

Table Views in iOS

Hands-On Challenges

Table Views: Beginning to Advanced Hands-On Challenges

Copyright © 2015 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

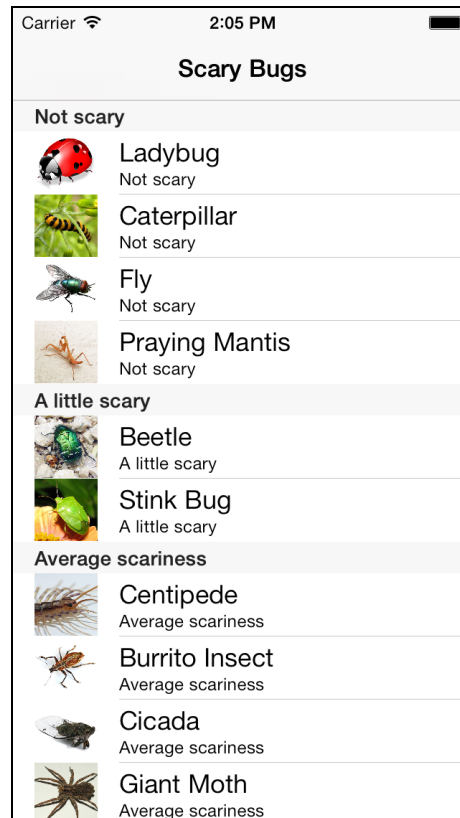
All trademarks and registered trademarks appearing in this book are the property of their respective owners.













Challenge #3: Sectioned Bugs

Your Scary Bugs project is coming along great so far, but right now it's not very easy to distinguish the less scary bugs from the more scary bugs.

Your challenge in this section is to sort the bugs into sections in your table view, so they look something like this:



The screenshot shows a mobile app interface with a status bar at the top displaying 'Carrier', signal strength, '2:05 PM', and battery level. The app title 'Scary Bugs' is centered at the top. Below the title, the bugs are organized into three sections, each with a header and a list of items. Each item consists of a small square image of the bug, followed by the bug's name and a descriptive label.

Scary Bugs		
Not scary		
	Ladybug	Not scary
	Caterpillar	Not scary
	Fly	Not scary
	Praying Mantis	Not scary
A little scary		
	Beetle	A little scary
	Stink Bug	A little scary
Average scariness		
	Centipede	Average scariness
	Burrito Insect	Average scariness
	Cicada	Average scariness
	Giant Moth	Average scariness

See if you can do this on your own based on what you learned on the video. If you get stuck, follow along with the full walkthrough below!

Full Walkthrough

Open the Scary Bugs project where you left it off in the last challenge, or use the starter project provided by the instructor.

You need to organize the list of bugs into different sections to display in the table view. The first step in doing this is to create a class to represent a section of bugs in the table view.



Create a new file with the **iOS\Source\Swift File** template, name it **BugSection**, and make it a subclass of **NSObject**.

Open **BugSection.swift** and replace the contents with the following:

```
class BugSection {

    let howScary: ScaryFactor
    var bugs = [ScaryBug]()

    init(howScary: ScaryFactor) {
        self.howScary = howScary
    }
}
```

You will sort the bugs in the table into sections based on their scary factor. This class represents one section in the table view, so it contains the scary factor and the list of bugs with that scary factor.

Next open **BugTableViewController.swift**.

Rename the property from `bugs` to `bugSections`:

```
var bugSections = [BugSection]()
```

Remember, in order to display the bugs in sections you are going to keep a list of sections, and each section will contain a list of bugs.

Next add this method to sort the bugs into sections:

```
private func setupBugs() {
    bugSections.append(BugSection(howScary: .NotScary))
    bugSections.append(BugSection(howScary: .ALittleScary))
    bugSections.append(BugSection(howScary: .AverageScary))
    bugSections.append(BugSection(howScary: .QuiteScary))
    bugSections.append(BugSection(howScary: .Aiiiiiieeeee))

    let bugs = ScaryBug.bugs()
    for bug: ScaryBug in bugs {
        let bugSection = bugSections[bug.howScary.rawValue]
        bugSection.bugs.append(bug)
    }
}
```

Here you create a section for each scary factor, and sort the bugs into the appropriate sections.

Replace `viewDidLoad` with the following:



```
override func viewDidLoad() {  
    super.viewDidLoad()  
    setupBugs()  
}
```

This calls the method you just wrote when the view is loaded. Next, implement these two new methods:

```
override func tableView(tableView: UITableView,  
    titleForHeaderInSection section: Int) -> String? {  
  
    let bugSection = bugSections[section]  
    return ScaryBug.scaryFactorToString(bugSection.howScary)  
}  
  
override func tableView(tableView: UITableView,  
    numberOfRowsInSection section: Int) -> Int {  
  
    let bugSection = bugSections[section]  
    return bugSection.bugs.count  
}
```

These are the two methods you generally want to implement when you want to have sections in your table view.

1. **numberOfSectionsInTableView(_:)** allows you to configure how many sections to display in the table. Here you simply return the number of sections you have in your `bugSections` array.
2. **tableView(_:titleForHeaderInSection:)** allows you to configure the text that appears in the header for each section. Note that if you want a custom look for the header, you'd probably want to implement `tableView:viewForHeaderInSection:` instead.

Next, replace **tableView(_:numberOfRowsInSection:)** with the following:

```
override func numberOfSectionsInTableView(tableView: UITableView)  
    -> Int {  
  
    return bugSections.count  
}
```

Here you look up the appropriate `BugSection` object, and return the count of the bugs in that section.

Finally, inside `tableView(_:cellForRowAtIndexPath:)`, replace the line that declares the `scaryBug` with the following:



```
let bugSection = bugSections[indexPath.section]
let bug = bugSection.bugs[indexPath.row]
```

This looks up the appropriate `BugSection` based on the `indexPath.section`, then looks up the appropriate bug from that section based on the `indexPath.row`.

And that's it; you're done! Build and run, and enjoy your now sorted list of bugs.

