Swift 2 Error using mutating function in Protocol extension "Cannot use mutating member on immutable value: 'self' is immutable

Not sure what's going on here, this seems like it should be pretty straight forward. I have a protocol that mutable var, an extension with a mutating function. Things are crapping out in the *testClass.testFunc*, when I try and use *mtkAnimQueAppend* declared in the extension, I get this error: **"Cannot use mutating member on immutable value: 'self' is immutable.**

```
protocol MTKAnimateValueDelegate {
    var mtkAnimQue:[MTKAnimateValue]? {get set}
}

extension MTKAnimateValueDelegate {
    ///Adds element to que
    mutating func mtkAnimQueAppend(element:MTKAnimateValue) {

        if mtkAnimQue != nil {
          mtkAnimQue?.append(element)
        } else {
          mtkAnimQue = [element]
        }
    }
}

class testClass: MTKAnimateValueDelegate {

  var mtkAnimQue:[MTKAnimateValue]?

  func testFunc() {
    var animValue = MTKAnimateValue(fromValue: 10, toValue: 20, inSeconds: 2)
    animValue.isAnimating = true
    mtkAnimQueAppend(animValue) //ERROR: "Cannot use mutating member on immutable
value: 'self' is immutable
  }

}
```

protocols swift2 ios9 swift-extensions swift-protocols

asked Oct 13 '15 at 9:23

Michael Kennedy
243 3 8

I find this whole issue quite incredible on a number of levels. The very first thing you want to do with Swift, is of course make a mixin that works like this, for view controllers. I mean it's a protocol oriented language. And ios "is view controllers". So what's the first thing you do? For me it's beyond belief that (A) Apple didn't tidy this from the getgo, and (B) the issue is so obscure. There should be 10,000 QA about this everywhere! Ah well. – Fattie Jan 22 '17 at 19:54

1 Answer

The problem is that, in the protocol you mark the function as mutating, which you need to do if you want to use the protocol on a struct. However, the self that is passed to `testFunc` is immutable (it's a reference to a instance of the class) and that is tripping up the compiler. This would make sense if testClass was actually a struct and you could make the function mutating to resolve the issue.

I can see two work arounds:

1. make the protocol class only

   ```
   protocol MTKAnimateValueDelegate: class { ...
   ```

2. Make testClass a struct and mark testFunc as mutating.

Either way, I think this is a bug that needs to be reported to Apple.

answered Oct 13 '15 at 9:40

JeremyP

70.5k 12 102 135

3 Oh my god. I never knew this. Awesome! – devxoul Feb 8 '16 at 15:28

4 I wish a class protocol was the first thing Xcode suggested. Thanks for your help – Aron Feb 26 '16 at 7:11

Thanks for the tidy answer JeremyP. I sent a bounty to say thanks. – Fattie Jan 22 '17 at 19:55

@JoeBlow Thanks – JeremyP Feb 10 '17 at 12:24