Tables

Mobile Application Development in iOS

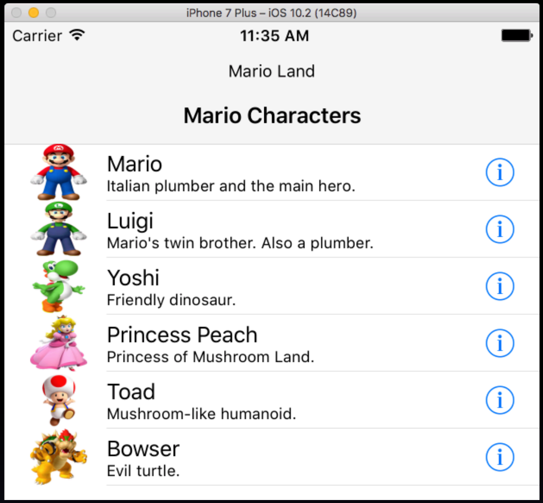
School of EECS

Washington State University

Instructor: Larry Holder

Outline

- Table View Controller
- Table View
- Table Cells
- Cell interaction
- Navigation



iPhone 7 Plus – iOS 10.2 (14C89)

Carrier 11:35 AM

Mario Land

Mario Characters

Mario
Italian plumber and the main hero.

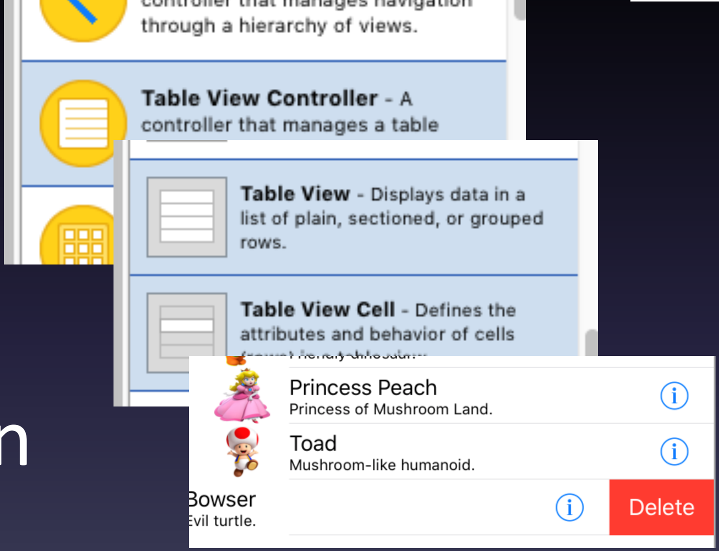
Luigi
Mario's twin brother. Also a plumber.

Yoshi
Friendly dinosaur.

Princess Peach
Princess of Mushroom Land.

Toad
Mushroom-like humanoid.

Bowser
Evil turtle.



controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table

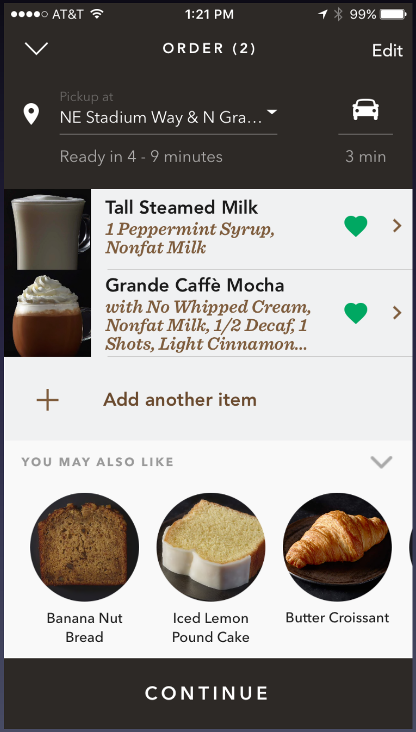
Table View - Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell - Defines the attributes and behavior of cells

Princess Peach
Princess of Mushroom Land.

Toad
Mushroom-like humanoid.

Bowser
Evil turtle.



●●●●○ AT&T 1:21 PM 99%

ORDER (2)

Pickup at
NE Stadium Way & N Gra…

Ready in 4 - 9 minutes

3 min

Tall Steamed Milk
1 Peppermint Syrup, Nonfat Milk

Grande Caffè Mocha
with No Whipped Cream, Nonfat Milk, 1/2 Decaf, 1 Shots, Light Cinnamon...

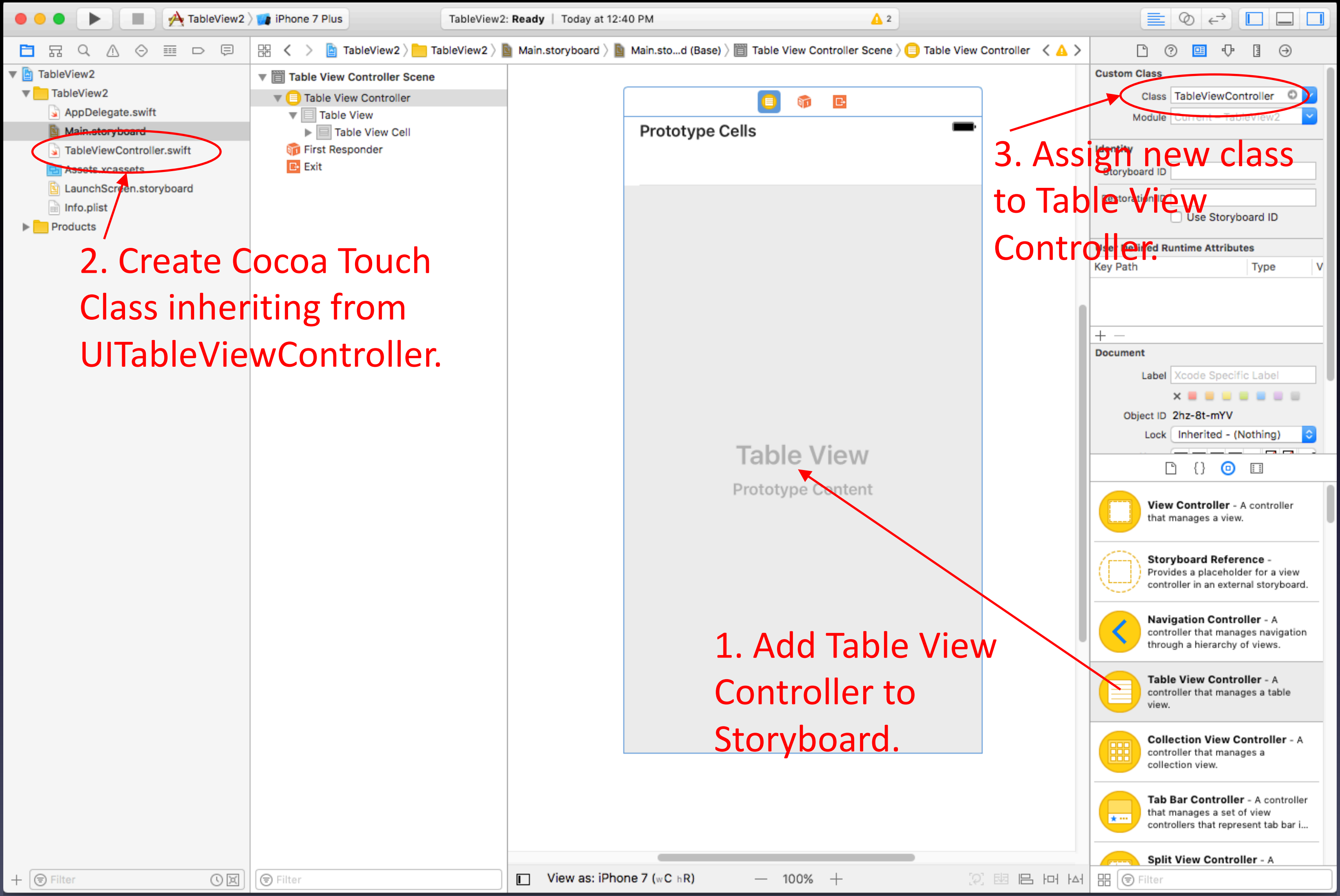
YOU MAY ALSO LIKE

Banana Nut Bread

Iced Lemon Pound Cake

Butter Croissant

Table View Controller



1. Add Table View Controller to Storyboard.

2. Create Cocoa Touch Class inheriting from UITableViewController.

3. Assign new class to Table View Controller.

Custom Class
Class: TableViewController
Module: Current – TableView2

TableView2
iPhone 7 Plus
TableView2: Ready | Today at 12:40 PM
TableView2
TableView2
AppDelegate.swift
Main.storyboard
TableViewController.swift
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Main.storyboard
Main.sto...d (Base)
Table View Controller Scene
Table View Controller
Table View
Table View Cell
First Responder
Exit
Prototype Cells
Table View
Prototype Content
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Document
Label
Xcode Specific Label
Object ID 2hz-8t-mYV
Lock
Inherited - (Nothing)
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar i...
Split View Controller - A
Filter
View as: iPhone 7 (wC hR)
100%

Table View Attributes

- TableView Content
  - Dynamic
    - One section
    - Multiple cell prototypes
    - Variable number of cells
  - Static
    - Multiple sections
    - One cell prototype per section
    - Fixed number of cells
  - Sections Plain or Grouped

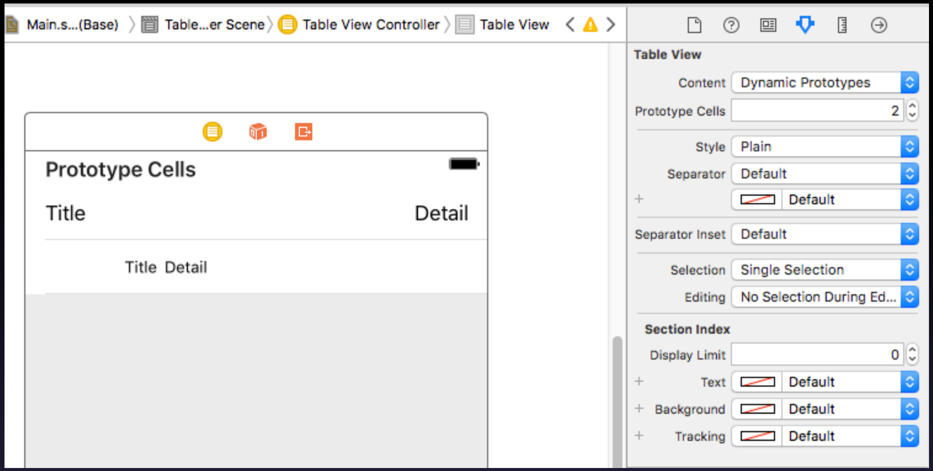


Table View
Content: Dynamic Prototypes
Prototype Cells: 2
Style: Plain
Separator: Default
Default
Separator Inset: Default
Selection: Single Selection
Editing: No Selection During Ed...
Section Index
Display Limit: 0
Text: Default
Background: Default
Tracking: Default

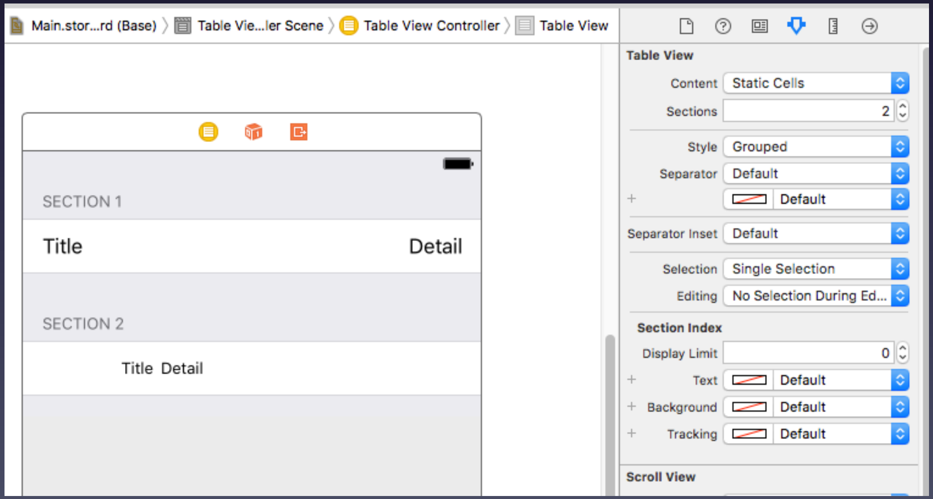
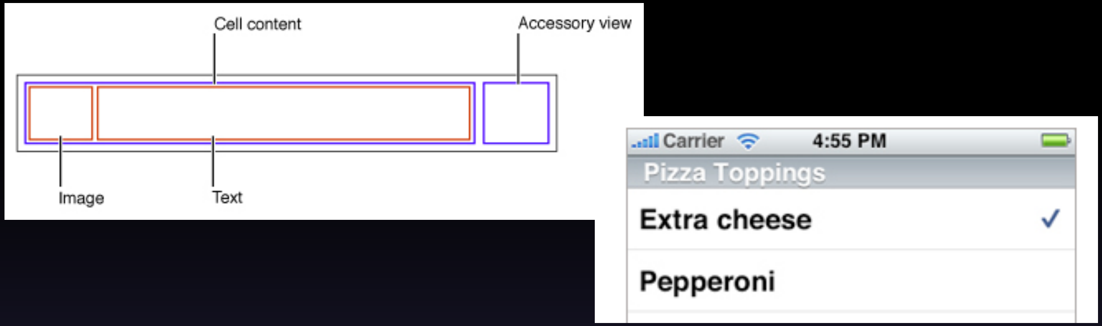
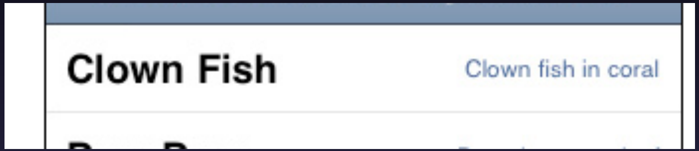


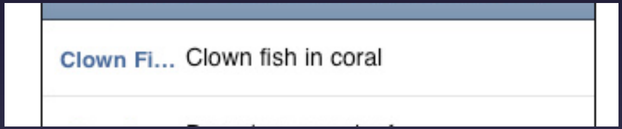
Table View
Content: Static Cells
Sections: 2
Style: Grouped
Separator: Default
Default
Separator Inset: Default
Selection: Single Selection
Editing: No Selection During Ed...
Section Index
Display Limit: 0
Text: Default
Background: Default
Tracking: Default
Scroll View

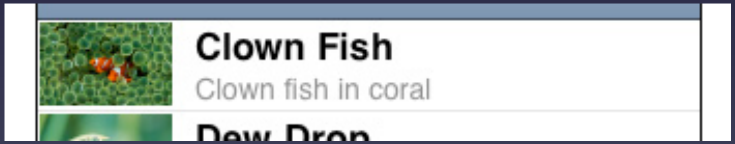
Table Cell Styles

- Table cell styles
  - Basic
  - Right detail
  - Left detail
  - Subtitle
  - Custom



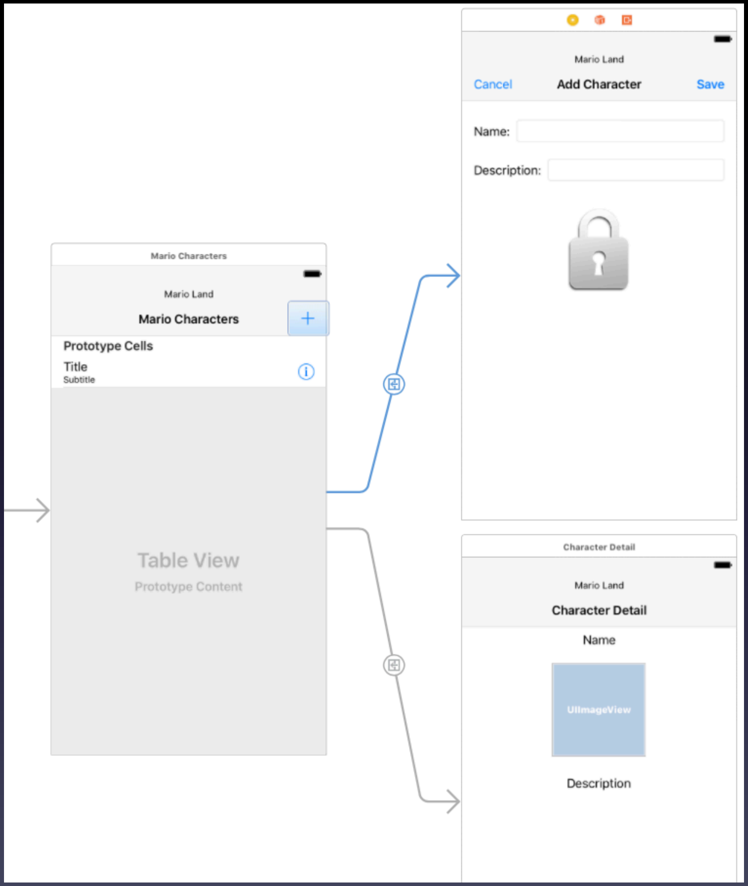






Navigation

- Create views for Details and Add new entries
- Create segue to Detail View
  - Perform when row/accessory selected
- Create Add bar button
- Create segue from Add button to Add View



Cell Interaction: Selection

- Row Selection

```
override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
  self.selectedRow = indexPath.row
  performSegue(withIdentifier: "toDetail", sender: nil)
}
```

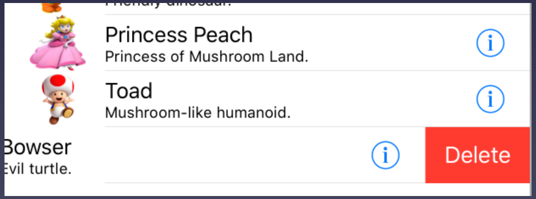
- Accessory Selection

```
override func tableView(_ tableView: UITableView,
                        accessoryButtonTappedForRowWith indexPath: IndexPath) {
  self.selectedRow = indexPath.row
  performSegue(withIdentifier: "toDetail", sender: nil)
}
```

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
    marioCharacters.remove(at: indexPath.row)
    tableView.deleteRows(at: [indexPath], with: .fade)
  }
}
```



Friendly dinosaur.

Princess Peach
Princess of Mushroom Land.

Toad
Mushroom-like humanoid.

Bowser
Evil turtle.

Insertion

- In AddViewController.swift
  - Maintain Bool indicating new entry ready
  - Save button sets Bool=true and performs unwind segue

```
class AddViewController: UIViewController, UITextFieldDelegate {

  var newCharacterReady: Bool = false

  @IBOutlet weak var nameTextField: UITextField!
  @IBOutlet weak var descriptionTextField: UITextField!

  @IBAction func saveButton(_ sender: UIBarButtonItem) {
    newCharacterReady = true
    performSegue(withIdentifier: "unwindFromDetail", sender: nil)
  }
// ...
```

Insertion (cont.)

- In TableViewController.swift
  - – In unwind segue
    - Check if new entry ready
    - If so, create new data instance, add to array, and reload data

```
@IBAction func unwindFromDetail (segue: UIStoryboardSegue) {
  let addVC = segue.source as! AddViewController
  if (addVC.newCharacterReady) {
    let name = addVC.nameTextField.text!
    let description = addVC.descriptionTextField.text!
    let newCharacter = MarioCharacter(name, description, "locked-128.png")
    marioCharacters.append(newCharacter)
    self.tableView.reloadData()
  }
}
```

Adding Table View to Existing View

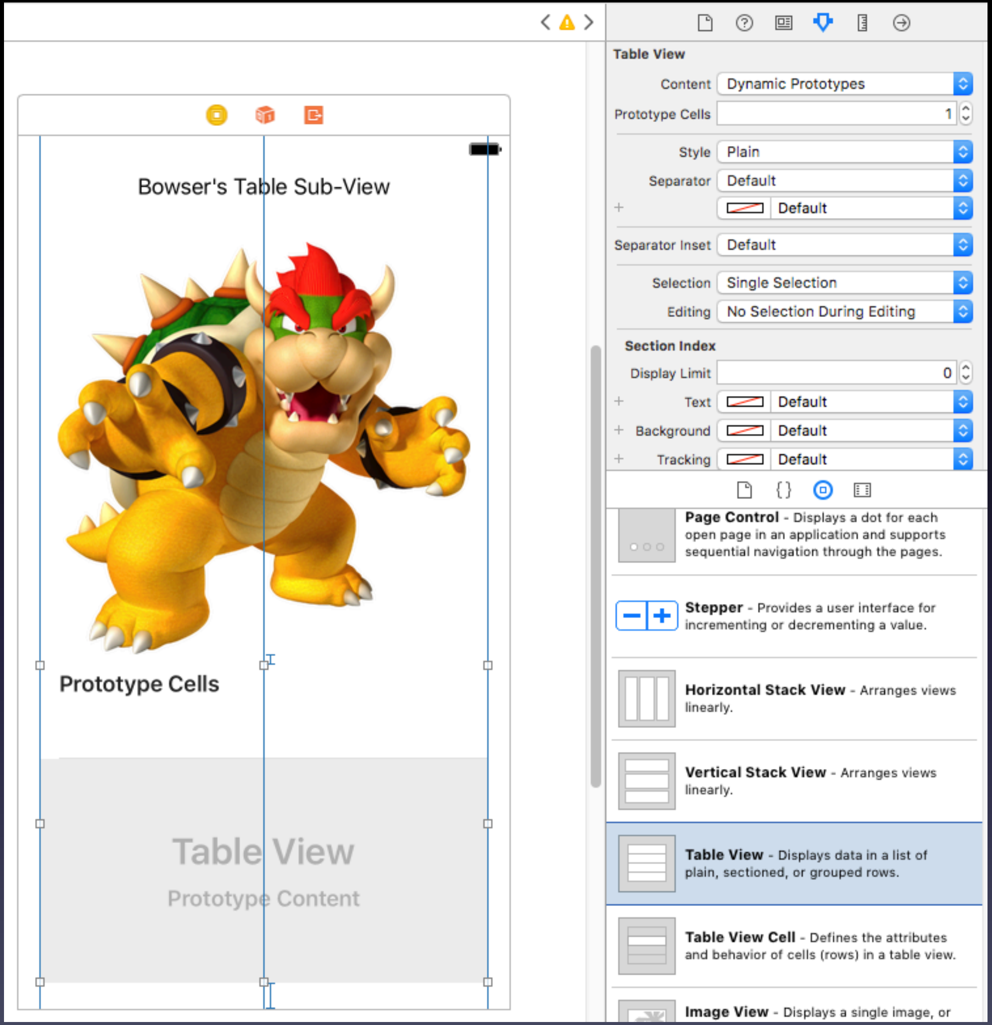


Table View

Content	Dynamic Prototypes
Prototype Cells	1
Style	Plain
Separator	Default
	Default
Separator Inset	Default
Selection	Single Selection
Editing	No Selection During Editing

Section Index

Display Limit	0
Text	Default
Background	Default
Tracking	Default

Page Control - Displays a dot for each open page in an application and supports sequential navigation through the pages.

Stepper - Provides a user interface for incrementing or decrementing a value.

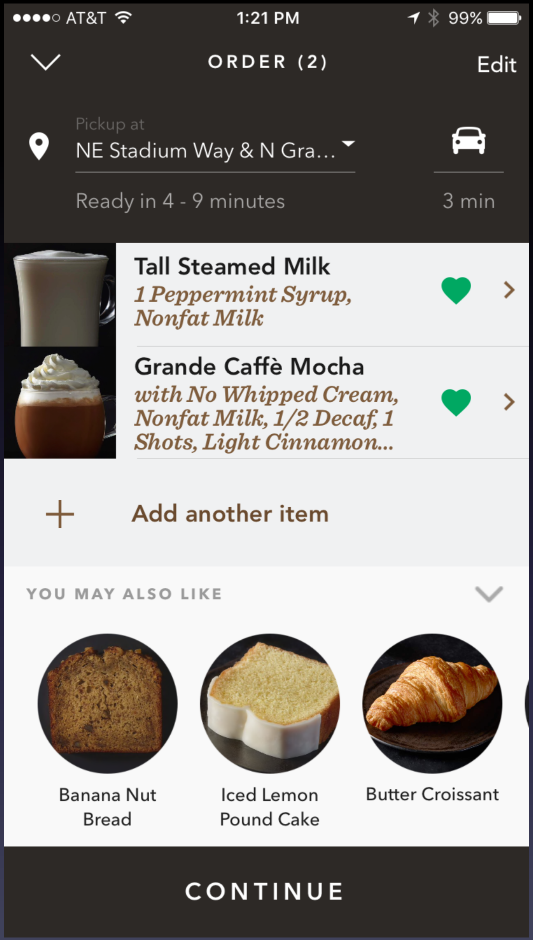
Horizontal Stack View - Arranges views linearly.

Vertical Stack View - Arranges views linearly.

Table View - Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell - Defines the attributes and behavior of cells (rows) in a table view.

Image View - Displays a single image, or



Delegate and Data Source

- Automatic for Table View Controller

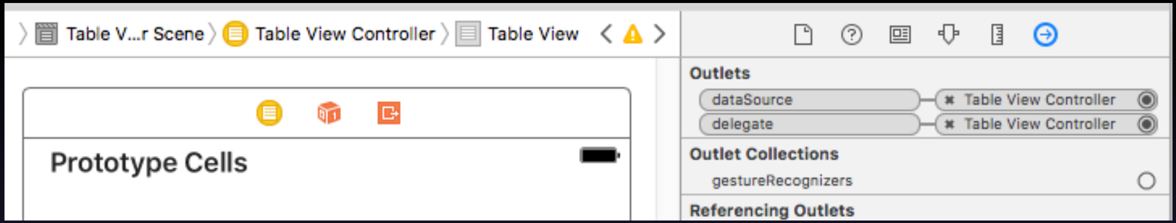


Table V...r Scene 〉 Table View Controller 〉 Table View

Prototype Cells

Outlets

dataSource — Table View Controller

delegate — Table View Controller

Outlet Collections

gestureRecognizers ○

Referencing Outlets

- But can setup programmatically for Table sub-View

```
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

    @IBOutlet weak var bowserTableView: UITableView!

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

Delegate and DataSource for Table View

```
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

  // ...

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

  func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return 3
  }

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

    // Configure the cell...
    cell.textLabel?.text = "Bowser"

    return cell
  }
```

Resources

- Start Developing iOS Apps (good Tables tutorial)
  - https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/
- UITableViewController (documentation)
  - https://developer.apple.com/reference/uikit/uitableviewcontroller