

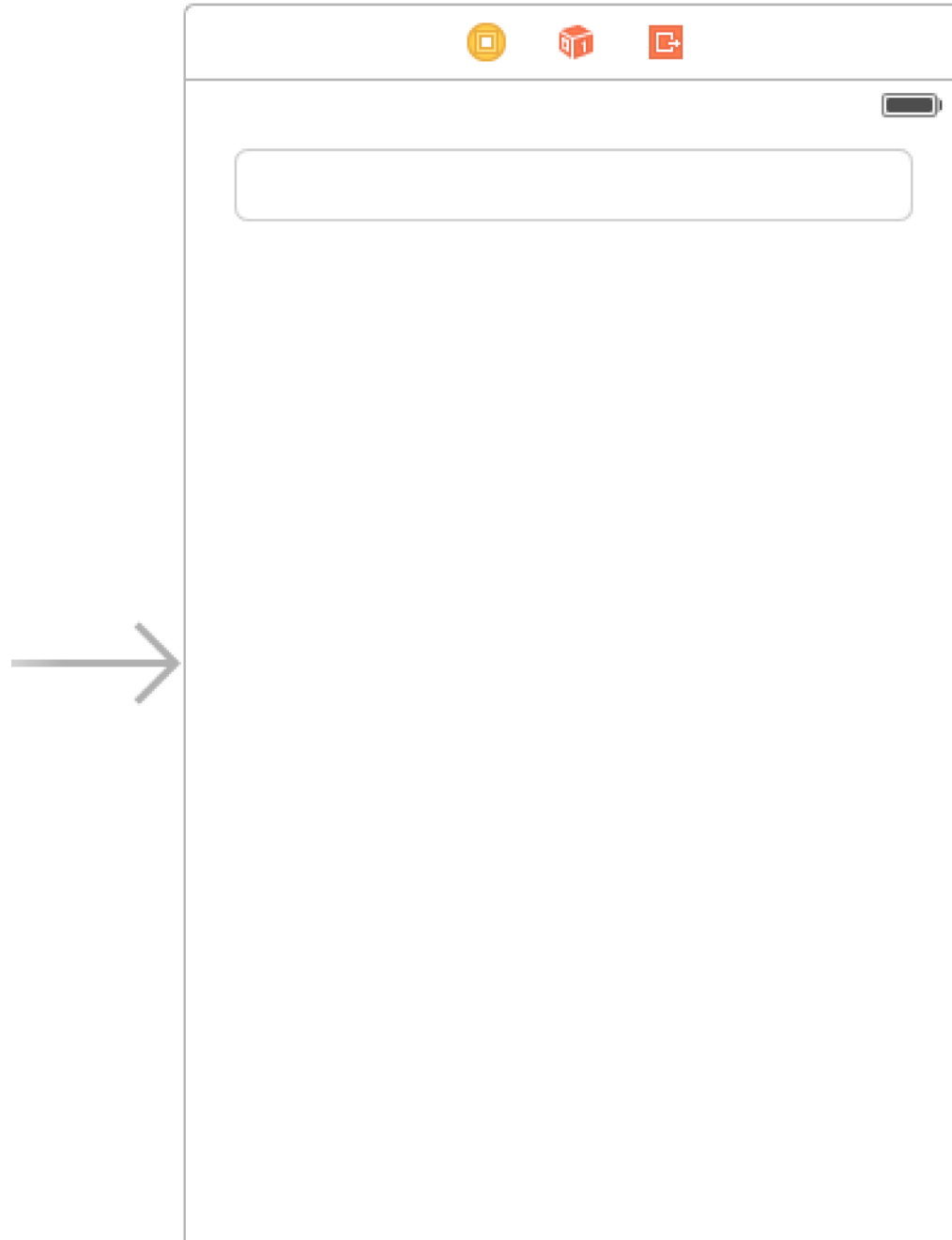
# Hiding Keyboard

It's not gonna hide itself

Create New Project

Add one Text Field at the top.

Let “topTextField” be the name of its outlet.



Try running the project.

You will see that the keyboard does not want to go  
away

# Method 1

# ViewController.swift

Add to viewDidLoad

```
self.topTextField.addTarget(  
    nil, action:"dummy:", forControlEvents:.EditingDidEndOnExit)
```

This will let you dismiss it when you hit return.  
This trick won't work on Text View.

## Method 2

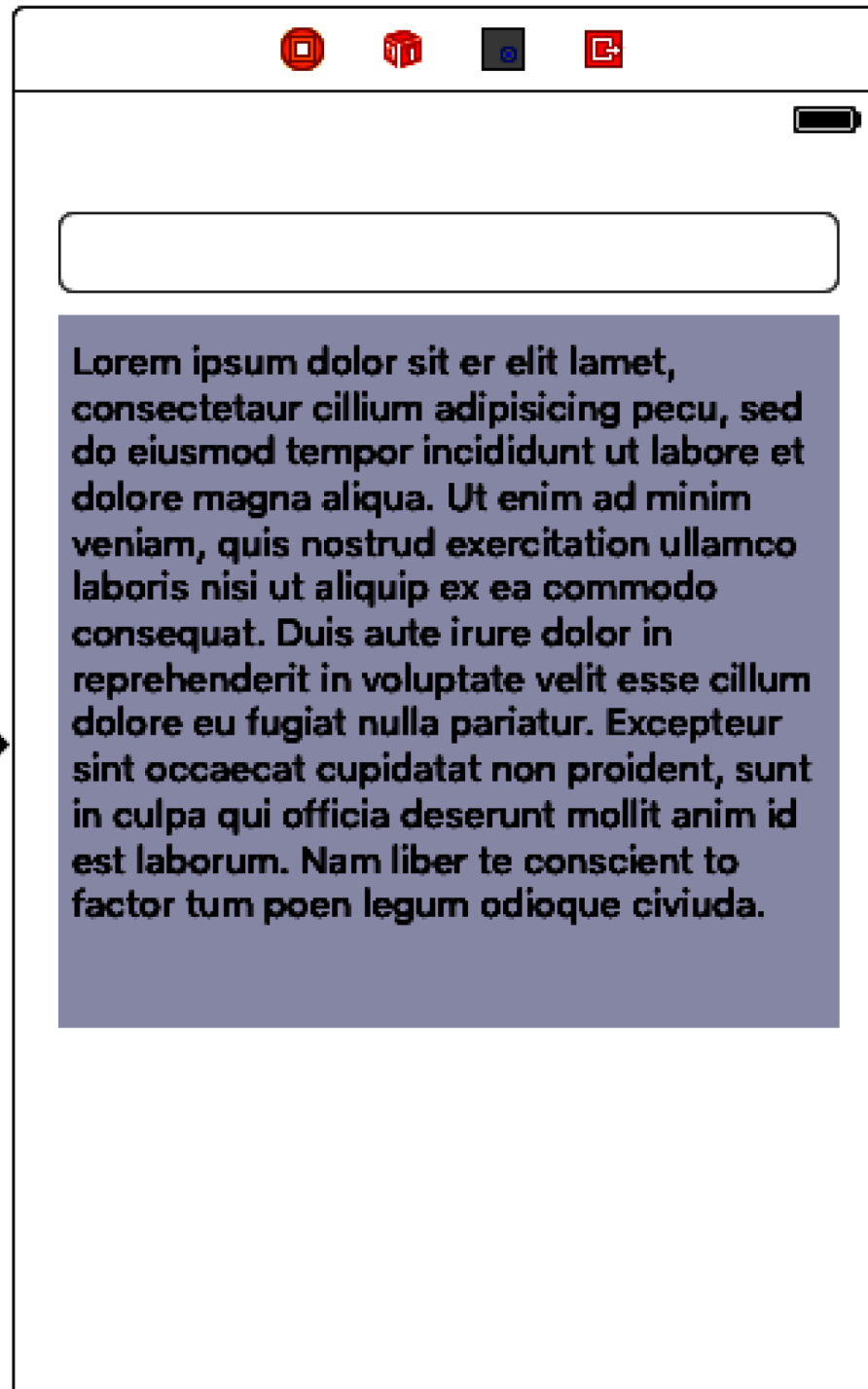


`resignFirstResponder()`

To use this you have to have a place to call it.

1. Create a new page with Text Field and Text Area
2. Create their outlet appropriately

(Method2ViewController is a view controller for the new page. textField and textArea are the outlet names)



# Method2ViewController.swift

- Make view controller `UITextFieldDelegate`
- Add a delegate method

```
func textFieldShouldReturn(textField: UITextField) -> Bool {  
    self.textField.resignFirstResponder()  
    return true  
}
```

# Link Text Field delegate

The screenshot shows the Xcode IDE with a project named 'KeyboardHiding on iPhone 5'. The interface is divided into three main sections: a top toolbar, a central canvas, and a right-hand inspector panel.

**Top Toolbar:** Contains icons for a document, a magnifying glass, a double-headed arrow, and a right-pointing arrow.

**Central Canvas:** Displays a 'View Controller' with a blue background. At the top, there are three icons: a blue circle with a white arrow, a red cube, and a red square. Below these is a white text field. The text field contains the following Lorem Ipsum text: 'Lorem ipsum dolor sit er elit lamet, consectetur cillum adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt'.

**Right-Hand Inspector Panel:** Contains several sections:

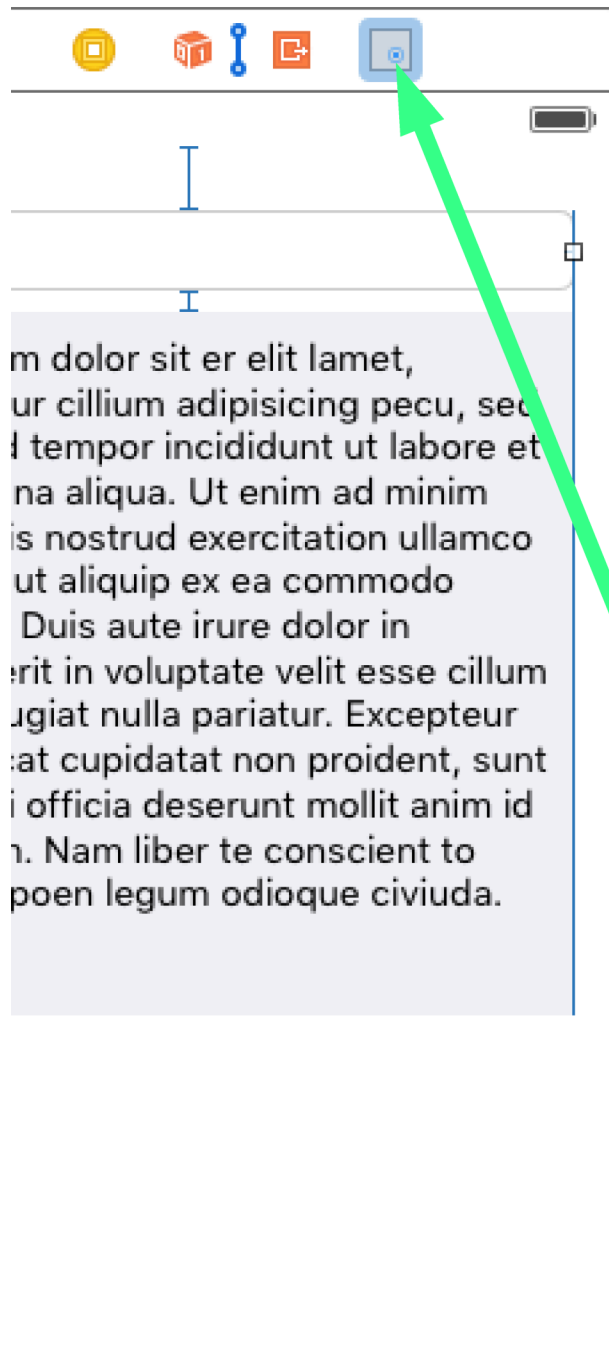
- Triggered Segues:** A section with a single item 'action' and a radio button.
- Outlets:** A section with a single item 'delegate' and a radio button. A blue line connects this radio button to the blue circle icon in the View Controller.
- Outlet Collections:** A section with a single item 'gestureRecognizers' and a radio button.
- Sent Events:** A section with a list of events, each with a radio button:
  - Did End On Exit
  - Editing Changed
  - Editing Did Begin
  - Editing Did End
  - Primary Action Triggered
  - Touch Cancel
  - Touch Down
  - Touch Down Repeat
  - Touch Drag Enter
  - Touch Drag Exit
  - Touch Drag Inside
  - Touch Drag Outside
  - Touch Up Inside

Try running project. When you press return while editing the text field, the keyboard will hide

What if we want to hide keyboard when touching other things?

# Method2ViewController

- Add Tap Gesture Recognizer to the view



**Triggered Segues**  
action


**Outlets**  
delegate


**Outlet Collections**  
gestureRecognizers

**Sent Events**  
Did End On Exit  
Editing Changed  
Editing Did Begin  
Editing Did End  
Primary Action Triggered  
Touch Cancel  
Touch Down  
Touch Down Repeat  
Touch Drag Enter  
Touch Drag Exit  
Touch Drag Inside  
Touch Drag Outside  
Touch Up Inside  
Touch Up Outside  
Value Changed

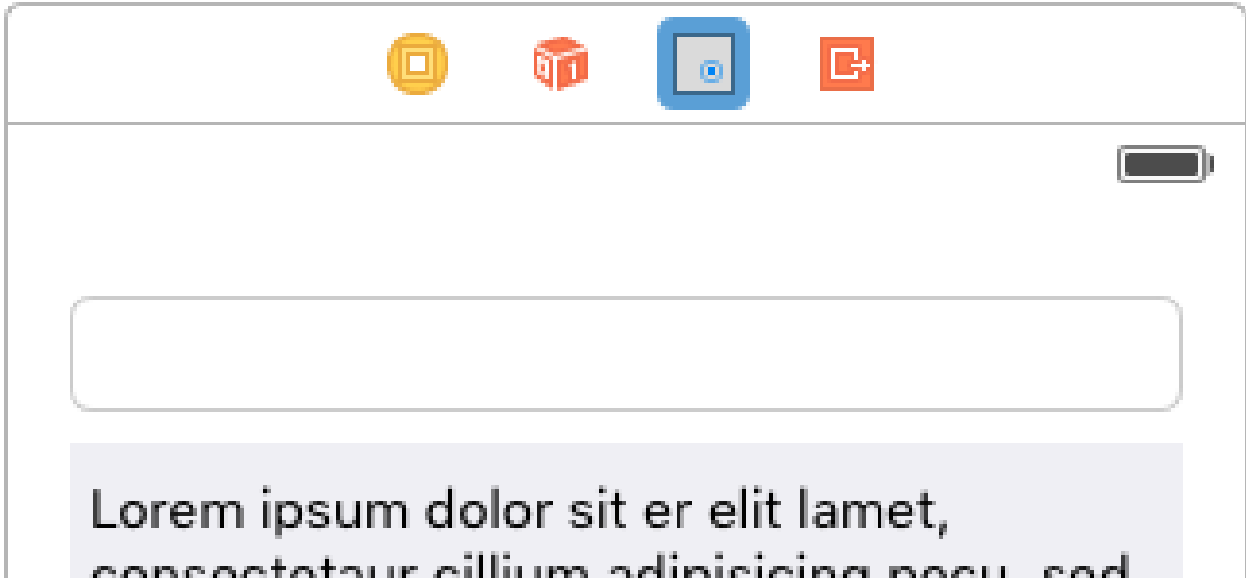
**Referencing Outlets**  
topTextField — \* View Controller  
New Referencing Outlet

**Referencing Outlet Collections**

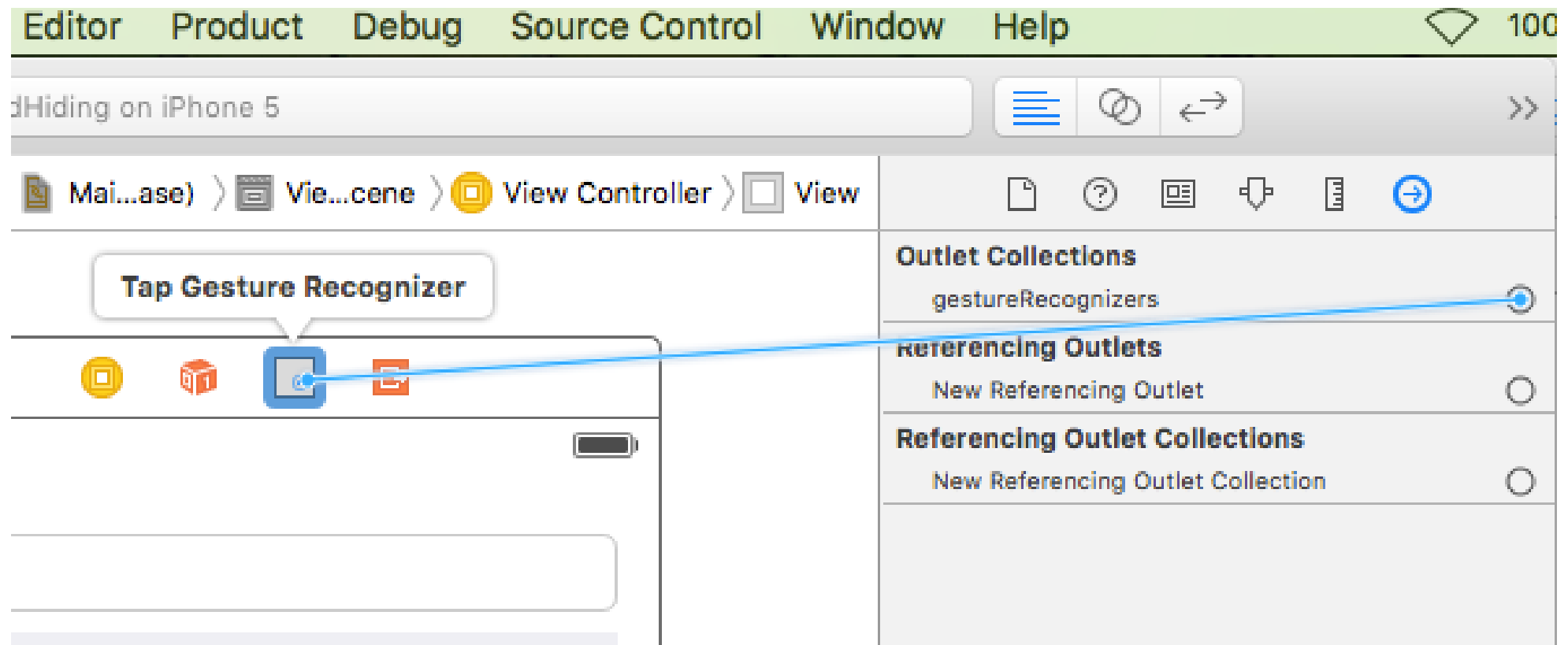
 **Web View** - Displays embedded web content and enables content navigation.

 **Tap Gesture Recognizer** - Provides a recognizer for tap gestures which land on the view.

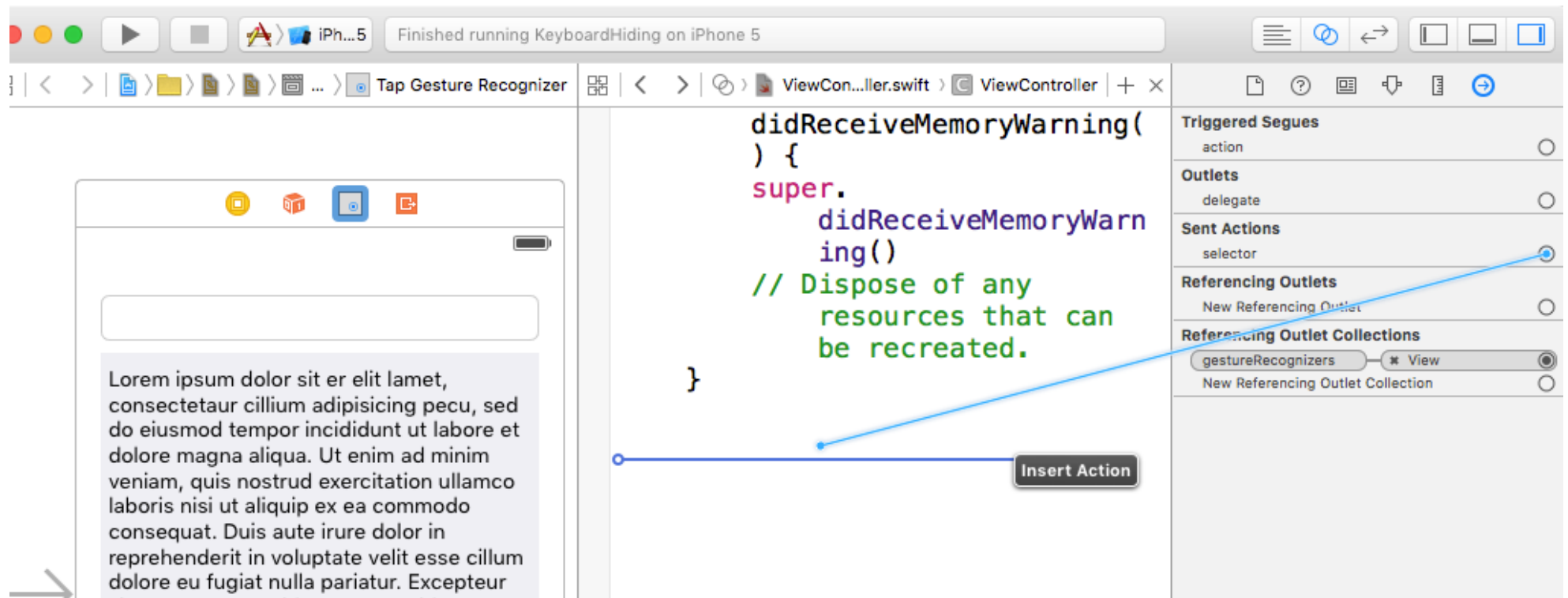




# Link the view controller to the gesture recognizer



# Add an action method for the gesture recognizer



# Method2ViewController.swift

- This example name the method 'tapAction'

```
@IBAction func tapAction(sender: UITapGestureRecognizer)
{
    self.topTextField.resignFirstResponder()
    self.textArea.resignFirstResponder()
}
```

Try running the project. The keyboard will be dismissed when you touch in on the empty space outside text field/view. (You are touching the View)

```
self.view.endEditing(false)
```