

Nguyen Huu Hoang @Nguyen.Huu.Hoang Theo dõi
1351 13 20

Đã đăng vào thg 6 11, 2018 1:42
PM - 4 phút đọc

429 0 1

Phát hiện Memory leaks với Unit Testing

Ở bài trước, tôi đã giới thiệu về [Memory leaks](#) và một số phương pháp để phát hiện. Phần này chúng ta sẽ tìm hiểu về phương pháp cụ thể là Unit testing. Chúng ta đã biết về cách hoạt động của cycles và weak references, chúng ta có thể viết code để kiểm tra các retain cycles. Ý tưởng là sử dụng các weak references thăm dò các cycles. Với một weak reference trả về một object chúng ta có thể kiểm tra đối tượng đó có leak hay không.

Bởi vì một weak reference không giữ một liên kết mạnh với instance mà nó tham chiếu, có thể instance đã deallocated trong khi các weak references vẫn tham chiếu đến nó. Do đó, ARC sẽ tự động set weak reference thành nil khi instance tham chiếu đã deallocated.

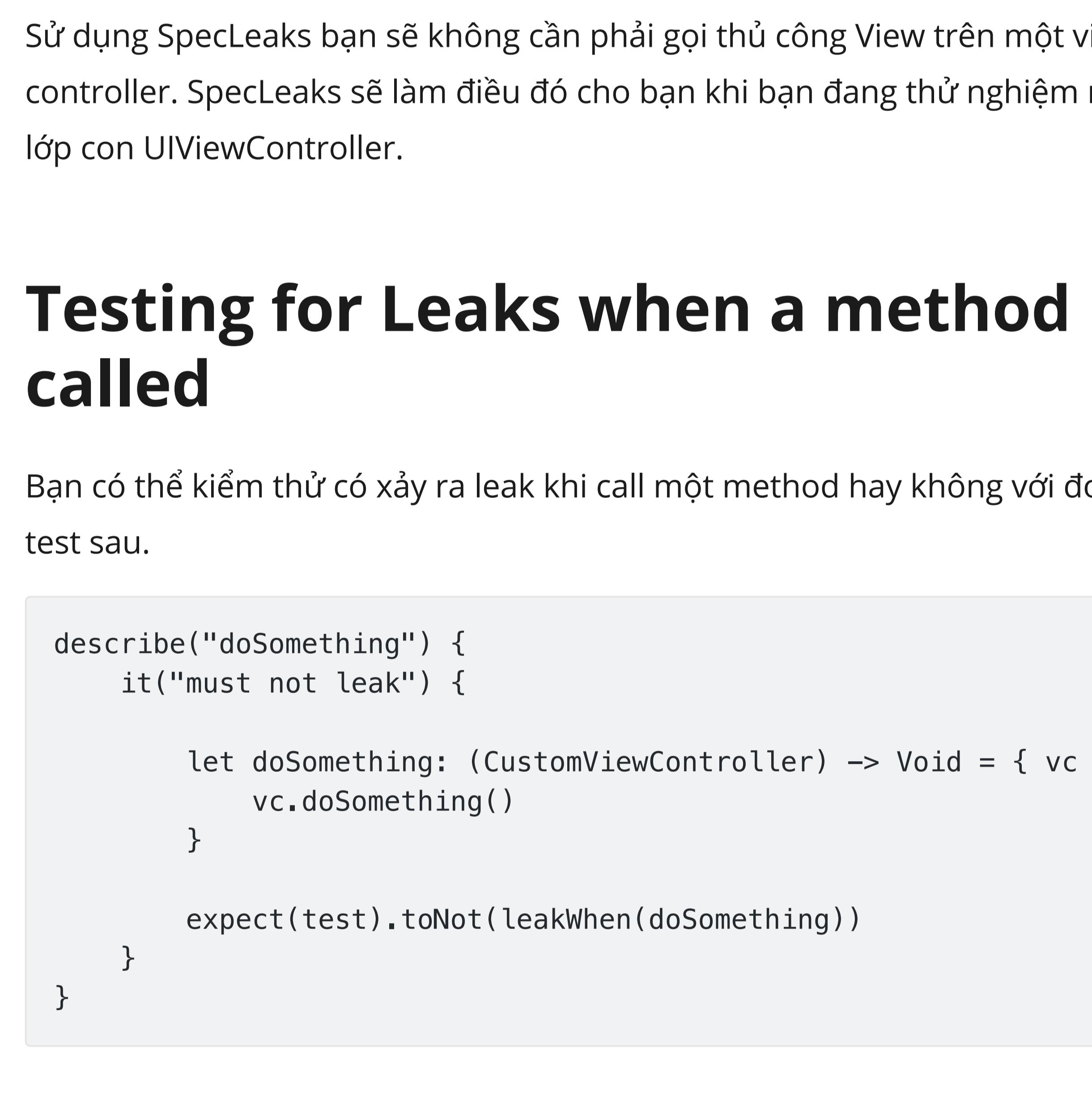
Giả sử chúng ta muốn biết object X có leak hay không. Chúng ta có thể tạo một weak reference trả về nó và gọi là leakReference. Nếu X được giải phóng, ARC sẽ đặt leakReference thành nil. Vì vậy nếu X leak, leakReference sẽ không thể là nil.

```
func isLeaking() -> Bool {
    var x: SomeObject? = SomeObject()
    weak var leakReference = x
    x = nil
    if leakReference == nil {
        return false //Not leaking
    } else {
        return true //Leaking
    }
}
```

Nếu X bị leak, leakReference sẽ trả về nil. Mặt khác, nếu X không bị leak, X sẽ không tồn tại nữa và leakReference sẽ trở thành nil.

SpecLeaks

[SpecLeaks](#) Là một vài additions cho các frameworks cho phép bạn tạo các unit tests để xem các objects bị leak hay không. Nếu bạn không biết về Unit testing có thể xem ảnh dưới hoặc tham khảo bài [Unit Testing](#)



Bạn có thể tạo một tập hợp các tests khởi tạo các đối tượng và thử nghiệm trên chúng. Bạn xác định các kết quả mong đợi, và nếu kết quả như vậy diễn ra, bài test sẽ vượt qua với chấm màu xanh lá. Nếu kết quả không như mong đợi, bài test sẽ thất bại và thể hiện chấm màu đỏ.

Testing for leaks in initialization

Bài test đơn giản nhất bạn có thể viết để kiểm tra object có bị leak hay không. Chỉ cần khởi tạo Object và kiểm tra nó. Đôi khi các object được đăng ký như observer, delegate hoặc notification. Đây là những trường hợp mà đoạn test có thể phát hiện leak.

```
describe("UIViewController") {
    let test = LeakTest {
        return UIViewController()
    }
}
```

```
describe("init") {
    it("must not leak") {
        expect(test).toNot(leak())
    }
}
```

```
}
```

Testing for Leaks in View Controllers

Một view controller có thể bị leak ngay lập tức khi các subview của nó được loaded. Sau đó, rất nhiều điều có thể xảy ra, nhưng với đoạn test đơn giản này bạn chắc chắn viewDidLoad không bị leak.

```
describe("a CustomViewController") {
    let test = LeakTest {
        let bundle = Bundle(for: CustomViewController.self)
        let storyboard = UIStoryboard.init(name: "CustomViewControler", bundle: bundle)
        return storyboard.instantiateInitialViewController() as! CustomViewController
    }
}
```

```
describe("init + viewDidLoad()") {
    it("must not leak") {
        expect(test).toNot(leak())
        //SpecLeaks will detect that a view controller is being created
        // It will create it's view so viewDidLoad() is called
    }
}
```

```
}
```

Sử dụng SpecLeaks bạn sẽ không cần phải gọi thủ công View trên một view controller. SpecLeaks sẽ làm điều đó cho bạn khi bạn đang thử nghiệm một lớp con UIViewController.

Testing for Leaks when a method is called

Bạn có thể kiểm thử có xảy ra leak khi call một method hay không với đoạn test sau.

```
describe("doSomething") {
    it("must not leak") {
        let doSomething: (CustomViewController) -> Void = { vc in
            vc.doSomething()
        }
        expect(test).toNot(leakWhen(doSomething))
    }
}
```

Bài viết khác từ Nguyen Huu Hoang

Hướng dẫn sử dụng Animation trong SwiftUI

Nguyen Huu Hoang
3 phút đọc
110 0 20

Memory Leaks in Swift

Nguyen Huu Hoang
9 phút đọc
2135 4 0

Reference Cycle for Closures

Phan Huynh Thien An
3 phút đọc
450 4 1

Xcode UI Testing

Phan Huynh Thien An
5 phút đọc
223 1 0

Swift 5: Sử dụng Async/Await với Result và GCD

Nguyen Huu Hoang
7 phút đọc
1322 4 2

Cleaner Architecture on iOS

Nguyen Huu Hoang
15 phút đọc
180 1 0

T

o

o

Trạng thái hệ thống

Tiếng Việt

v

Về chúng tôi Phản hồi Giúp đỡ FAQs RSS Điều khoản

DMCA PROTECTED © Viblo 2021

Đã đăng ký Bản quyền

f t o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o