Table Views in iOS

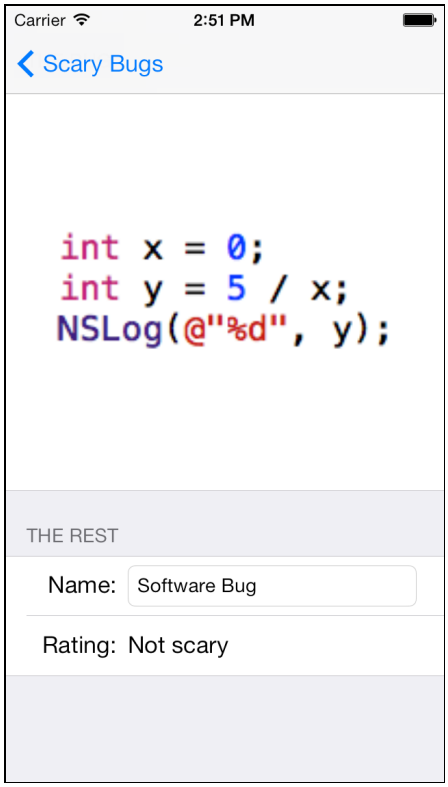
Hands-On Challenges

Table Views: Beginning to Advanced Hands-On Challenges

Challenge #8: Static Cells

You can add, delete, and move bugs, but your app is currently lacking one critical feature: editing bugs!

Your challenge this time is to add an editing view into the app. You should create the editing view using a table view controller with static cells, like this:



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!

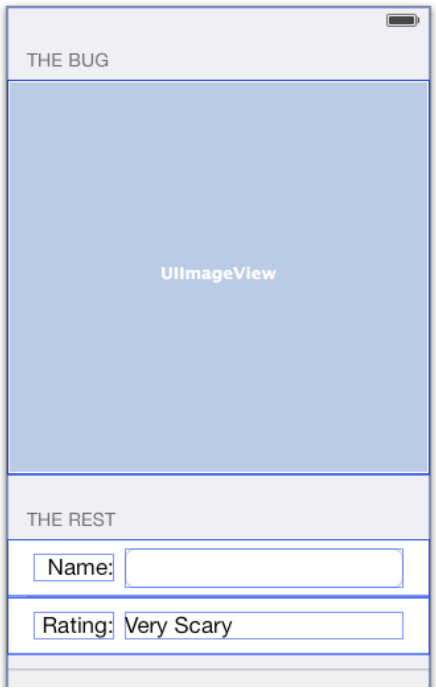
Full Walkthrough

> **Note**: You're getting more advanced at this point, so this time I am not listing out each and every step so it's a bit more of a challenge. If you get stuck, refer back to the video or check out the challenge solution. Good luck!

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

Open **Main.storyboard** and drag a new table view controller into the canvas. Select the table view and set the **Content** to **Static Cells**, the **Sections** to **2**, and the **Style** to **Grouped**.

Then lay out your static cells so they look something like this:



Select the two cells under "The Rest" and in the Attributes Inspector (3rd tab) set the **Selection** to **None**.

Next, create a new class named **EditTableViewController** that derives from **UITableViewController**. Switch back to **Main.storyboard**, and set the class of the view controller to **EditTableViewController** in the identity inspector.

Using the assistant editor, connect the image view to an outlet in **EditViewController.swift** named **bugImageView**, the name text field to an outlet named **bugNameTextField**, and the rating label ot an outlet named **bugRatingLabel.**

Finally, set the `EditViewController` as the delegate of the name text field.

Next open **EditTableViewController.swift** and a new property for the bug to edit underneath the opening class brace:

```
var bug: ScaryBug?
```

Add the following methods:

```
override func viewWillAppear(animated: Bool) {
  super.viewWillAppear(animated)
```

```
    guard let bug = bug else {
      return
    }
    if let bugImage = bug.image {
      bugImageView.image = bugImage
    }
    bugNameTextField.text = bug.name
    bugRatingLabel.text = ScaryBug.scaryFactorToString(bug.howScary)
  }

  override func viewWillDisappear(animated: Bool) {
    super.viewWillDisappear(animated)
    bug?.image = bugImageView.image
    bug?.name = bugNameTextField.text!
  }
```

These methods work to populate the view controller when it is first displayed, then clears it out when it is dismissed.

Next, replace all the existing table view controller methods with the following:

```
  override func tableView(tableView: UITableView,
    didSelectRowAtIndexPath indexPath: NSIndexPath) {

    if indexPath.row == 0 && indexPath.section == 0 {
      tableView.deselectRowAtIndexPath(indexPath, animated: true)
      let picker = UIImagePickerController()
      picker.sourceType = .PhotoLibrary
      picker.allowsEditing = false
      picker.delegate = self
      presentViewController(picker, animated: true,
        completion: nil)
    }
  }
```

By tapping the row with the image view, the user will be presented with the ability to select an image from their own their own photo library

> **Note:** You can learn more about working with the image pickers by watching our **iOS 101 video tutorial series**.

Finally, underneath the class definition, add the following extension.

```
extension EditTableViewController: UITextFieldDelegate,
  UIImagePickerControllerDelegate,
```

```
    UINavigationControllerDelegate {

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

  func imagePickerController(picker: UIImagePickerController,
    didFinishPickingMediaWithInfo info: [String : AnyObject]) {

    if let image = info[UIImagePickerControllerOriginalImage]
      as? UIImage {

      bug?.image = image
      bugImageView.image = image
      dismissViewControllerAnimated(true, completion: nil)
    }
  }
}
```

First, the code implements the UITextFieldDelegate protocol. By implementing `textFieldShouldReturn(_:)`,you allow the keyboard to be dimissed when the user taps on the return key.

The UIImagePickerControllerDelegate and UINavigationControllerDelegate allow the view controller to respond when an image has been picked. If one has been picked, then the bug image view is replaced with the user selected one.

That's it for your new view controller; you just need to configure the bug list view controller to display it. To do this, open **Main.storyboard** and control-drag from your custom bug cell to your new view controller. Choose **show**, and name the segue **GoToEdit**.

Then open **BugTableViewController.swift** and add the following:

```
override func prepareForSegue(segue: UIStoryboardSegue,
  sender: AnyObject?) {

  if segue.identifier == "GoToEdit" {
    if let editController = segue.destinationViewController as?
      EditTableViewController {
      if let indexPath  = tableView.indexPathForSelectedRow {
        let bugSection = bugSections[indexPath.section]
        let bug = bugSection.bugs[indexPath.row]
        editController.bug = bug
      }
    }
```

```
    }
}
```

Build and run, and you should now be able to edit bugs!