CoreDataProject) iPhone 11 Pro Max   Running CoreDataProject on iPhone 11 Pro Max

CoreDataProject) CoreDataProject) SceneDelegate.swift) scene(_:willConnectTo:options:)

```
19        // Use this method to optionally configure and attach the UIWindow `window` to the provided UIWindowScene `scene`.
20        // If using a storyboard, the `window` property will automatically be initialized and attached to the scene.
21        // This delegate does not imply the connecting scene or session are new (see
             `application:configurationForConnectingSceneSession` instead).
22
23        // Get the managed object context from the shared persistent container.
24        let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext
25        context.mergePolicy = NSMergeByPropertyStoreTrumpMergePolicy
26        // Create the SwiftUI view and set the context as the value for the managedObjectContext environment keyPath.
27        // Add `@Environment(\.managedObjectContext)` in the views that will need the context.
28        let contentView = ContentView().environment(\.managedObjectContext, context)
29
30        // Use a UIHostingController as window root view controller.
31        if let windowScene = scene as? UIWindowScene {
32            let window = UIWindow(windowScene: windowScene)
33            window.rootViewController = UIHostingController(rootView: contentView)
34            self.window = window
35            window.makeKeyAndVisible()
36        }
37    }
38
39    func sceneDidDisconnect(_ scene: UIScene) {
40        // Called as the scene is being released by the system.
41        // This occurs shortly after the scene enters the background, or when its session is discarded.
42        // Release any resources associated with this scene that can be re-created the next time the scene connects.
43        // The scene may re-connect later, as its session was not neccessarily discarded (see
             `application:didDiscardSceneSessions` instead).
44    }
45
46    func sceneDidBecomeActive(_ scene: UIScene) {
47        // Called when the scene has moved from an inactive state to an active state.
48        // Use this method to restart any tasks that were paused (or not yet started) when the scene was inactive.
49    }
50
51    func sceneWillResignActive(_ scene: UIScene) {
52        // Called when the scene will move from an active state to an inactive state.
53        // This may occur due to temporary interruptions (ex. an incoming phone call).
54    }
55
56    func sceneWillEnterForeground(_ scene: UIScene) {
57        // Called as the scene transitions from the background to the foreground.
58        // Use this method to undo the changes made on entering the background.
59    }
60
61    func sceneDidEnterBackground(_ scene: UIScene) {
62        // Called as the scene transitions from the foreground to the background.
63        // Use this method to save data, release shared resources, and store enough scene-specific state information
64        // to restore the scene back to its current state.
```

CoreDataProject   M
  CoreDataProject
    CoreDataProject
      AppDelegate.swift   A
      SceneDelegate.swift   A
      ContentView.swift   A
      Assets.xcassets   M
      LaunchScreen.storyboard   A
      Info.plist   A
      CoreDataProject.xcdatamodeld   M
    Preview Content   A
  Products

Identity and Type

Name   SceneDelegate.swift
Type   Default - Swift Source
Location   Relative to Group
         SceneDelegate.swift
Full Path   /Users/benkeys/Documents/
           GitHub/100-Days-of-
           SwiftUI/Apps/
           CoreDataProject/
           CoreDataProject/
           SceneDelegate.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

☑ CoreDataProject

Text Settings

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

Filter

CoreDataProject) Thread 1) 7 main

```swift
//
//  ContentView.swift
//  CoreDataProject
//
//  Created by Benjamin Keys on 6/29/20.
//  Copyright © 2020 Ben Keys. All rights reserved.
//

import SwiftUI

struct ContentView: View {
    @Environment(\.managedObjectContext) var moc
    @FetchRequest(entity: Wizard.entity(), sortDescriptors: []) var wizards: FetchedResults<Wizard>

    var body: some View {
        VStack {
            List(wizards, id: \.self) { wizard in
                Text(wizard.name ?? "unknown")
            }

            Button("Add") {
                let wizard = Wizard(context: self.moc)
                wizard.name = "Harry Potter"
            }

            Button("Save") {
                do {
                    try self.moc.save()
                } catch {
                    print(error.localizedDescription)
                }
            }
        }
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```