Golang: An interface holding a nil value is not nil

Gary Lu Follow
May 21, 2019 · 2 min read ★

Let's start directly with an example (go playground):

```
package main

import "fmt"

func main() {
    var b interface{}
    var p *int = nil
    b = p
    fmt.Printf("b == nil is %t\n", b == nil)
}
```

What is your expectation for the output? It will end up like:

```
a == nil is true
b == nil is false
```

It is somehow counter-intuitive at first glance, but it makes a lot more sense if we explore a little bit about the reflection model in Golang.

Under the hood, an interface in Golang consists of two elements: type and value. When we assign a nil integer pointer to an interface in our example above, the interface becomes `(*int)(nil)` , and it is not nil. An interface equals nil only if both the type and value are nil.

Here is another example of this (go playground), which is a bad pattern to return the error:

```
package main

import "fmt"

func main() {
    err := doSomething()
    if err != nil {
        fmt.Println("ERROR")
    } else {
        fmt.Println("NO ERROR")
    }
}

func doSomething() error {
    var p *MyError = nil
    if false {   // will not come in this block, the value of p will
    return p
}

type MyError struct{
    msg string
}

func (e MyError) Error() string {
    return e.msg
}
```

This code will always print “ERROR”. Actually, `doSomething()` will always return a non-nil error, because `error` is an interface that `MyError` implements and it is not nil when only the value is nil.

To fix this code, we need to update `doSomething()` to return nil error explicitly.

```
func doSomething() error {
    if false {
        return &MyError{"error"}
    }
    return nil
}
```

So, how can we check if the value of an interface is nil? We need to use the functions in `reflect` package. In our first example, we can validate if the value of b is nil with:

```
fmt.Printf("b.value == nil is %t\n", b == nil ||
(reflect.ValueOf(b).Kind() == reflect.Ptr &&
reflect.ValueOf(b).IsNil()))
```

Conclusion

- An interface holding nil value is not nil. An interface equals nil only if both type and value are nil.

- Law of Reflection

Golang