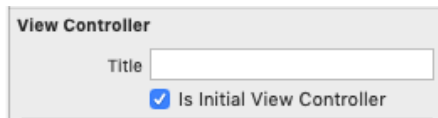


In-class activity on Navigation Controller with Table View

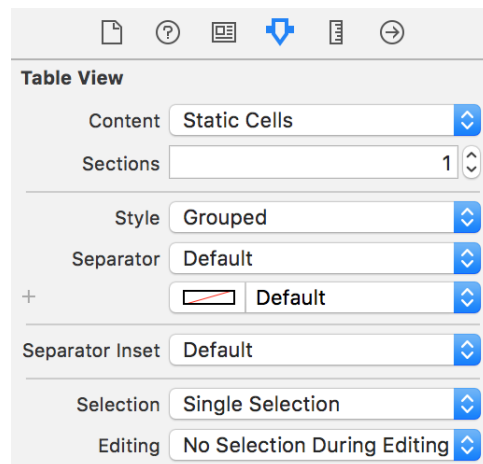
1. Create a single view app with XCode and name it as **NavigationControllerTableView**.
2. Go to the Storyboard → delete the View Controller → open object library and drag a **Navigation Controller** to the Storyboard → Select the Navigation Controller, go to the Attribute Inspector, then check the “**Is Initial View Controller**” checkbox



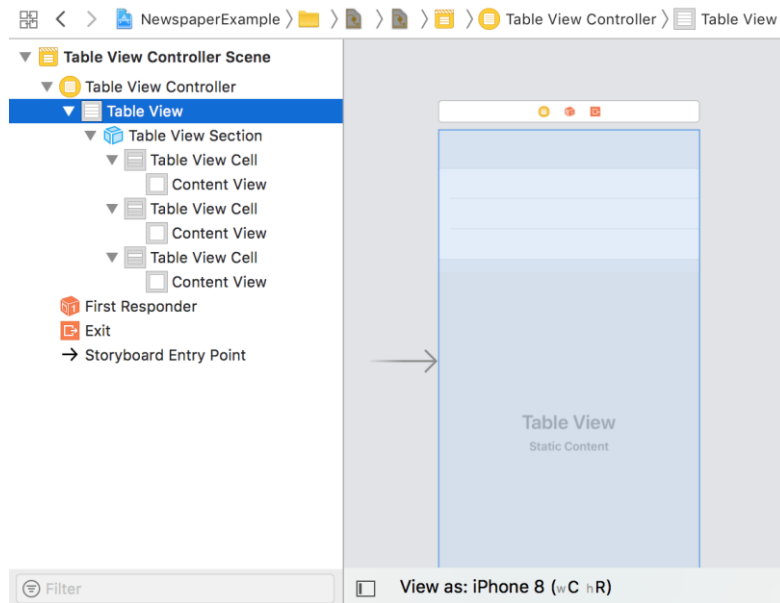
The Navigation Controller will automatically contain a Table View Controller.

3. Let's first try to set table cells as static cells directly from storyboard:

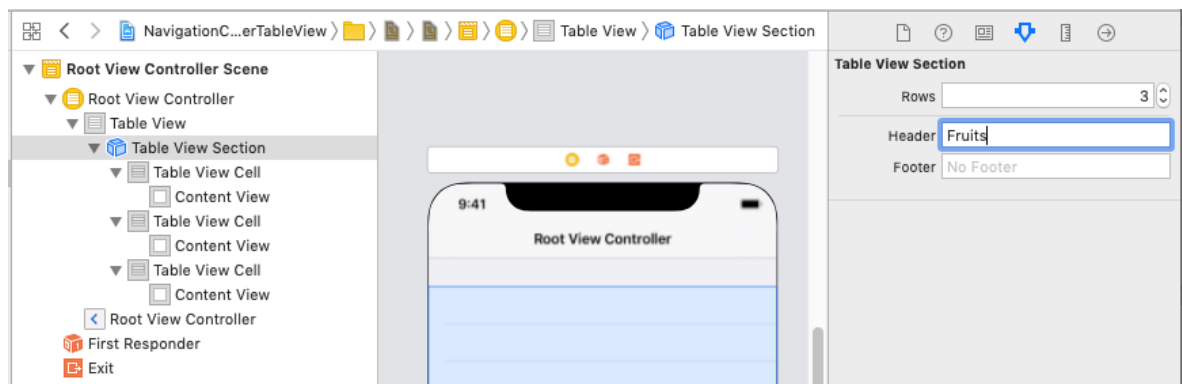
Select **Table View** in the Table View Controller. In the *Attributes Inspector* configure *Static Cells* for *Content* to configure the Table View cells in the storyboard instead of writing code to provide the data. Set the *Style* of the *Table View* to *Grouped*:



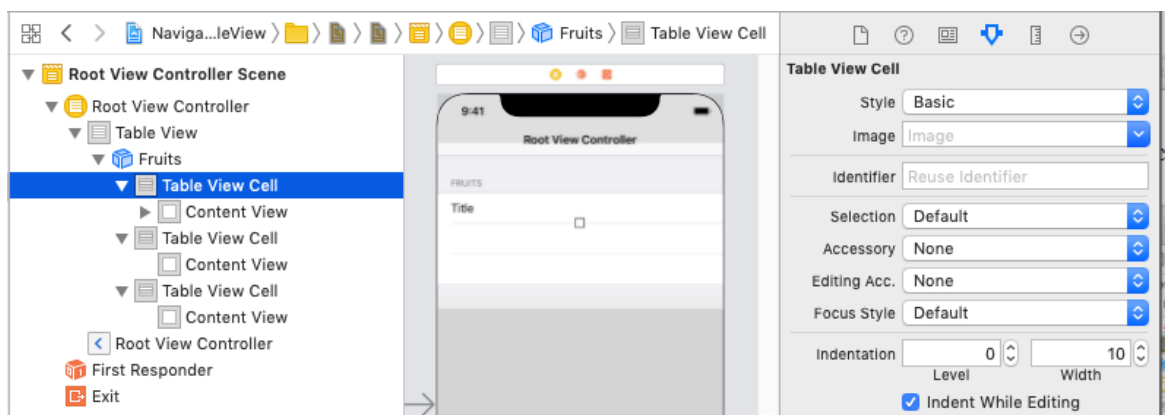
4. Show the *Document Outline* and familiarize yourself with the object structure. The *Table View Controller* manages its *Table View* consisting of a single *Table View Cell* row and an empty *Content View* in the cell:



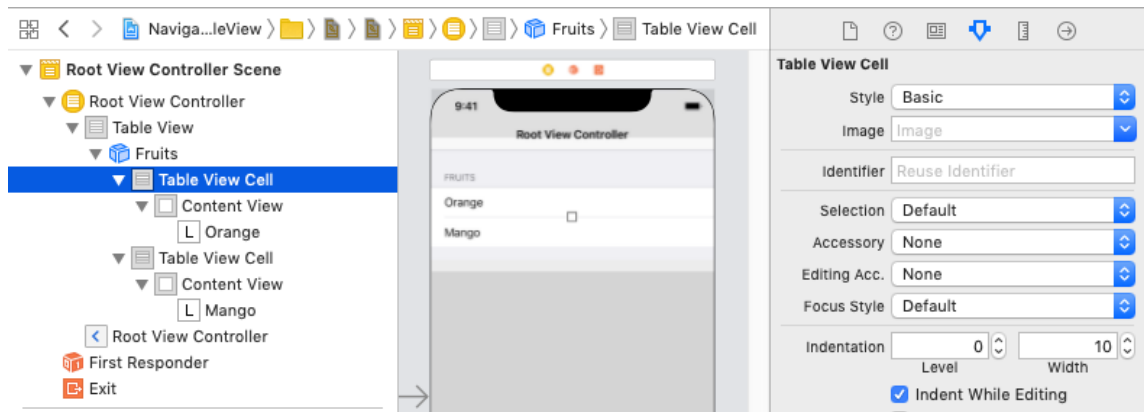
5. Select the **Table View Section** using the outline and go to Attributor Inspector, enter a title “Fruits” into the header:



6. Select the first cell and configure *Basic* as *Style* for all cells. A *Basic Cell* has one *Label* by default:



7. Configure *Basic* as *Style* for the second cell as well and delete the third cell. Configure some label texts:



8. Run the app to see how it looks.

9. Next we try set table cells as dynamic cells, and provide section and row content programmatically:

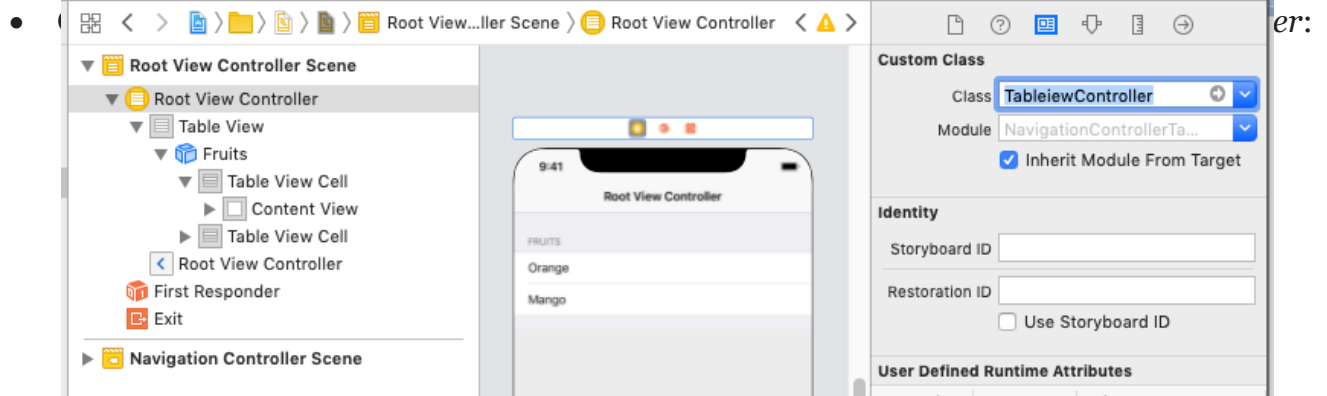
- Create a new swift file, and leave the original **ViewController.swift** file to be used later for another View Controller.

Right click on the project folder → New File... → select **Swift File** → Next → Save As: **TableViewController** → Create. Modify the file content into something like this:

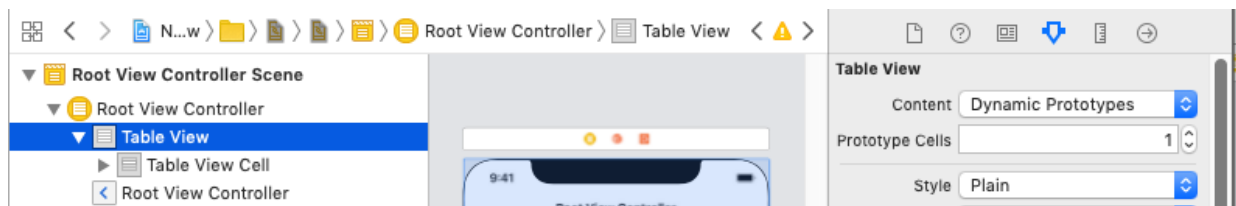
```
import UIKit

class TableViewController: UITableViewController {

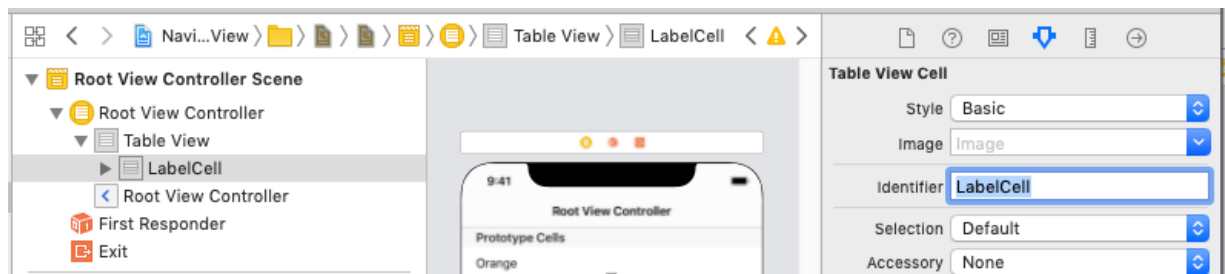
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
}
```



- Select the *Table View* and configure *Dynamic Prototypes* for *Content* to define a prototype for the cells in the storyboard but to provide the content in the controller class. Configure 1 *prototype* cell and select *Plain* as *Style*:



- Select one *Table View Cell* and configure *LabelCell* as *Identifier*. Delete other Table View cell(s). The *identifier* is used in the controller implementation to create cells according to the prototype from the Storyboard:



- In the *TableViewController* class, customize the *numberOfSectionsInTableView* and *tableView:numberOfRowsInSection:* methods from the [UITableViewDataSource](#) protocol to return a fixed number of sections and rows and remove the *#warning* comments:

// MARK: - Table view data source

```
override func numberOfSectionsInTableView (in tableView: UITableView) -> Int {
    return 3    //exact number to be determined later
}
```

```
override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 5    // exact number to be determined later  
               // with the data source we used  
}
```

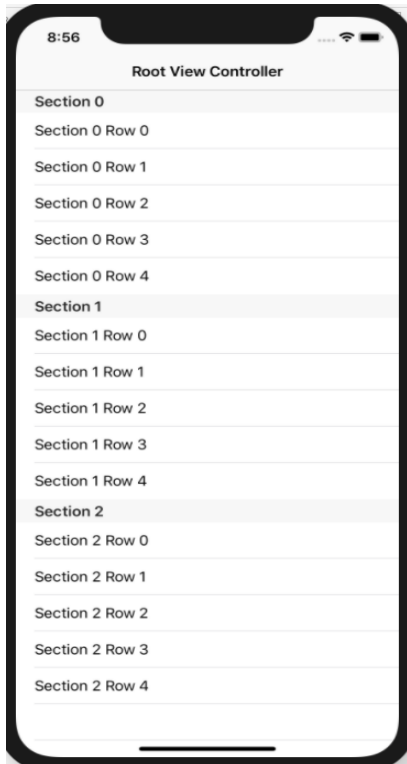
- Add the *tableView:cellForRowAtIndexPath:* method in the class and customize it to create cells according to the *Prototype cell* and configure the cell text to show the section and row numbers:

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    let cell = tableView.dequeueReusableCell(withIdentifier: "LabelCell", for: indexPath)  
  
    cell.textLabel?.text = "Section \(indexPath.section) Row \(indexPath.row)"  
  
    return cell  
}
```

- Implement the *tableView:titleForHeaderInSection:* method to return a title according to the section number:

```
override func tableView(_ tableView: UITableView, titleForHeaderInSection section: Int) -> String? {  
    return "Section \(section)"  
}
```

10. Run the app and see how it looks!



11. Now let's try with some data to provide data source to the app:

- Add a Struct named **Headline**, and an array named **headlines** like this below:

```
import UIKit

struct Headline {
    var id : Int
    var title : String
    var text : String
    var image : String
}

class TableviewController: UITableViewController {
    var headlines = [
        Headline(id: 1, title: "Apple", text: "The apple tree (Malus domestica) is a tree that grows fruit in the rose family best known for its juicy, tasty fruit. ", image: "Apple"),
        Headline(id: 2, title: "Banana", text: "Most banana plants are grown for their fruits, which botanically are a type of berry.", image: "Banana"),
        Headline(id: 3, title: "Cantaloupe", text: "The cantaloupe, rockmelon (Australia), sweet melon, or spanspek (South Africa) is a melon that is a variety of the muskmelon species (Cucumis melo) from the family Cucurbitaceae.", image: "Cantaloupe"),
    ]
}
```

- Customize the *TableViewController* class to use data from the *Array*:

```

override func numberOfSections (in tableView: UITableView) -> Int {
    return 1
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
    return headlines.count // exact number to be determined later
                          // with the data source we used
}

override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "LabelCell", for: indexPath)

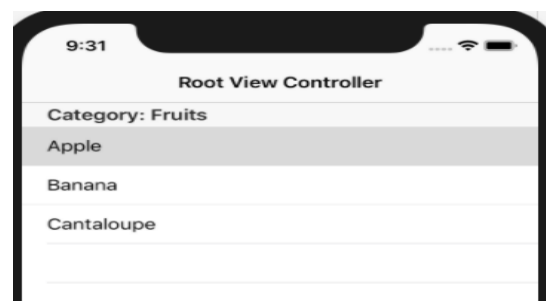
    cell.textLabel?.text = headlines[indexPath.row].title

    return cell
}

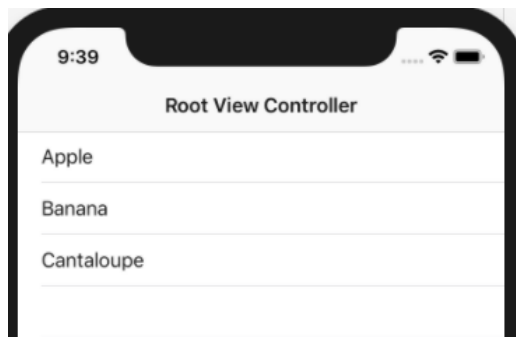
override func tableView(_ tableView: UITableView, titleForHeaderInSection section: Int)
-> String? {
    if section == 0 {
        return "Category: Fruits"
    }
    else if section == 1 {
        return "Category: Wines"
    }
    else {
        return "Category: Others"
    }
}

```

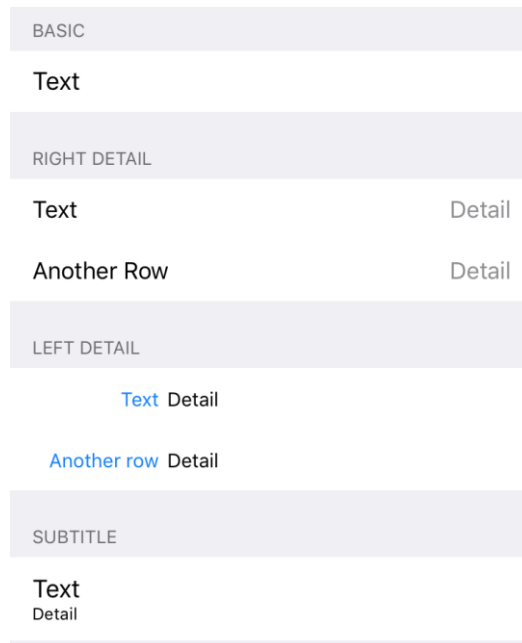
- Run the app to see how it looks.



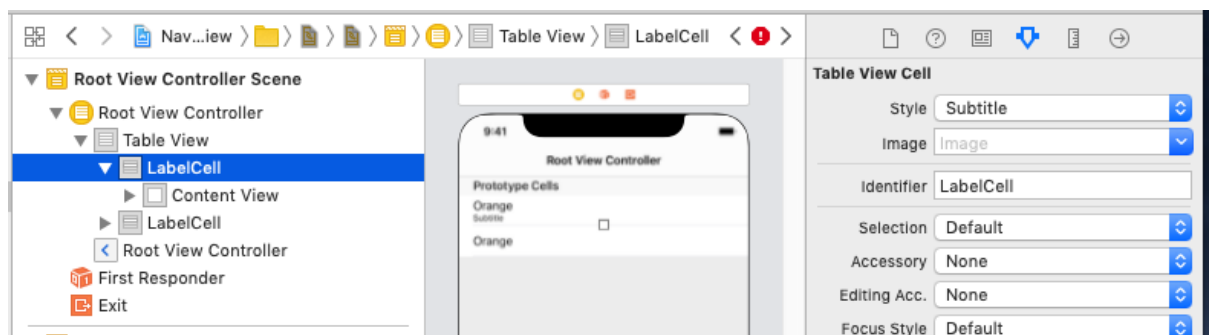
- If there is only one Section, we **can remove two methods** from *TableViewController* class: **numberOfSections()** and **TableView (titleForHeaderInSection)**. Remove or comment out the two methods and run the app again to see how it looks:



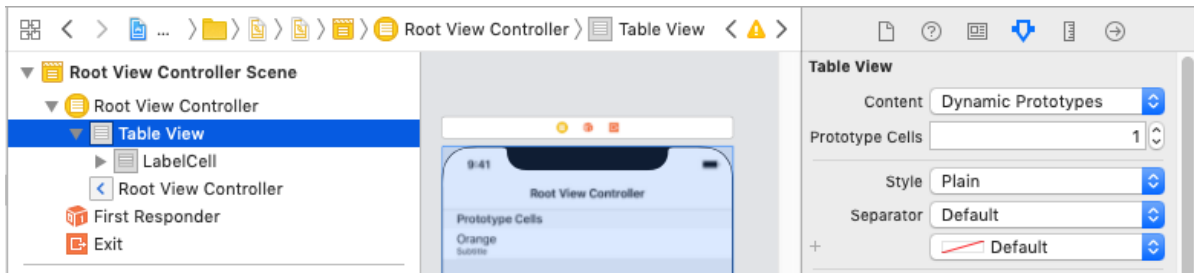
13. Next we'll configure cells using cell styles. There are four different types of cell styles that have views out of the box - *Basic*, *Left/Right Detail* and *Subtitle*:



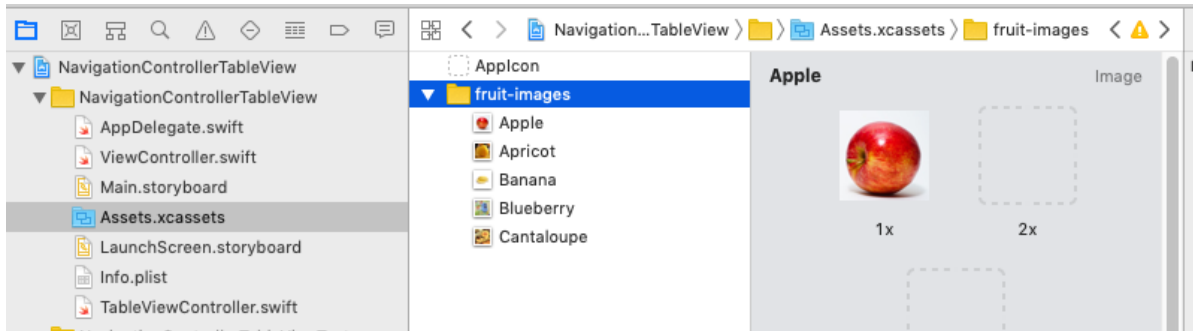
- For the *TableViewController* in the storyboard, select the Table View Cell (LabelCell) and set the style *Subtitle*:



- Select *Table View* and go to Attributes Inspector, change prototype cells from 2 to 1:



- Drag the fruit-image folder (available from Moodle) into Assets.xcassets area of the project.

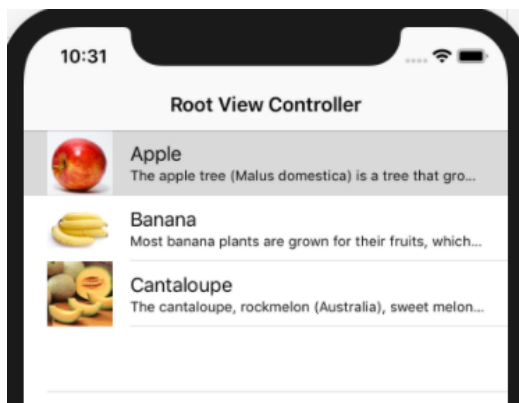


- Modify the tableView (cellForRowAt indexPath) method to set the title, text and image for each cell :

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "LabelCell", for: indexPath)
    let headline = headlines[indexPath.row]
    cell.textLabel?.text = headlines[indexPath.row].title
    cell.detailTextLabel?.text = headline.text
    cell.imageView?.image = UIImage(named: headline.image)

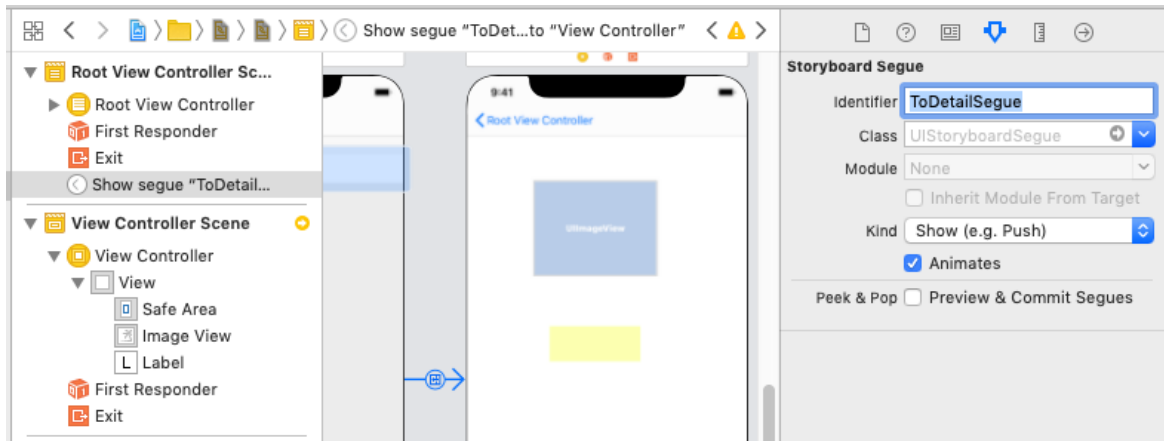
    return cell
}
```

14. Run the app to see how it looks.

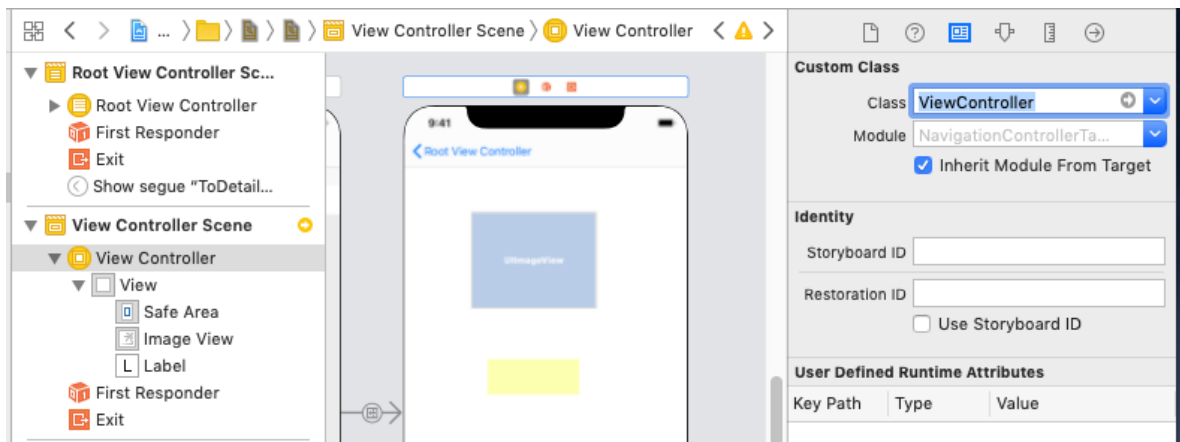


15. Next we'll work on a new View Controller so that the app can respond and go to the new View Controller using a segue when one of table cell item is selected.

- Add a new View Controller into storyboard next to UITableViewController (Root View Controller) → Add an image view and a label into the new View Controller → control-drag from **Table View Cell (LabelCell)** to the new View Controller → select **Show** under Selection Segue menu → assign the segue with an identifier name: **ToDetailSegue**



- Link the new View Controller to the original ViewController.swift file (which we haven't used yet):



- Connect the new View Controller's image view and label to its associated swift file (ViewController.swift) as new referencing outlets (viewImage and viewLabel):

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var viewImage: UIImageView!
    @IBOutlet weak var viewLabel: UILabel!
```

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view.
}

```

- Add a String variable named `titleStringViaSegue` and one method named `viewWillAppear()` to accept the data via the Segue. With the data received, we display the data (which is the name of a fruit like “Apple”, “Banana”, or “Cantaloupe”) and we can determine what image file to display. The file should now look like this:

```

import UIKit

class ViewController: UIViewController {
    var titleStringViaSegue : String?

    @IBOutlet weak var viewImage: UIImageView!
    @IBOutlet weak var viewLabel: UILabel!

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

        if titleStringViaSegue == "Cantaloupe"
        {
            viewImage.image = UIImage(named: "Cantaloupe")
        }
        else if titleStringViaSegue == "Banana" {
            viewImage.image = UIImage (named: "Banana")
        }
        else {
            viewImage.image = UIImage (named: "Apple")
        }

        viewLabel.text = titleStringViaSegue
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
}

```

- Now let switch to the Root View Controller and its associated file (TableViewController.swift). Add two methods at the bottom into the class file:

```

func tableView(tableView: UITableView, didSelectRowAt indexPath: IndexPath){

    performSegue(withIdentifier: "ToDetailSegue", sender: self)

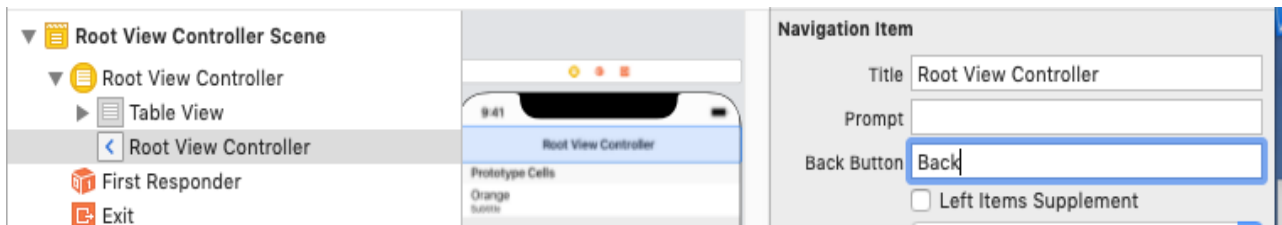
}

override func prepare (for segue: UIStoryboardSegue, sender: Any?){
    if segue.identifier == "ToDetailSegue" {
        let sdest = segue.destination as? ViewController
        let indexPath = tableView.indexPathForSelectedRow
        let titleString = headlines[(indexPath?.row)!].title

        sdest?.titleStringViaSegue = titleString
    }
}

```

16. Run the app to see the results. If you don't like the back button being labelled with "Root View Controller", you can change it by selecting the Navigation item on the Document Outline, go to Attributes Inspector and type in text like "Back" for the Back Button.



- Take screen shots of the results and submit them.