

Table View Controllers-05

In this assignment you will create an app called Homeowner that keeps an inventory of all your possessions. In the case of a fire or other catastrophe, you will have a record for your insurance company. This app will use a UITableViewController.

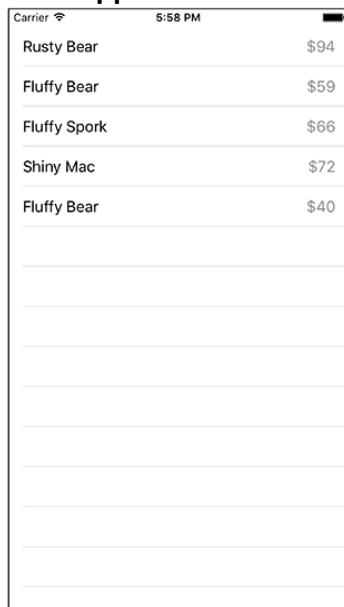
1. Create a new iOS app in Xcode named Homeowner. Create a repository on GitHub and connect your Xcode project to your GitHub repository.
2. Create a new Swift file named ItemsViewController. In ItemsViewController.swift, define a class named ItemsViewController that is a subclass of UITableViewController.
3. Open Main.storyboard. You want the initial view controller to be a table view controller. Select the existing View Controller on the canvas and press Delete. Then drag a Table View Controller from the object library onto the canvas.
4. With the Table View Controller selected, open its identity inspector and change the class to ItemsViewController.
5. Open the attributes inspector for Items View Controller and check the box for Is Initial View Controller.
6. You no longer need the ViewController.swift file that was created for you. Select this file in the project navigator and press Delete.
7. Create a new Swift file named Item that is a subclass of NSObject. In Item.swift, define the Item class and give it four properties:
 - a) name (String)
 - b) valueInDollars (Int)
 - c) serialNumber (String?)
 - d) dateCreated (Date)
8. In your Item.swift file add an initializer that takes 4 parameters (name, valueInDollars, serialNumber, and dateCreated) and sets the Item class variables' values to the values of these 4 parameters (be sure to use "super." if you named the parameter variables the same as the class variables and also call the super classes init method).
9. Also, in your Item.swift file, add a convenience initializer that will create a blank Item.
10. Create a new Swift file named ItemStore. In ItemStore.swift do the following:
 - a) Define a class called ItemStore
 - b) Declare a property to store the list of Items (array)
 - c) Create a @discardableResult function named createItem() that creates a new Item, adds the new item to the list of items, then returns the item
 - d) Create an initializer that adds 5 Items to the Items array

11. Open ItemsViewController and create a property for an ItemStore.
12. Open AppDelegate.swift, put the following code in the didFinishLaunchingWithOptions method:

```
// Create an ItemStore
let itemStore = ItemStore()

// Access the ItemsViewController and set its item store
let itemsController = window!.rootViewController as! ItemsViewController
itemsController.itemStore = itemStore
```
13. Tell your class you are going to conform to the UITableViewDataSource and UITableViewDelegate protocols
14. Implement the tableView(_:numberOfRowsInSection:) method from the UITableViewDataSource protocol in your ItemsViewController to return the count of your Items array
15. Open Main.storyboard and select your Table View. Add a prototype cell to the table view and set a cell identifier for the prototype cell.
16. Implement the tableView(_:cellForRowAt:) method from the UITableViewDataSource protocol in your ItemsViewController to return a cell with the data from the Item at the row given in the IndexPath variable. Set the cell's titleLabel to the name of the Item and set the cell's detailTextLabel to the valueInDollars of the Item.

Final App



Carrier	5:58 PM
Rusty Bear	\$94
Fluffy Bear	\$59
Fluffy Spork	\$66
Shiny Mac	\$72
Fluffy Bear	\$40