iOS Design Patterns

Hands-On Challenges

iOS Design Patterns

Joshua Greene

Notice of Rights

Notice of Liability

Trademarks

Table of Contents: Overview

Table of Contents: Extended

7 Memento - Challenge

By Joshua Greene

Persisting the `BasicAuthToken` and `User` into `UserDefaults` works, but is it a good idea? Since `UserDefaults` doesn't use encryption, this isn't a great choice to be storing sensitive authentication and user data...!

The keychain is the right place to store this, yet it's more difficult to use. Writing a "keychain wrapper" class is beyond the scope of this challenge, but fortunately there are already several excellent, open-source options you can use!

Challenge

Using your preferred dependency manager (e.g. CocoaPods, Carthage, etc), install **SwiftKeychainWrapper**. Here's the GitHub URL for this library:

https://github.com/jrendel/SwiftKeychainWrapper

Replace `UserDefaults` with `KeychainWrapper.standard` instead.

Challenge Solution

This example solution uses CocoaPods to install **SwiftKeychainWrapper**. If you're new to CocoaPods and would like to learn how to use it, read our tutorial on it here:

https://www.raywenderlich.com/97014

Alternatively, feel free to use whichever dependency manager you prefer. ;]

Close **RWClean.xcodeproj** if you currently have it open.

Open terminal and `cd` into your working directory for your project.

Run `pod init` to create a default `Podfile`. Open it, and replace its contents with the following:

```
platform :ios, '10.0'
use_frameworks!

target 'RWClean' do
  pod 'SwiftKeychainWrapper', '~> 3.0'
end
```

Still in Terminal, enter enter and run `pod install` like normal to install this dependency.

Open the newly generated **RWClean.xcworkspace** and then open **AuthClient.swift**.

Add the following import to the top of the file:

```
import SwiftKeychainWrapper
```

Replace the **let persistantStore** declaration with the following:

```
fileprivate let persistantStore = KeychainWrapper.standard
```

Fortunately, `KeychainWrapper` has the exact same method signatures used by `UserDefaults`, so you don't need to make any other changes. Nice! ;]

Build and run and verify everything works as expected.

Note that you will need to sign in again, so your credentials will be stored in the keychain now.