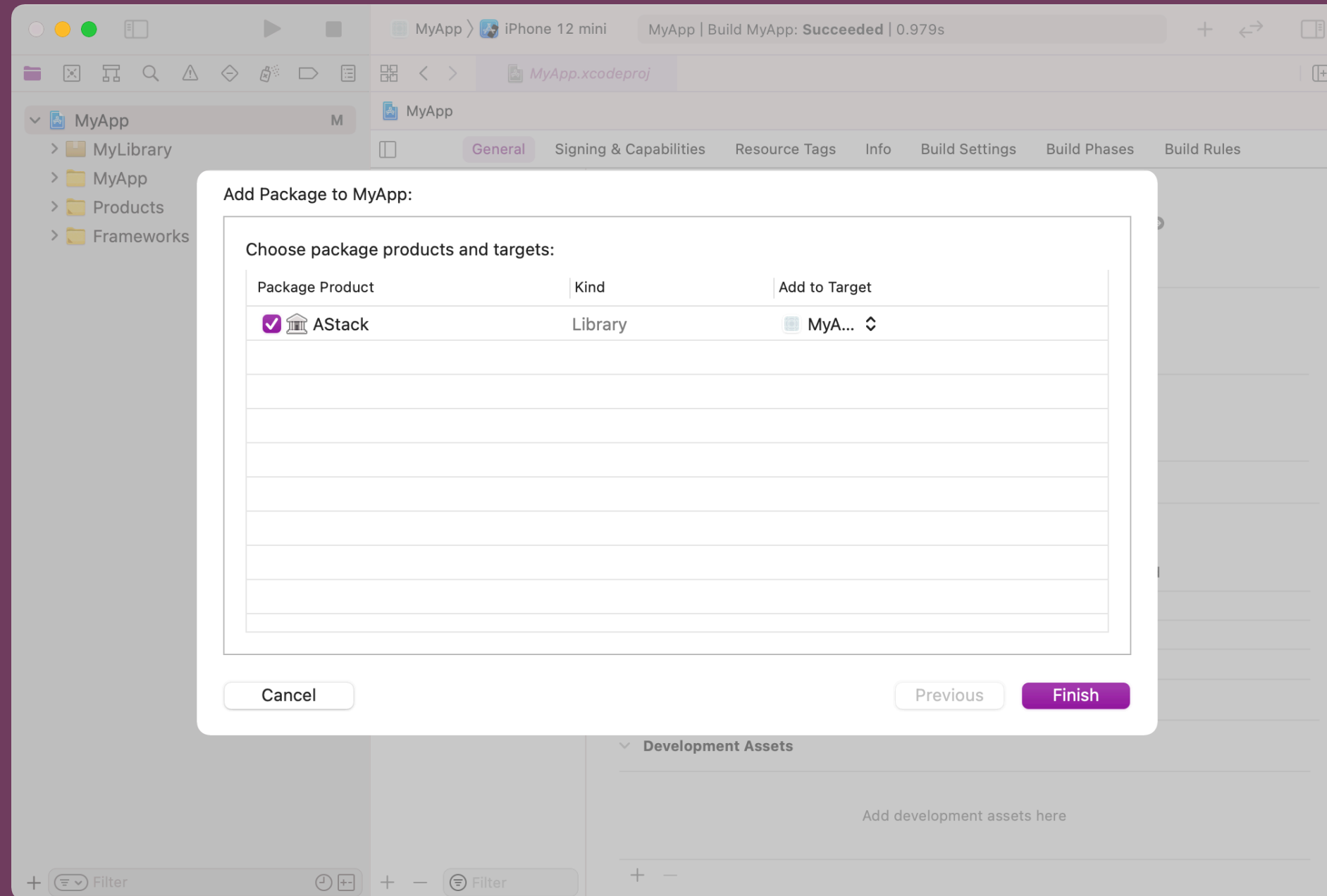
Adding an External Package (2/2)



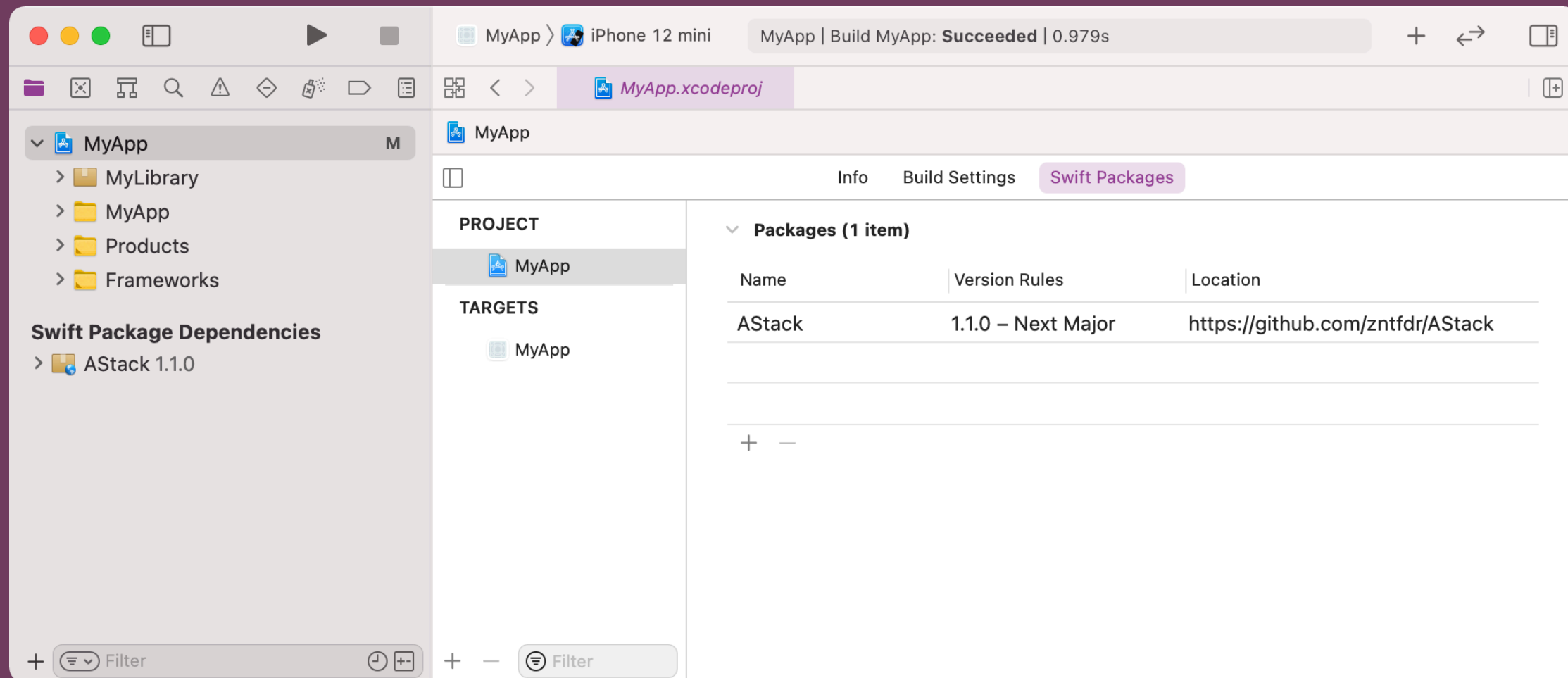
Adding an External Package (2/2)



Adding an External Package (2/2)



Done!



Adding an External Package

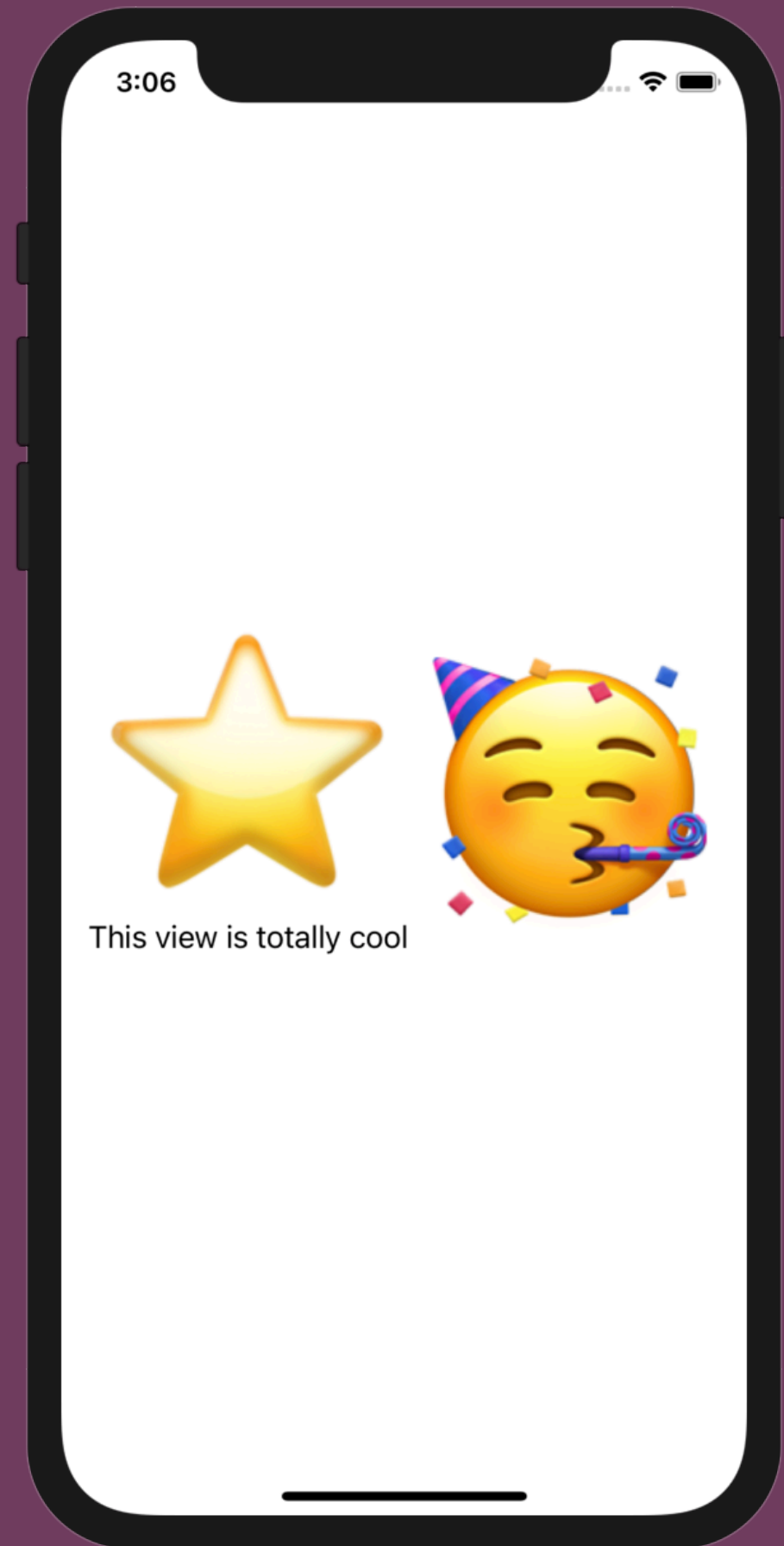
```
import AStack
import MyLibrary
import SwiftUI

struct ContentView: View {
    var body: some View {
        AStack {
            MyCoolView()
            AnotherCoolView()
        }
    }
}
```

Adding an External Package

```
import AStack
import MyLibrary
import SwiftUI

struct ContentView: View {
    var body: some View {
        AStack {
            MyCoolView()
            AnotherCoolView()
        }
    }
}
```



Swift Package all the things!



Federico Zanetello

★★★★★ fivestars.blog • @zntfdr