

NEW: My new book Pro SwiftUI is out now – level up your SwiftUI skills today! >>

[< How to fix “Cannot assign to property: 'self' is immutable”](#)

[How to fix “Ambiguous reference to member 'buildBlock\(\)'” >](#)

How to fix “Initializer 'init(_:rowContent:)' requires that 'SomeType' conform to 'Identifiable'”

[Paul Hudson](#) [@twostraws](#) December 1st 2022

Updated for Xcode 14.2

There are three common reasons this error occurs, and all are relatively easy to fix.

The first reason is using a **List** or **ForEach** like this:

```
ForEach(1...10) {
```

SwiftUI supports the half-open range operator, `..<`, but not the closed range operator. As a result, you need to rewrite the above code to this:

```
ForEach(0..<10) {
```

The second reason is using a **List** or **ForEach** on primitive types that don't conform to the **Identifiable** protocol, such as an array of strings or integers. In this situation, you should use **id: \.self** as the second parameter to your **List** or **ForEach**, like this:

```
ForEach(stringArray, id: \.self) {
```

That tells SwiftUI that each value in the array is unique, and so can be used to identify each item in the loop.

The final reason its happens is if you're looping over an array of custom structs that don't conform to **Identifiable**. Here you should add either add a conformance to that protocol (which means adding an **id** property that is unique), or use **id: \.someUniquePropertyName** as the second parameter to your **List** or **ForEach**, which uses that property instead of the ID.

For example:

```
struct User: Identifiable {  
    let id = UUID()  
    NEW: My new book Pro SwiftUI is out now – level up your SwiftUI skills today! >>  
}
```

If you were looping over an array of those **User** objects and wanted to identify them uniquely by their username, you’d use this:

```
ForEach(users, id: \.username) {
```

SPONSORED

From May 15th to 21st, you can join a FREE crash course for mid/senior iOS devs who want to achieve an expert level of technical and practical skills – it’s the fast track to being a complete senior developer!

Click to save your free spot now

[Sponsor Hacking with Swift and reach the world's largest Swift community!](#)

Similar solutions...

- [How to fix “Referencing initializer 'init\(wrappedValue:\)' on 'ObservedObject' requires that 'SomeType' conform to 'ObservableObject'”](#)
- [How to fix “Fatal error: No ObservableObject of type SomeType found”](#)
- [How to fix “Property declares an opaque return type, but has no initializer expression from which to infer an underlying type”](#)
- [All SwiftUI property wrappers explained and compared](#)
- [How to fix a Form Picker or a NavigationLink that isn’t tappable](#)

[< How to fix “Cannot assign to property: 'self' is immutable”](#)

[How to fix “Ambiguous reference to member 'buildBlock\(\)'” >](#)

Was this page useful? Let us know!



Average rating: 4.5/5

NEW: My new book **Pro SwiftUI** is out now – level up your **SwiftUI** skills today! >>



@twostraws



paul@hackingwithswift.com



Sponsor the site

[About](#)

[Glossary](#)

[Privacy Policy](#)

[Refund Policy](#)

[Update Policy](#)

[Code of Conduct](#)

Thanks for your support, Dennis Kunc!

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 ©2022 Hudson Heavy Industries.



You are not logged in

[Log in or create account](#)