

Tables

Mobile Application Development in iOS

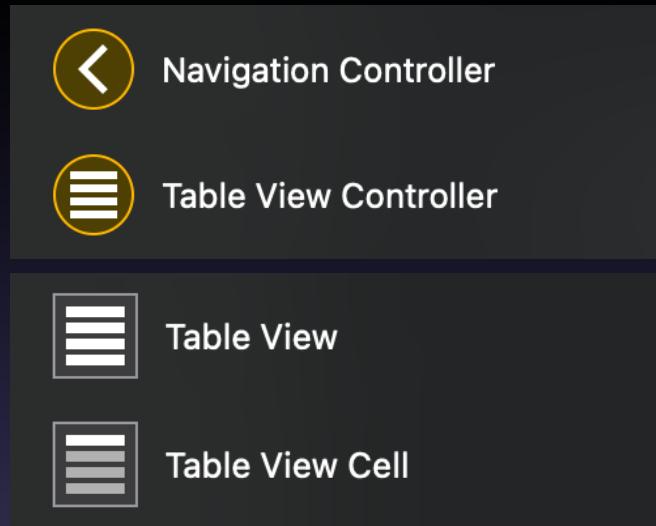
School of EECS

Washington State University

Instructor: Larry Holder

Outline

- Table View Controller



- Table View

- Table Cells

- Cell interaction

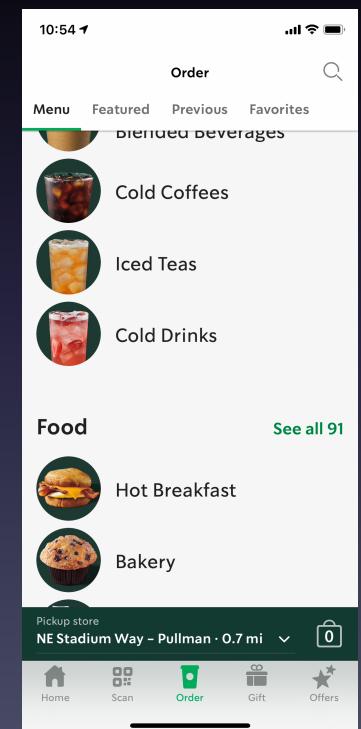
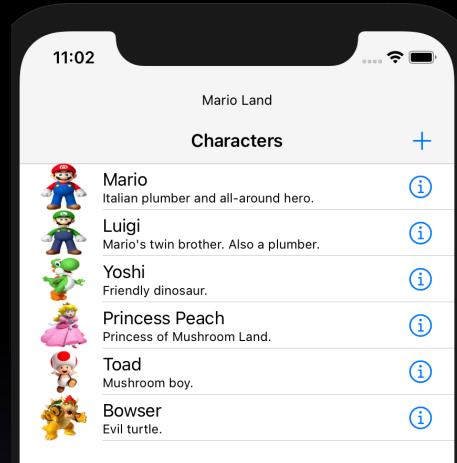
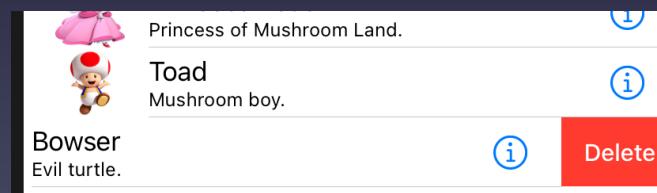


Table View Controller

The screenshot shows the Xcode interface with a storyboard project named 'TableDemo'. A red arrow points from the text '2. Create Cocoa Touch Class' to the 'TableViewController.swift' file in the project navigator. Another red arrow points from the text '1. Add Table View Controller to Storyboard.' to the storyboard canvas where a 'Table View' is added. A third red arrow points from the text '3. Assign new class to Table View Controller.' to the 'Identity' inspector, which shows the 'Class' dropdown set to 'TableViewController'.

2. Create Cocoa Touch Class inheriting from UITableViewController.

1. Add Table View Controller to Storyboard.

3. Assign new class to Table View Controller.

Table View
Prototype Content

TableViewController.swift

Custom Class
Class TableViewController

Table View Content: Static

- Multiple sections
- Fixed number of cells
- Design each cell individually

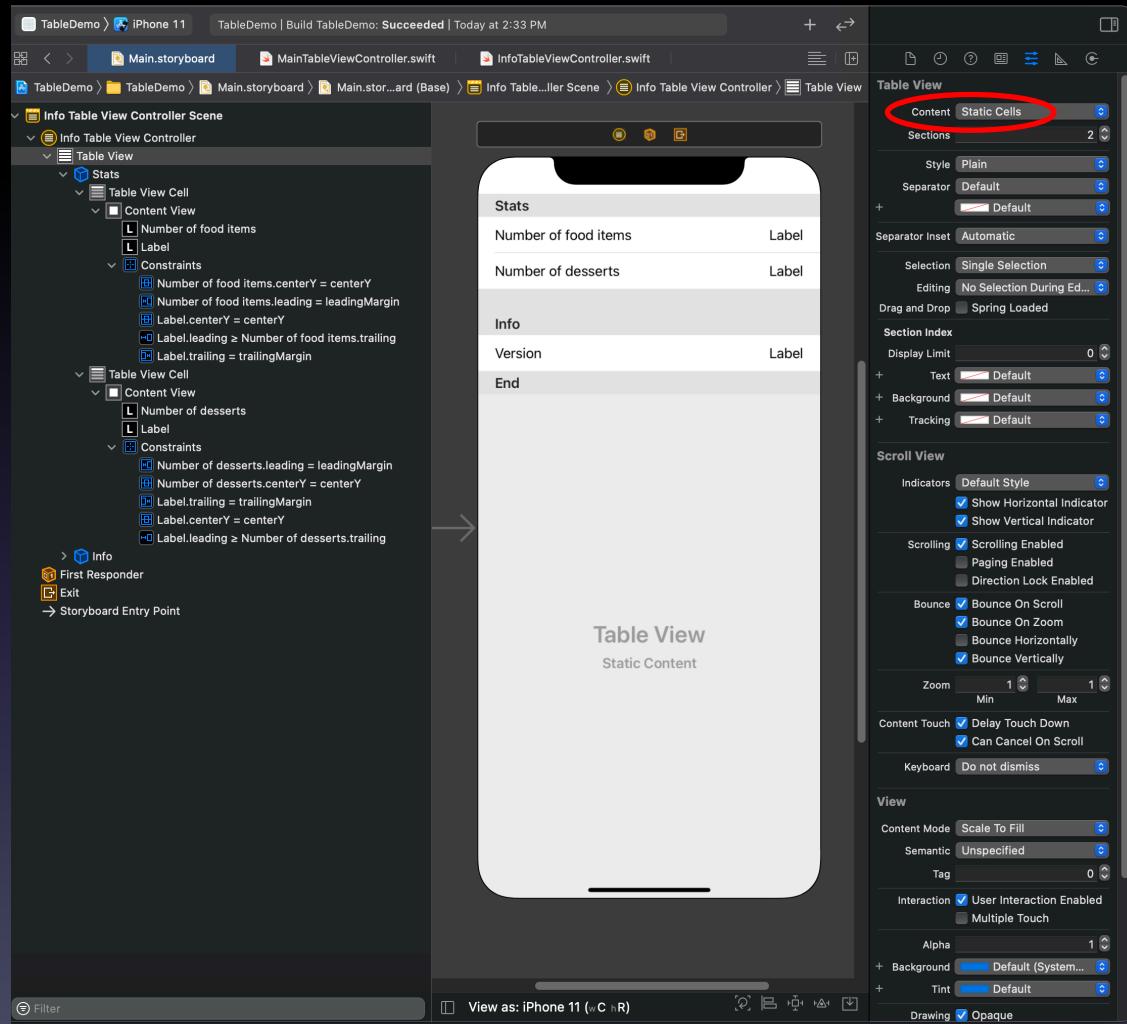


Table View Controller: Static

```
class InfoTableViewController: UITableViewController {

    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        // #warning Incomplete implementation, return the number of sections
        return 2
    }

    override func tableView(_ tableView: UITableView,
                          numberOfRowsInSection section: Int) -> Int {
        // #warning Incomplete implementation, return the number of rows
        switch section {
        case 0: return 2
        case 1: return 1
        default: return 0
        }
    }
}
```

Table View Content: Dynamic

- Variable number of cells
- Preset cell style or custom
- Must specify identifier for each cell prototype

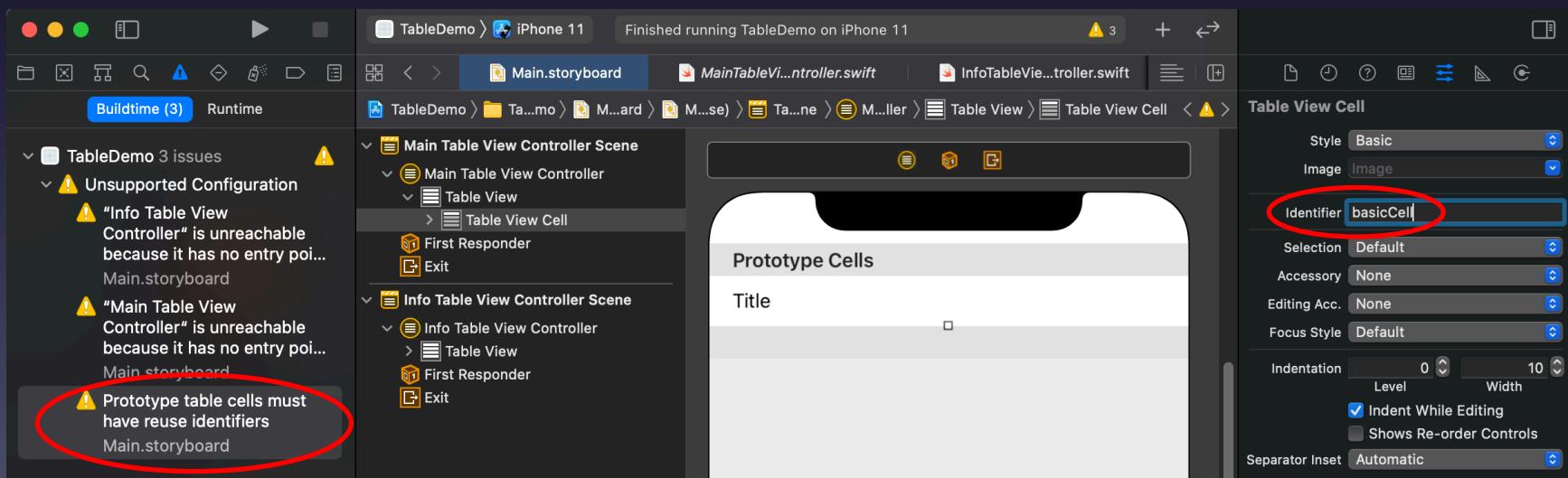
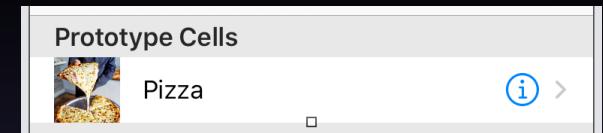
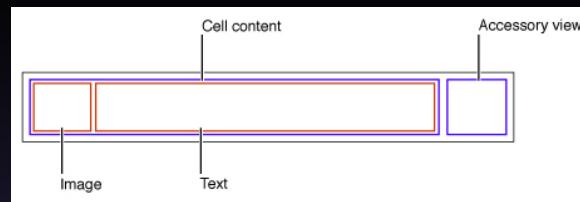


Table Cell Styles

- Table cell styles

- Basic

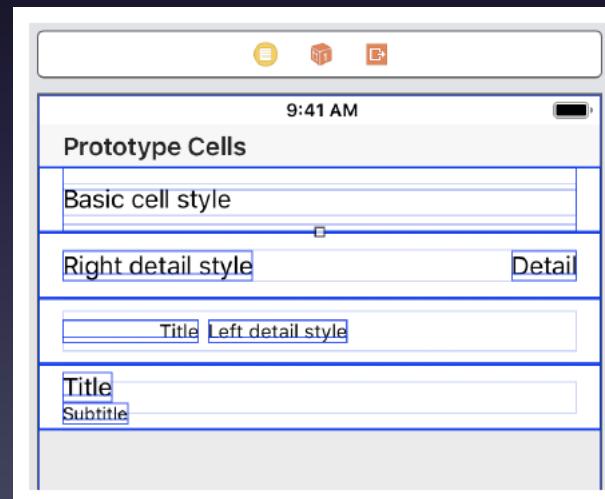


- Right detail

- Left detail

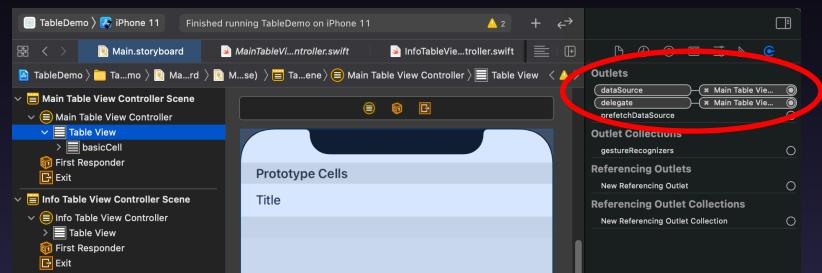
- Subtitle

- Custom



Delegate and Data Source

- Where does table send messages? Get data?
- UITableViewDelegate protocol
 - Configuring table
 - Handling table interaction
- UITableViewDataSource
 - Source for each row's data
 - Handling insertion/deletion



```
class MainTableViewController: UITableViewController {  
  
    override func viewDidLoad() {  
  
        self.tableView.delegate = self  
        self.tableView.dataSource = self  
    }  
}
```

Table View Controller: Dynamic

```
class MainTableViewController: UITableViewController {

    var foodItems: [FoodItem] = [ ]

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

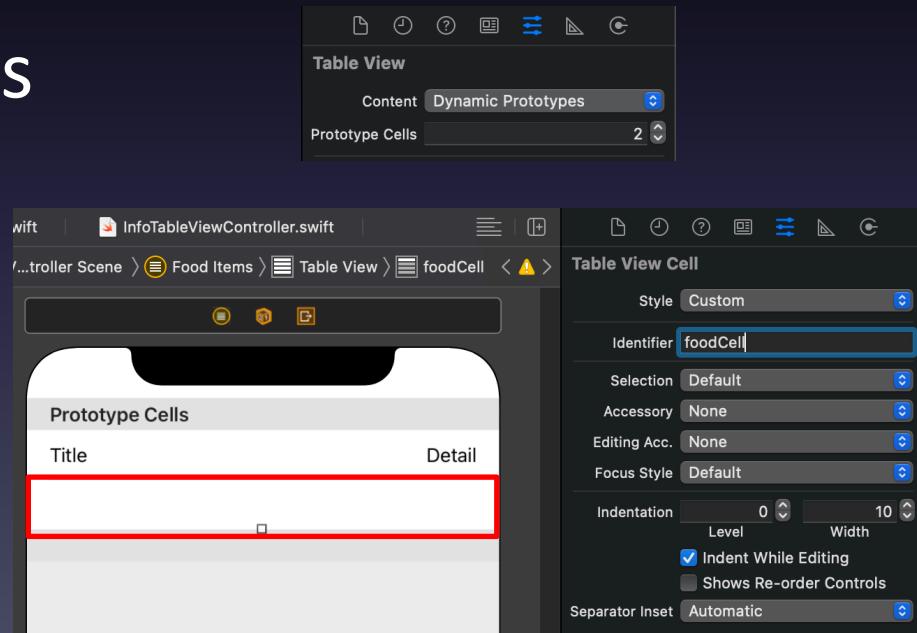
    override func tableView(_ tableView: UITableView,
                          numberOfRowsInSection section: Int) -> Int {
        return foodItems.count
    }

    override func tableView(_ tableView: UITableView,
                          cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier:
            "basicCell", for: indexPath)
        // Configure the cell...
        let foodItem = foodItems[indexPath.row]
        cell.textLabel?.text = foodItem.name
        return cell
    }
}
```

Custom Cells

- Create a custom table cell class
- Connect cell items in Storyboard to class

outlets and actions



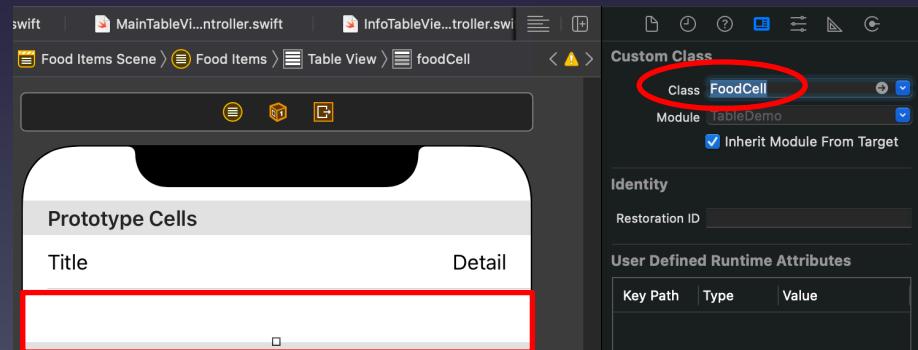
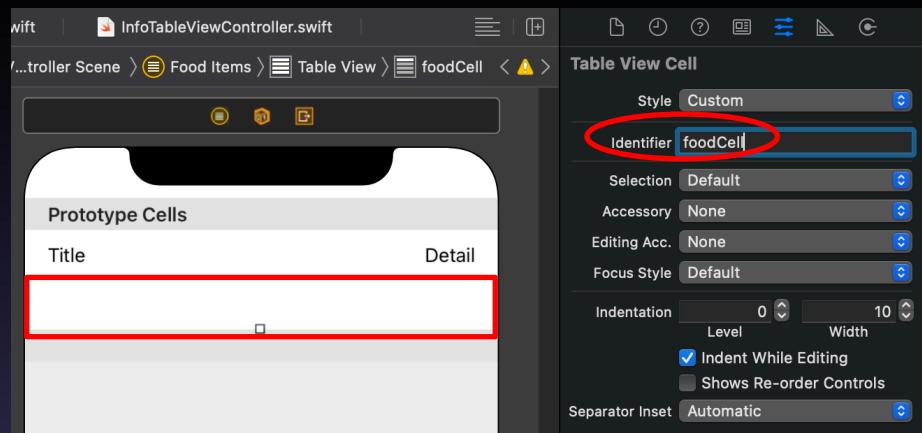
Custom Cells: Step 1

- Create custom class inheriting UITableViewCell
- Initialize and configure defaults as needed

```
class FoodCell: UITableViewCell {  
  
    override func awakeFromNib() {  
        super.awakeFromNib()  
        // Initialization code  
    }  
  
    override func setSelected(_ selected: Bool, animated: Bool) {  
        super.setSelected(selected, animated: animated)  
        // Configure the view for the selected state  
    }  
}
```

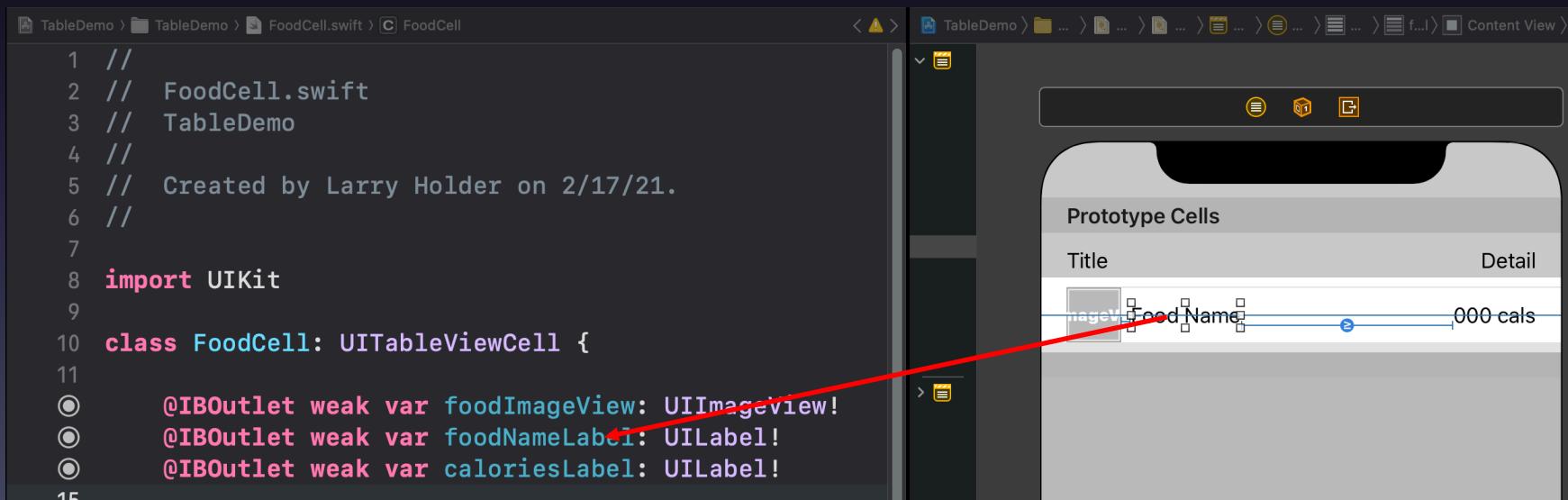
Custom Cells: Step 2

- Create custom table view prototype cell on Storyboard
- Set Identifier
- Set Custom Class
 - to class from Step 1



Custom Cells: Step 3

- Design cell (like view, just smaller)
- Connect cell elements to cell class



The screenshot shows the Xcode interface with two main panes. On the left is the code editor for `FoodCell.swift`, and on the right is the storyboard editor showing a prototype cell.

Code Editor (FoodCell.swift):

```
1 //  
2 // FoodCell.swift  
3 // TableDemo  
4 //  
5 // Created by Larry Holder on 2/17/21.  
6 //  
7  
8 import UIKit  
9  
10 class FoodCell: UITableViewCell {  
11  
12     @IBOutlet weak var foodImageView: UIImageView!  
13     @IBOutlet weak var foodNameLabel: UILabel!  
14     @IBOutlet weak var caloriesLabel: UILabel!  
15 }
```

Storyboard Editor:

The storyboard shows a prototype cell with three labels: "Title", "Food Name", and "Detail". A red line connects the `foodNameLabel` outlet in the code editor to the "Food Name" label in the storyboard. The storyboard also shows a placeholder value "000 cals" for the "Detail" label.

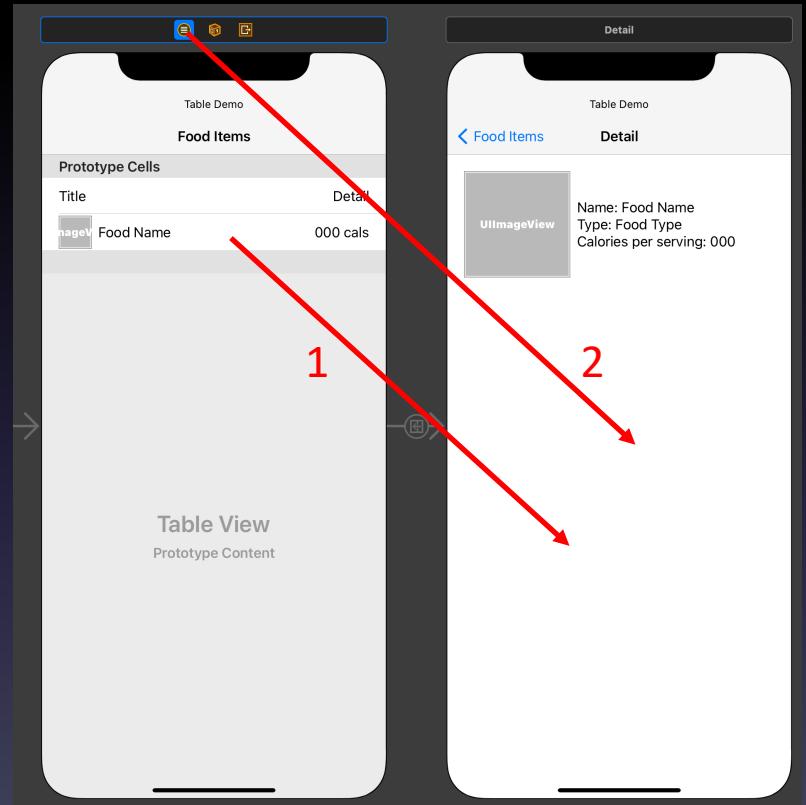
Custom Cells: Step 4

- Configure cells in UITableViewController

```
class MainTableViewController: UITableViewController {
// ...
    override func tableView(_ tableView: UITableView,
        cellForRowAt indexPath: IndexPath) -> UITableViewCell
    {
        let cell = tableView.dequeueReusableCell(withIdentifier:
            "foodCell", for: indexPath) as! FoodCell
        let foodItem = foodItems[indexPath.row]
        cell.foodNameLabel.text = foodItem.name
        cell.caloriesLabel.text = "\(foodItem.calories) cals"
        switch foodItem.type {
            case .appetizer:
                cell.foodImageView?.image = UIImage(named: "appetizer.png")
            case .dessert:
                cell.foodImageView?.image = UIImage(named: "dessert.png")
            case .entree:
                cell.foodImageView?.image = UIImage(named: "entree.png")
        }
        return cell
    }
}
```

Navigation

- Option 1: Create segue from prototype cell to destination view
 - Access `tableView.indexPathForSelectedRow` in `prepare for segue`
- Option 2: Create segue from view icon to destination view
 - Override `tableView:didSelectRowAtIndexPath` method
 - Process and store data from selected row
 - Perform segue programmatically
 - Pass stored data in `prepare for segue`



Cell Interaction: Selection

- Row Selection

```
override func tableView(_ tableView: UITableView,  
                      didSelectRowAt indexPath: IndexPath) {  
    let cell = tableView.cellForRow(at: indexPath) as! FoodCell  
    cell.foodNameLabel.textColor = UIColor.red  
}
```

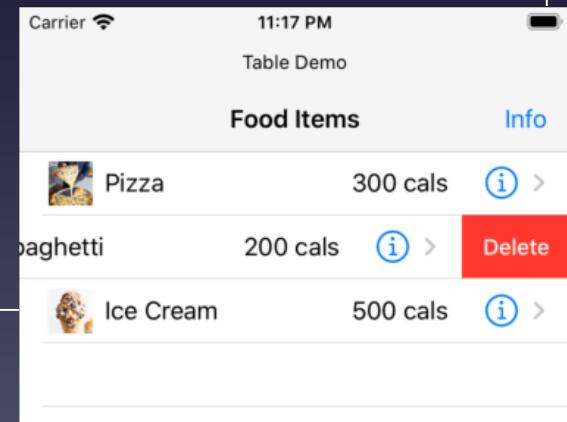
- Accessory Selection (only for Detail )

```
override func tableView(_ tableView: UITableView,  
                      accessoryButtonTappedForRowWith indexPath: IndexPath) {  
    performSegue(withIdentifier: "toDetailView",  
                sender: indexPath.row)  
}
```

Cell Interaction: Deletion

```
// Override to support conditional editing of the table view.
override func tableView(_ tableView: UITableView,
    canEditRowAt indexPath: IndexPath) -> Bool {
    // Return false if you do not want the specified item to be editable.
    return true
}

// Override to support editing the table view.
override func tableView(_ tableView: UITableView,
    commit editingStyle: UITableViewCellEditingStyle,
    forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        // Delete the row from the data source first
        foodItems.remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath],
            with: .fade)
    }
}
```

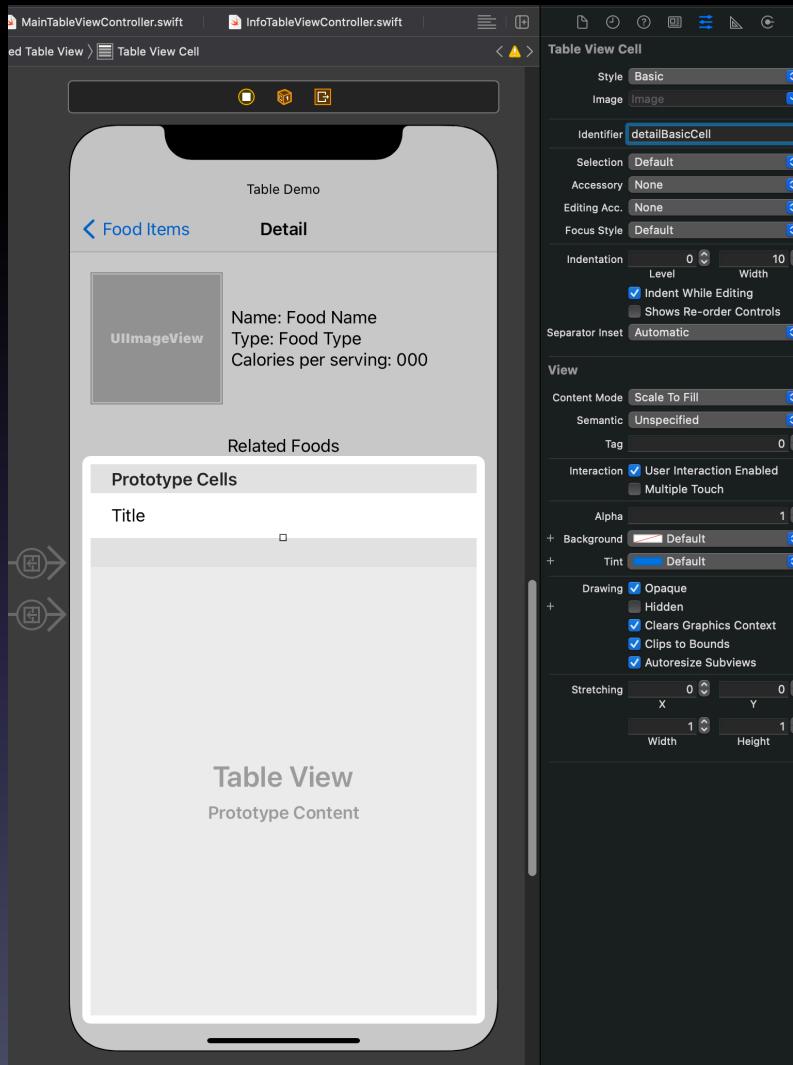


Insertion

- Add items to data source
- `tableView.reloadData()`

```
@IBAction func addTapped(_ sender: UIBarButtonItem) {  
    let foodItem = FoodItem(name: "New Food", type: .dessert, calories: 99)  
    foodItems.append(foodItem)  
    self.tableView.reloadData()  
}
```

Adding Table View to Existing View



Set Delegate and Data Source Programmatically

- Conform view controller to `UITableViewDelegate` and `UITableViewDataSource` protocols
- Create outlet connection to table view
- Set table view delegate and datasource to `self`

```
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
    @IBOutlet weak var relatedTableView: UITableView!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        relatedTableView.delegate = self
        relatedTableView.dataSource = self
    }
}
```

Resources

- Start Developing iOS Apps
 - developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift
 - Archived, but still good tutorial on Tables
- Table Views (documentation)
 - developer.apple.com/documentation/uikit/views_and_controls/table_views