



LearnToiOS

Delivered By:



**SECURE
BROADCAST**

CHARGE  **D**

 **NSBelfast**

WiFi: SB Guest
Password: Broadcast2018

#LearnToiOS

Week 3

Closures, Delegates &
UITableViews

Closures

What is a Closure?

- Pass a block of code around. “Closing over” the variables involved
- It’s a reference type!

```
func someFunctionThatTakesAClosure(value: Int, multiplyBy: (Int) -> Int) -> Int {  
    print("Oh hi, LearnTo 🖐️ – thanks for the value \(value)")  
    let result = multiplyBy(value)  
  
    print("Here's the result of what you passed in as a block \(result)")  
    return result  
}
```


What is a Closure?

```
func someFunctionThatTakesAClosure(value: Int, multiplyBy: (Int) -> Int) -> Int {  
    print("Oh hi, LearnTo 🖐️ – thanks for the value \(value)")  
    let result = multiplyBy(value)  
  
    print("Here's the result of what you passed in as a block \(result)")  
    return result  
}
```

```
someFunctionThatTakesAClosure(value: 2) { input in  
    print("We could do some stuff here... probably with the View")  
    let output = input * 2  
    label.text |= String(output)  
    return output  
}
```

```
someFunctionThatTakesAClosure(value: 3) { $0 * 3 }
```

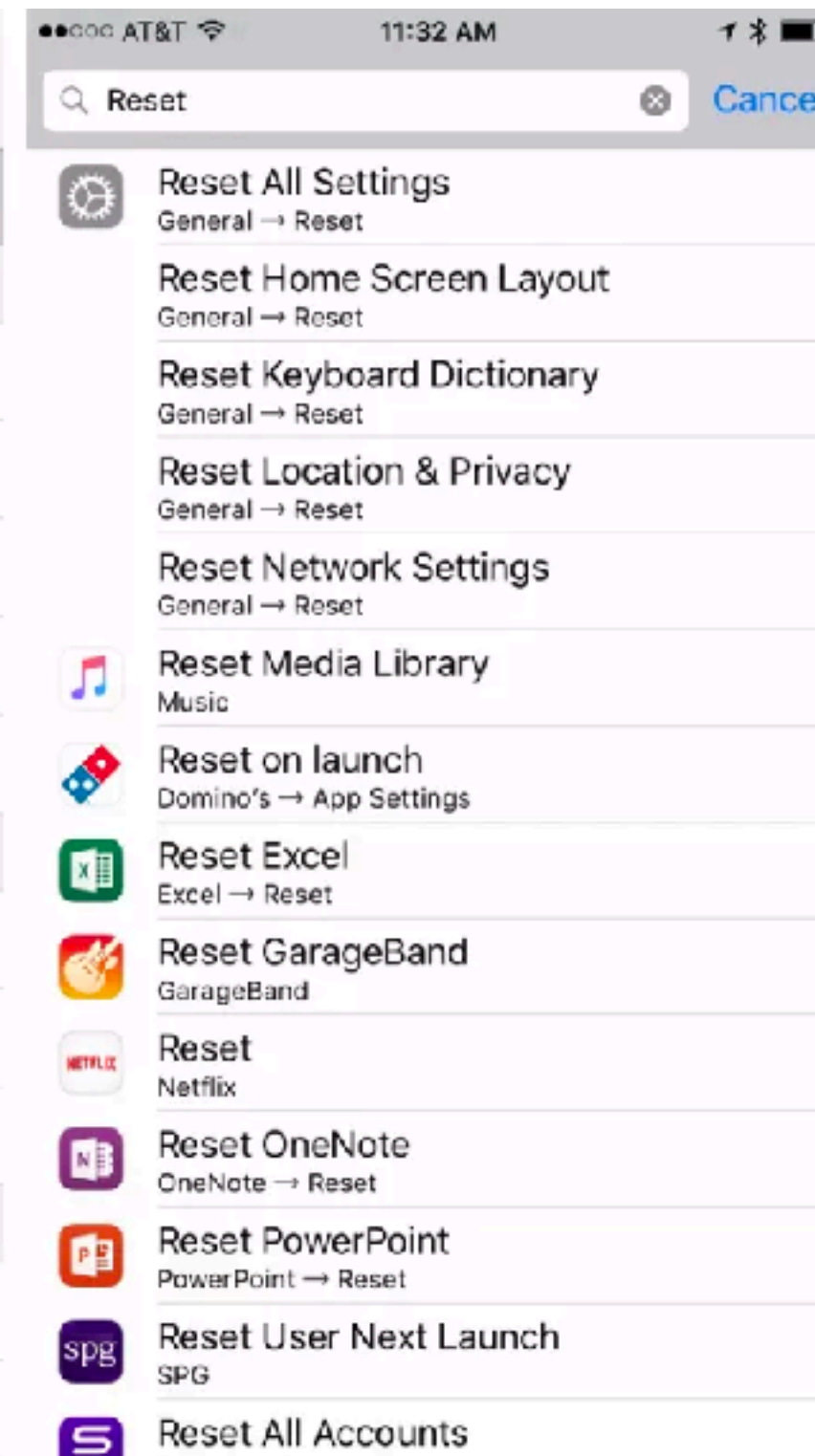
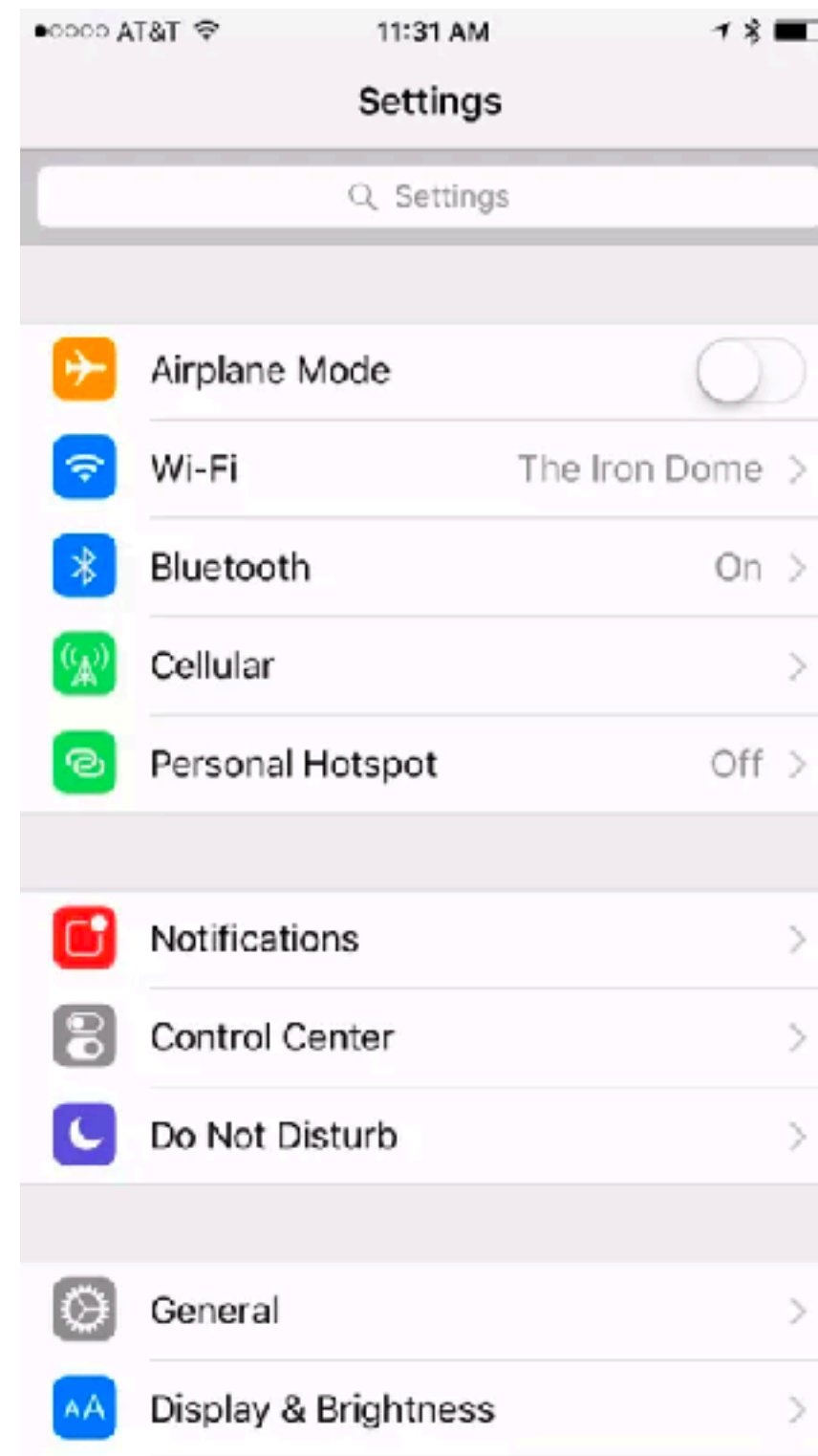
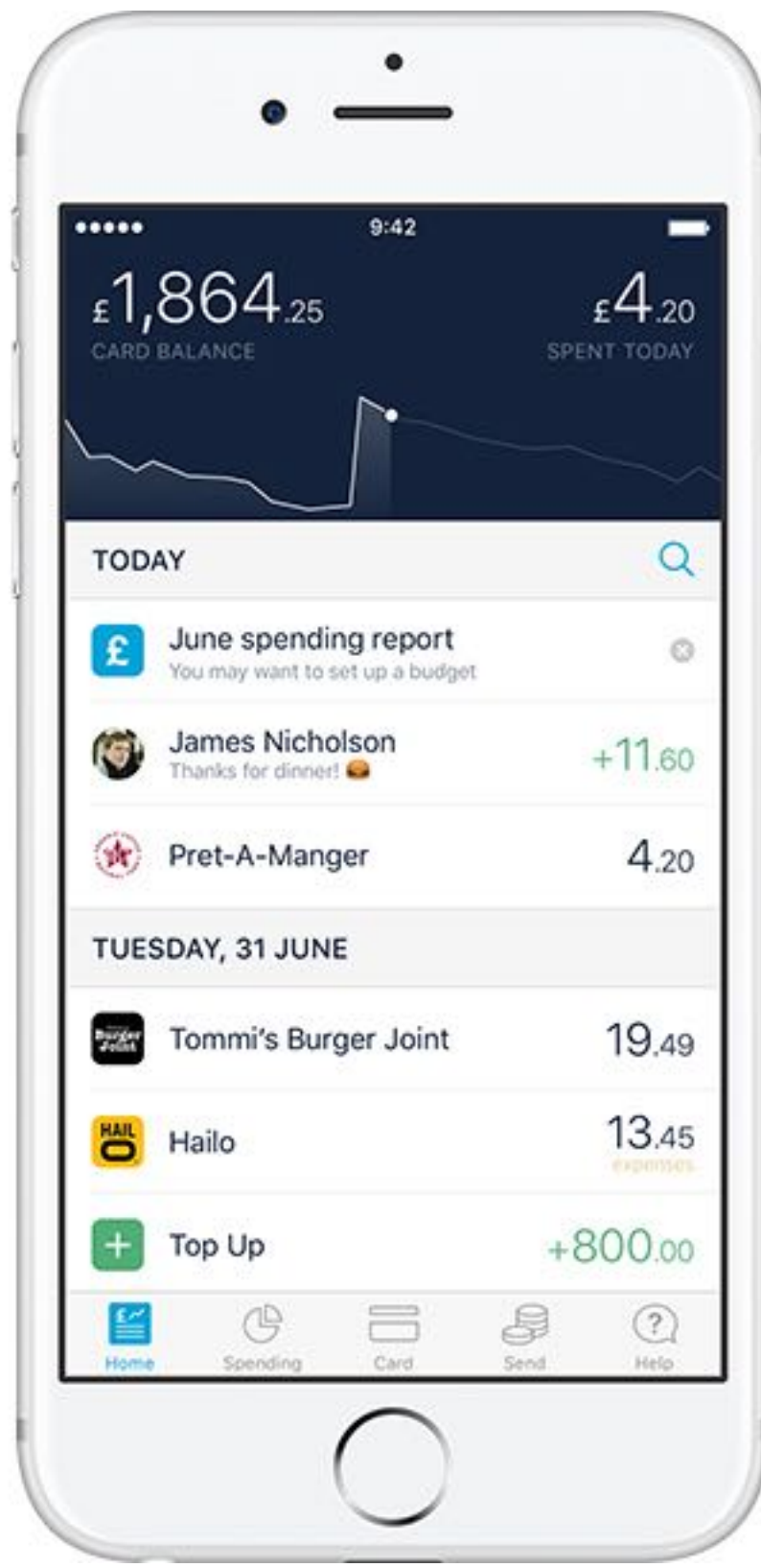
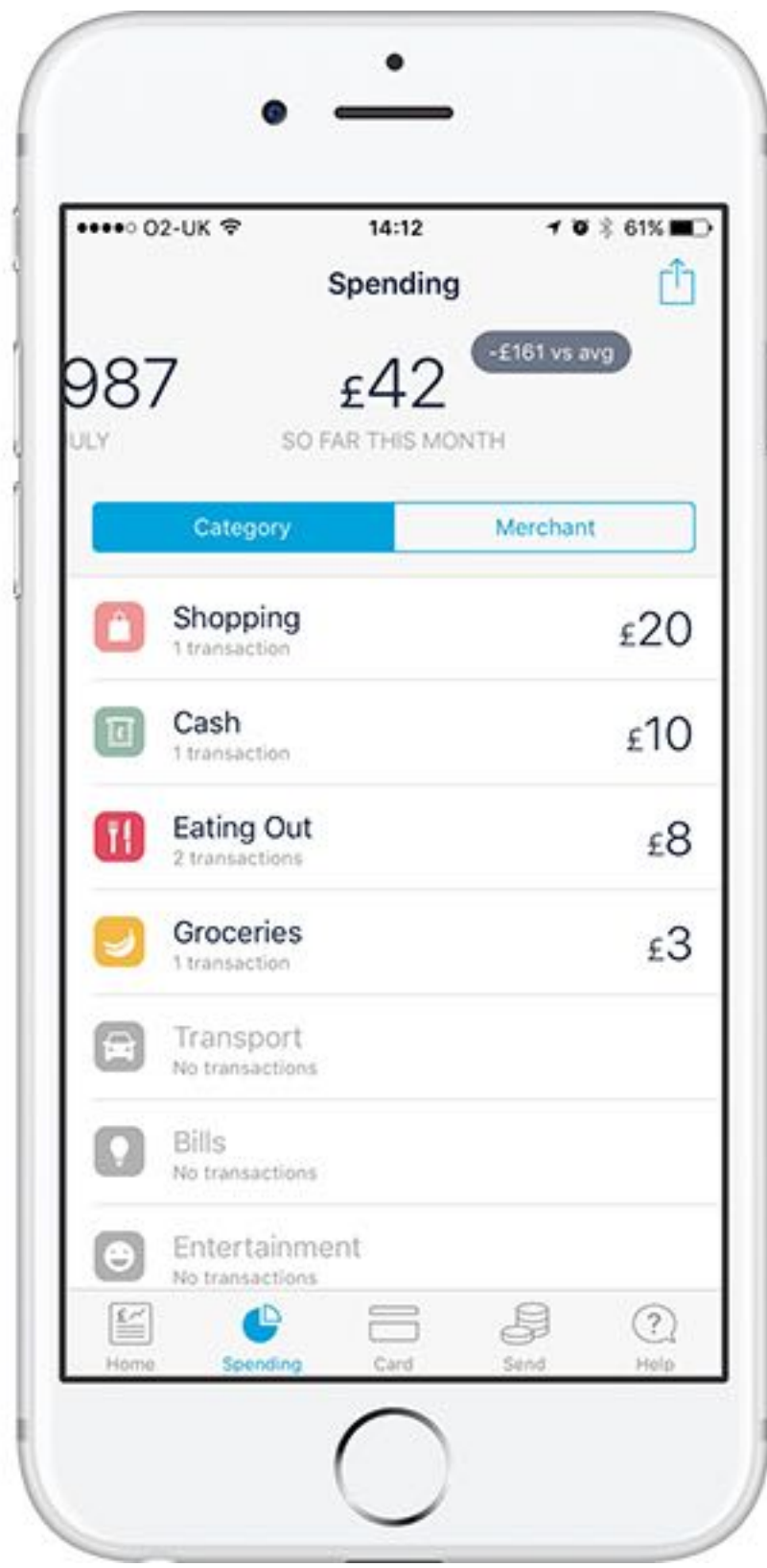
Why a closure?

- Useful for when you want to manipulate the action outside of the class in which it's reference
- Useful for networking (coming soon...) - think of the logic being “when I hear back from the API, perform x on the result”
-  Memory leaks in the form of retain cycles

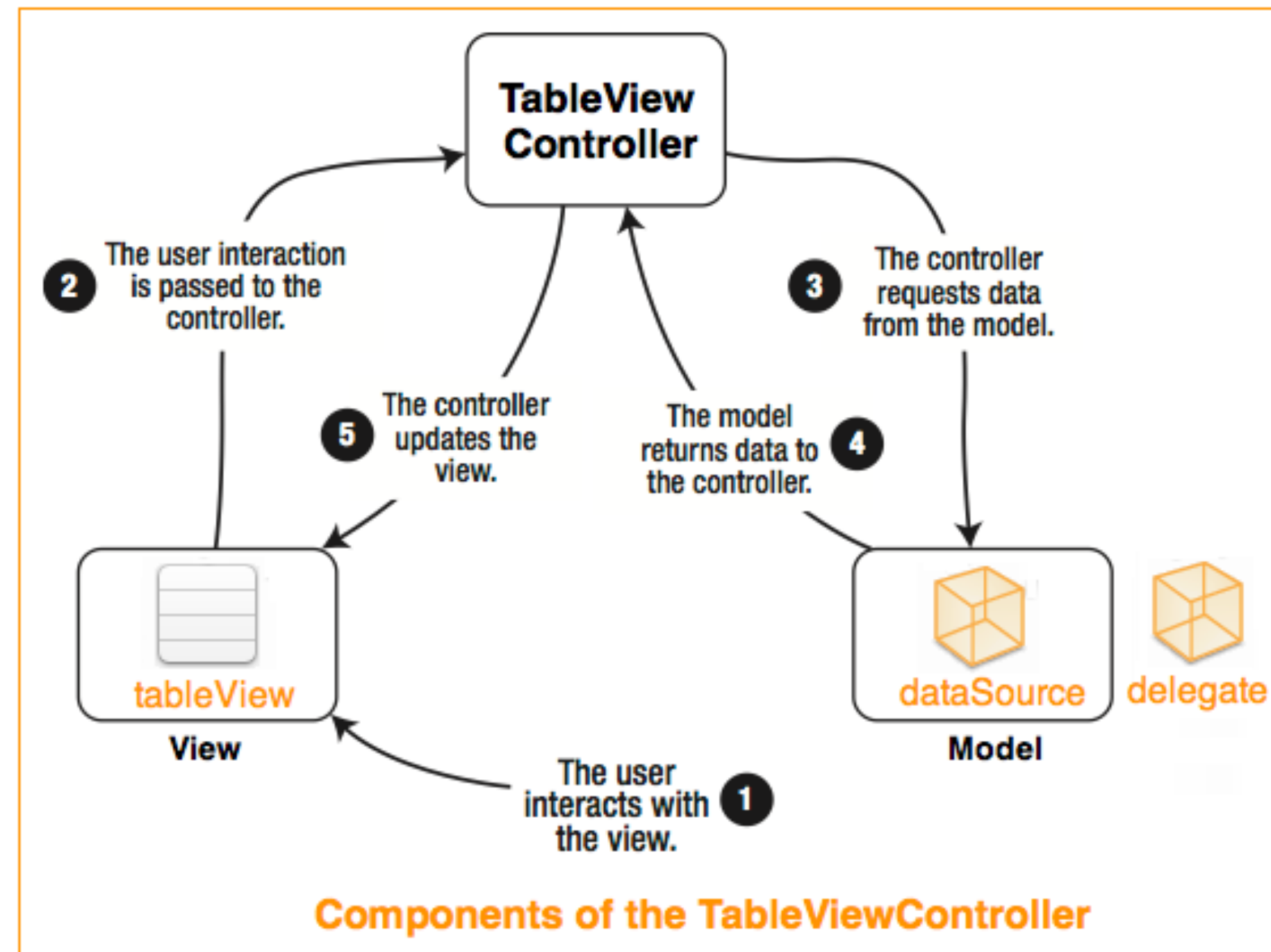
What is Delegation?

- **Protocols** - a class can conform to a protocol (perhaps more later...)
- **Delegate** - “blind communication” between a view and a controller (or any 2 or more classes)
- Allows classes to conform to a certain structure, and allows certain lifecycle events or data events to communicate between classes

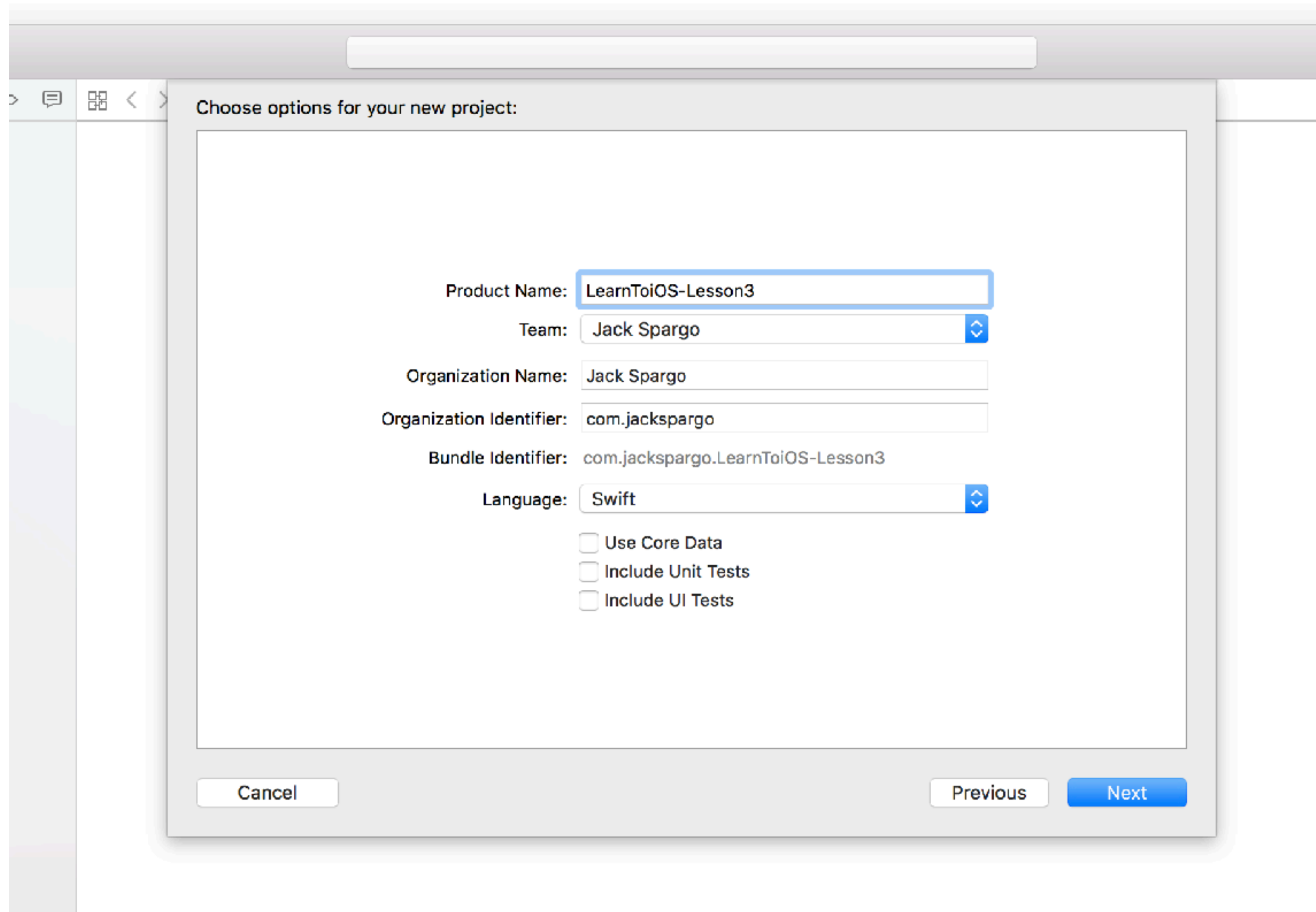
What is a TableView?



TableView Delegation



Let's Create a New App!



The image shows the 'Choose options for your new project' dialog in Xcode. The dialog is a light gray window with a title bar. Inside, there are several text input fields and dropdown menus. The 'Product Name' field is highlighted with a blue border and contains the text 'LearnToiOS-Lesson3'. The 'Team' dropdown menu shows 'Jack Spargo'. The 'Organization Name' field contains 'Jack Spargo'. The 'Organization Identifier' field contains 'com.jackspargo'. The 'Bundle Identifier' field contains 'com.jackspargo.LearnToiOS-Lesson3'. The 'Language' dropdown menu shows 'Swift'. Below these fields are three unchecked checkboxes: 'Use Core Data', 'Include Unit Tests', and 'Include UI Tests'. At the bottom of the dialog are three buttons: 'Cancel', 'Previous', and 'Next'.

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

Bundle Identifier:

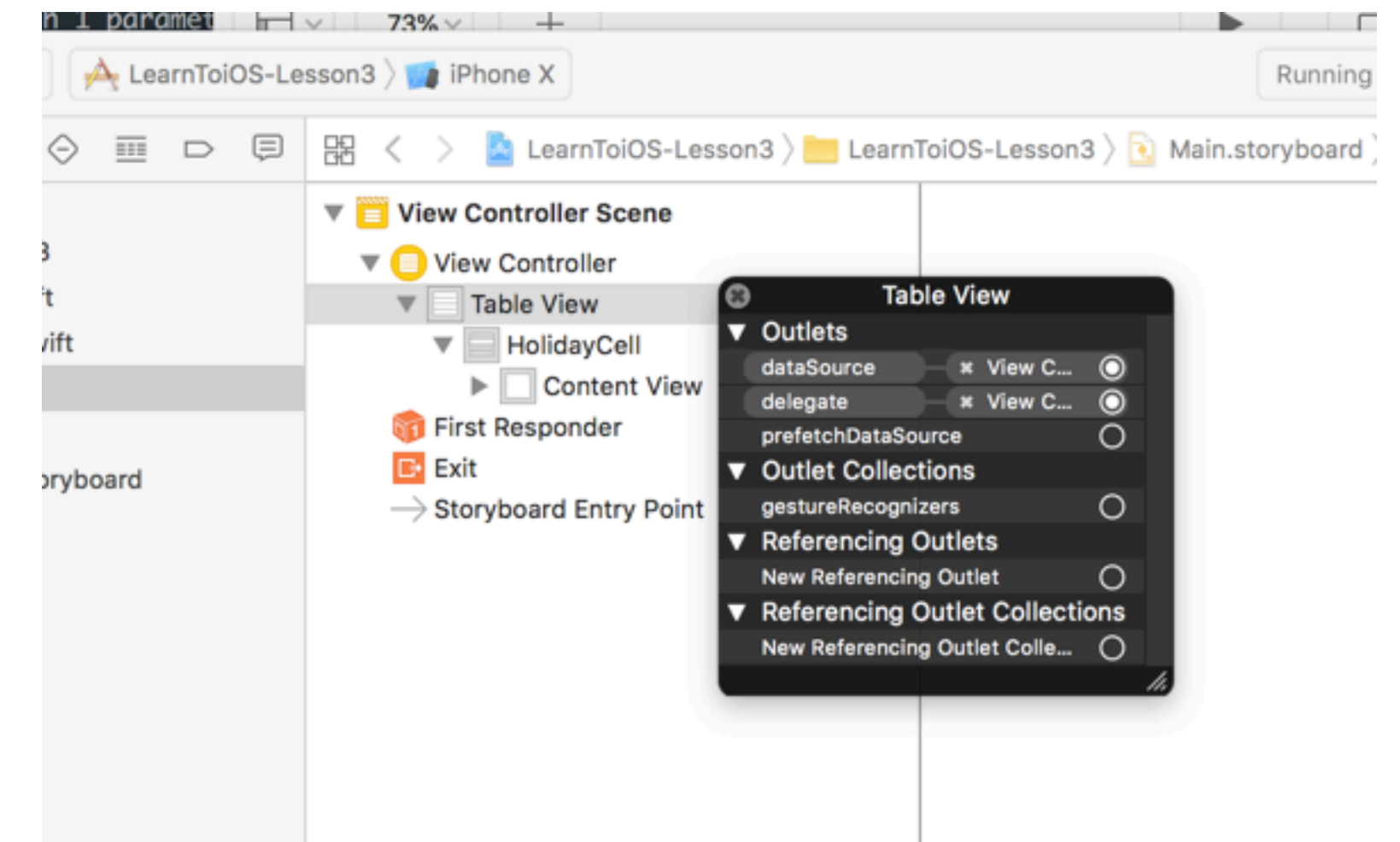
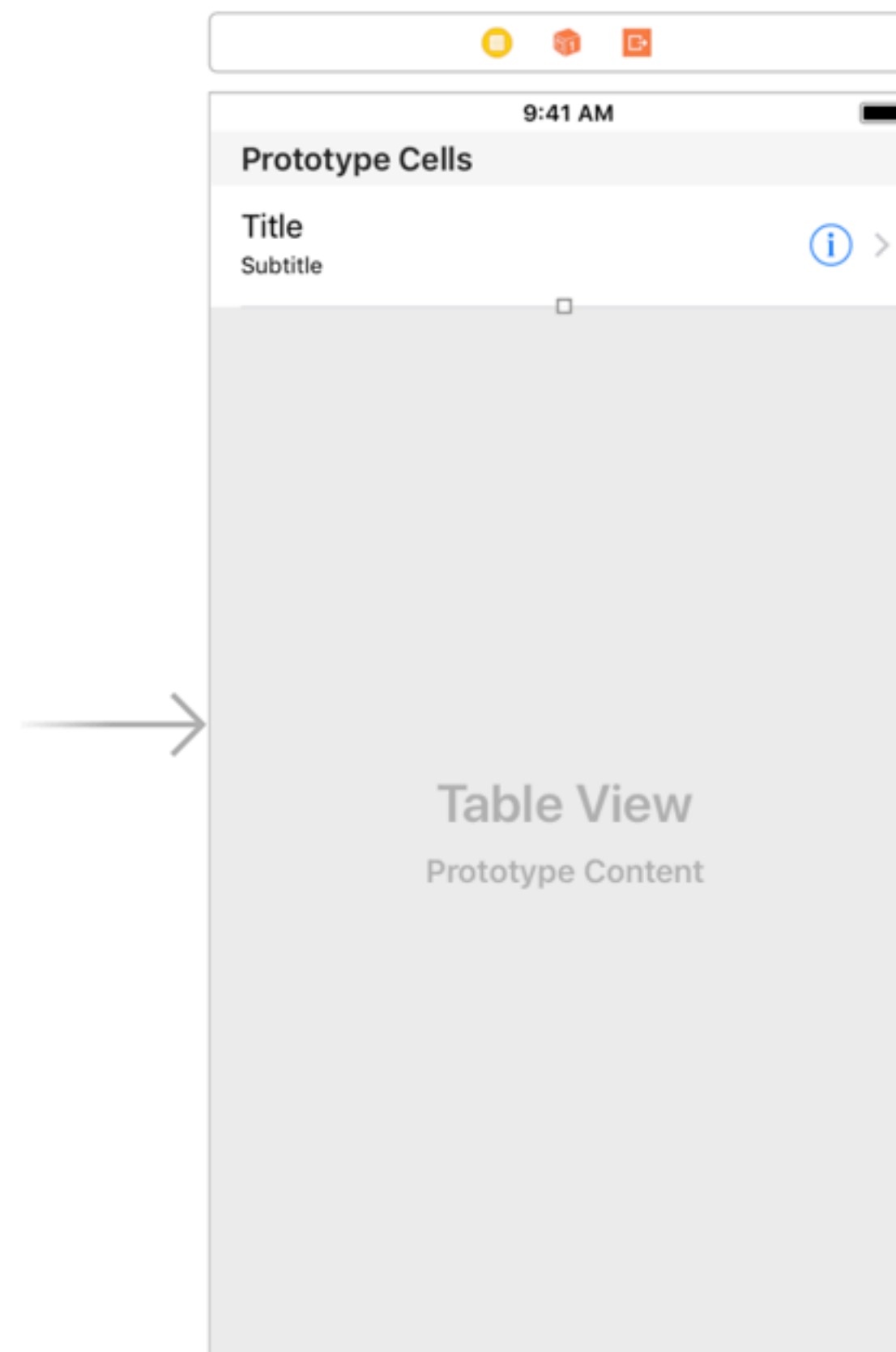
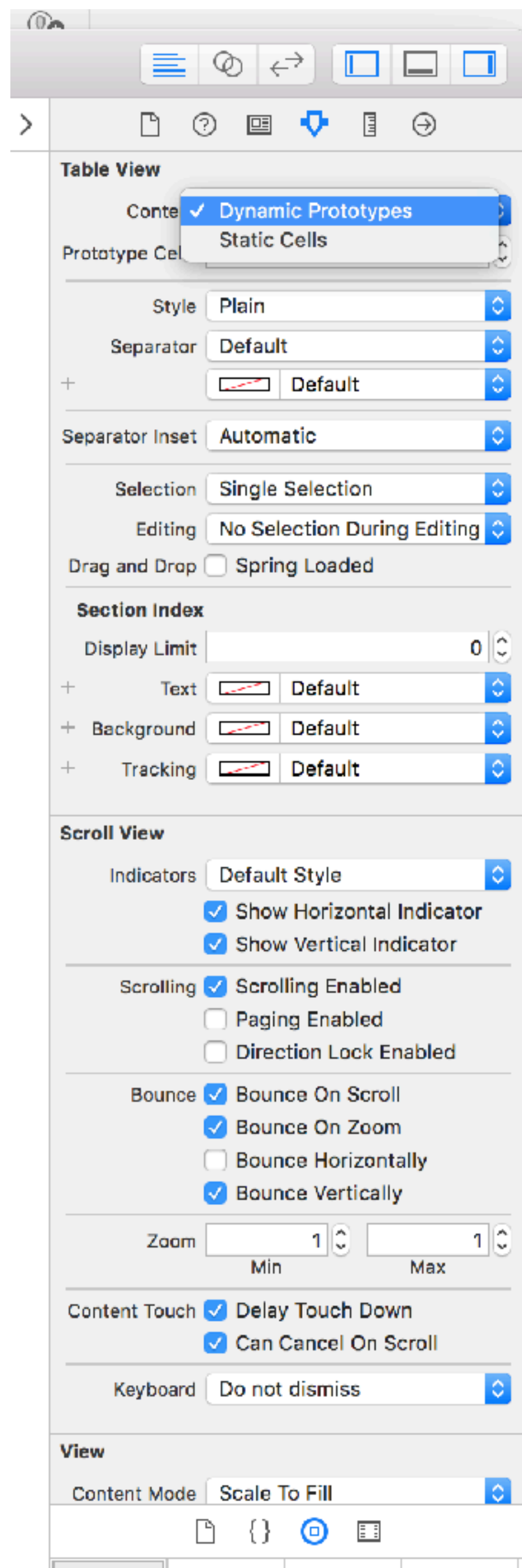
Language:

☐ Use Core Data

☐ Include Unit Tests

☐ Include UI Tests

TableView



First Exercise

Add a new UITableViewController, create some data to show, then display the data in some UITableView cells.

Second Exercise

Create a Custom TableViewCell, hook up some outlets and populate using the TableView delegate methods

Third Exercise

Categorise the data into some sections and display these as sections in the TableView