Initialising a view controller from a test using Swift

Testing is important, so we really need to get this right

[](https://medium.com/@stevenpcurtis.sc?source=post_page-----7ce28a712724----------------------)

[Steven Curtis](https://medium.com/@stevenpcurtis.sc?source=post_page-----7ce28a712724----------------------)

[Dec 27, 2019](https://medium.com/@stevenpcurtis.sc/initialising-a-view-controller-from-a-test-7ce28a712724?source=post_page-----7ce28a712724----------------------) · 3 min read

Testing is extremely important in software, but there are few guides on how you can manipulate a view controller to help you out. This might be just the right guide for you!





<img class="s t u hq ai" src="https://miro.medium.com/max/5898/1\*twjdsrFnU3x6fOVLb1ht2g.png" width="2949" height="1134" role="presentation"/>

Photo by [tanialee gonzalez](https://unsplash.com/@catchstars?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) on [Unsplash](https://unsplash.com/s/photos/create?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

Difficulty: Beginner | Easy| **Normal** | Challenging

**Prerequisites:**

* Be able to produce a “Hello, World!” iOS application (guide [HERE](https://medium.com/@stevenpcurtis.sc/your-first-swift-application-without-a-mac-79598ad839f8))
* Have some understanding of basic testing in Swift

**Terminology**

storyboard: A way to graphically lay out the UI in Xcode

unit testing: Software testing where individual units or components of the software is tests

view controller: A view controller is an intermediary between the views it manages and the data of your app

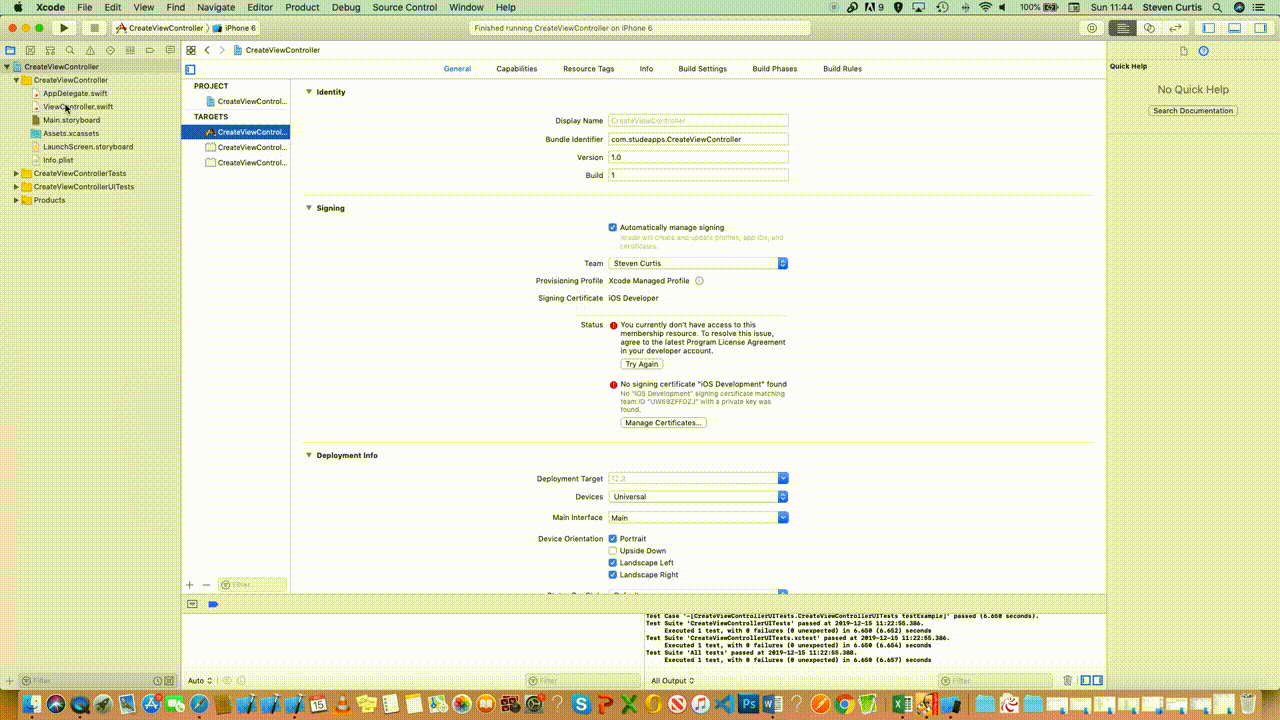
**The Setup**

**Creating a basic view controller**

The setup here is simple: create a basic “Hello, World!” style application with a single UILabel that is centred on the view controller.

Now we need to give the view controller an identifier, that is set the Storyboard ID to ViewController (which is set in the Storyboard).





<img class="s t u hq ai" src="https://miro.medium.com/max/2560/1\*1ahOjjBRajDBH7fdjjGPPw.gif" width="1280" height="720" role="presentation"/>

**Setting up our view controller**

Our code is going to display on the label whether we created the view controller programatically or through the storyboard (more on this later).

We will make the label reflect the source with the following snazzy viewDidLoadfunction

var labelText = "View Controller created from: "

var source: String?

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

centreLabel.text = labelText + (source ?? "")

}

In order to do this we need to have two different initiliasers. So when we create the view with the storyboard we use the following to set the source text:

**required** **init**?(coder aDecoder: NSCoder) {

**super**.init(coder: aDecoder)

**self**.source = "storyboard"

}

and the following to do the same when the view controller is programatically instantiated

**init**(source : String) {

**super**.init(nibName: nil, bundle: nil)

**self**.source = "storyboard"

)

**Creating a test file**

Now the creation of a Unit Test File is is a pretty simple New>File>Unit Test Case Class, which in this case we call ViewControllerTests (I do apologise for these names, by the way).

**The Methods of View Controller Unit Testing**

**Creating the View Controller Programatically**

Here we create a test that instantiates the view controller and we inject the value for the source. As expected the source changes (this should be no surprise!)

func testViewController(){

let viewController = ViewController(source: "tests")

XCTAssertEqual(viewController.source, "tests")

}

The issue is that we are basically testing setting a property, and testing to see that we have changed the property.

*It’s kind of disatisfying.*

**Developing Storyboard**

We can actually instantiate the storyboard, which gives us a view controller which in turn creates the views and eventually the label.

**func** testViewControllerNib(){

**let** vc = (UIStoryboard(name: "Main", bundle: Bundle.main).instantiateViewController(withIdentifier: "ViewController") **as**! ViewController)

UIApplication.shared.keyWindow?.rootViewController = vc

XCTAssertEqual( vc.centreLabel.text, "View Controller created from: storyboard")

}

This means that we have created the view controller vc here, and we enter the usual instantiation path for a view controller — leading to the UILabel here displaying *“View Controller created from: storyboard”*.

**The Debate**

Usually when posting an article about testing someone will claim that “*You shouldn’t use unit testing for UIElements*”, which is the old application of **shouldn’t** to a situation where it *may* make sense for your particular needs (which might even be a manager insisting you work in a certain way).

So you should be aware that people might declare that they know what you **should** and **shouldn’t** do, but in fact you are the one who decides, and this article will hopefully help you out.

**Conclusion…**

The above article should help you out in testing view controllers both programatically and through testing using the storyboard. Perhaps this gives you enough techniques to get beyond the mythical 80% code testing, but in any case

*Happy coding!*

**The Repo? Just here:**

<https://github.com/stevencurtis/createviewcontrollertests>

**Want to get in contact? Try the link here:**

<https://twitter.com/stevenpcurtis>

41