

FREE TRIAL: Accelerate your app development career with Hacking with Swift+! >>

< What is the @State property wrapper? What is the @Published property wrapper? >

What is the @StateObject property wrapper?

Paul Hudson @twostraws February 9th 2021

Updated for Xcode 13.2

New in iOS 14

SwiftUI's **@StateObject** property wrapper is designed to fill a very specific gap in state management: when you need to create a reference type inside one of your views and make sure it stays alive for use in that view and others you share it with.

As an example, consider a simple **User** class such as this one:

```
class User: ObservableObject {
    var username = "@twostraws"
}
```

If you want to use that inside various views, you either need to create it externally to SwiftUI and inject it in, or create it inside one of your SwiftUI views and use **@StateObject**, like this:

```
struct ContentView: View {
    @StateObject var user = User()

    var body: some View {
        Text("Username: \(user.username)")
    }
}
```


That will make sure the **User** instance does not get destroyed when the view updates.

Previously you might have used **@ObservedObject** to get the same result, but that was dangerous – sometimes, and only sometimes, **@ObservedObject** could accidentally release the object it was storing, because it wasn't designed to be the ultimate source of truth for the object. This won't happen with **@StateObject**, so you should use it instead.

Important: You should use **@StateObject** only once per object, which should be in whichever view is responsible for creating the object. All other views that share your object should use **@ObservedObject**.

BUY OUR BOOKS





Raycast

SPONSORED Get work done faster with [Raycast](#). The Mac app brings the macOS Spotlight experience to the next level: Search files, create GitHub pull requests, close Jira issues and so much more. Use the [Xcode extension](#) to open recent projects, search iOS documentation and clear derived data. Or, build your own extension with an easy-to-use API.

[Find out more](#)

Sponsor Hacking with Swift and reach the world's largest Swift community!

Similar solutions...

- How to use @StateObject to create and monitor external objects
- All SwiftUI property wrappers explained and compared
- What is the @GestureState property wrapper?
- What is the @Published property wrapper?
- What is the @ObservedObject property wrapper?

< What is the @State property wrapper? What is the @Published property wrapper? >

Was this page useful? Let us know!



Average rating: 4.0/5

Click here to visit the Hacking with Swift store >>



@twostraws



paul@hackingwithswift.com



Sponsor the site

About

Glossary

Privacy Policy

Refund Policy

Update Policy

Code of Conduct

Thanks for your support, Jeff Summers!

Swift, SwiftUI, the Swift logo, Swift Playgrounds, Xcode, Instruments, Cocoa Touch, Touch ID, AirDrop, iBeacon, iPhone, iPad, Safari, App Store, watchOS, tvOS, Mac and macOS are trademarks of Apple Inc., registered in the U.S. and other countries. Pulp Fiction is copyright © 1994 Miramax Films. Hacking with Swift is ©2021 Hudson Heavy Industries.



You are not logged in

Log in or create account