

Table View Navigation Controller

Brent Nycum
brentnycum@gmail.com

Create Our Project

- Choose a Navigational Based Project
- I will be using the name “TableApp”
- Make sure **not** to select “Use Core Data”

Choose a template for your new project:

iOS

Application

Framework & Library

Other

Mac OS X

Application

Framework & Library

Application Plug-in

System Plug-in

Other



Navigation-based
Application



OpenGL ES
Application



Split View-based
Application



Tab Bar Application



Utility Application



View-based
Application



Window-based
Application



Navigation-based Application

This template provides a starting point for an application that uses a navigation controller. It provides a user interface configured with a navigation controller to display a list of items.

Cancel

Previous

Next

Table View

- Need to tell our App how many sections our table view has and how many rows in each section
- Open up RootViewController.m

RootViewController

```
- (NSInteger)numberOfSectionsInTableView:  
  (UITableView *)tableView {  
  
    return 2;  
  
}
```

RootViewController

```
- (NSInteger)tableView:(UITableView *)tableView  
numberOfRowsInSection:(NSInteger)section {
```

```
    if (section == 0) {  
        return 4;  
    } else {  
        return 2;  
    }  
}
```

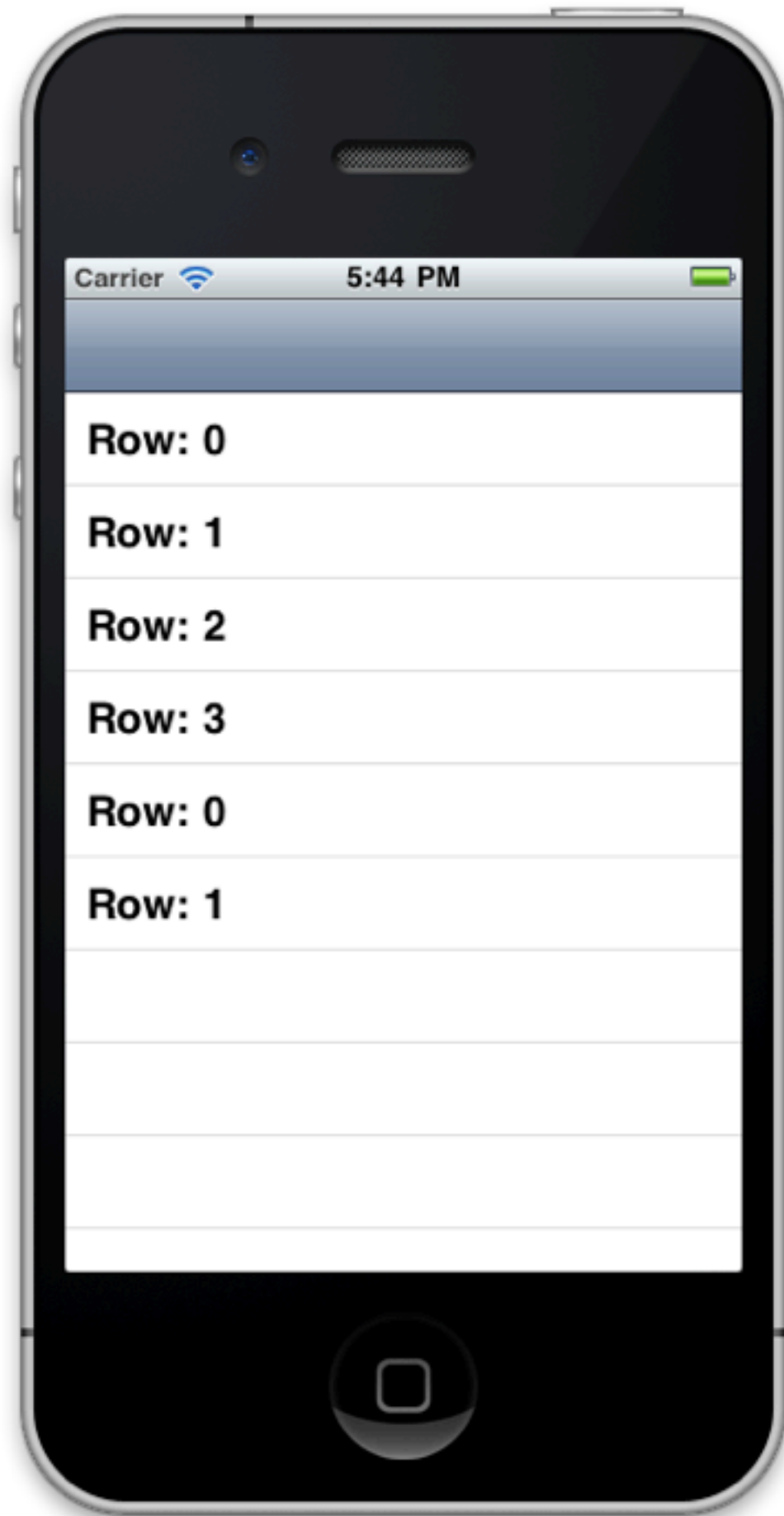
```
}
```

Configure the Cells

- Lets give our rows a title
- – `(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath`
- Under the `// Configure the Cell.` comment

Configuring the Cell

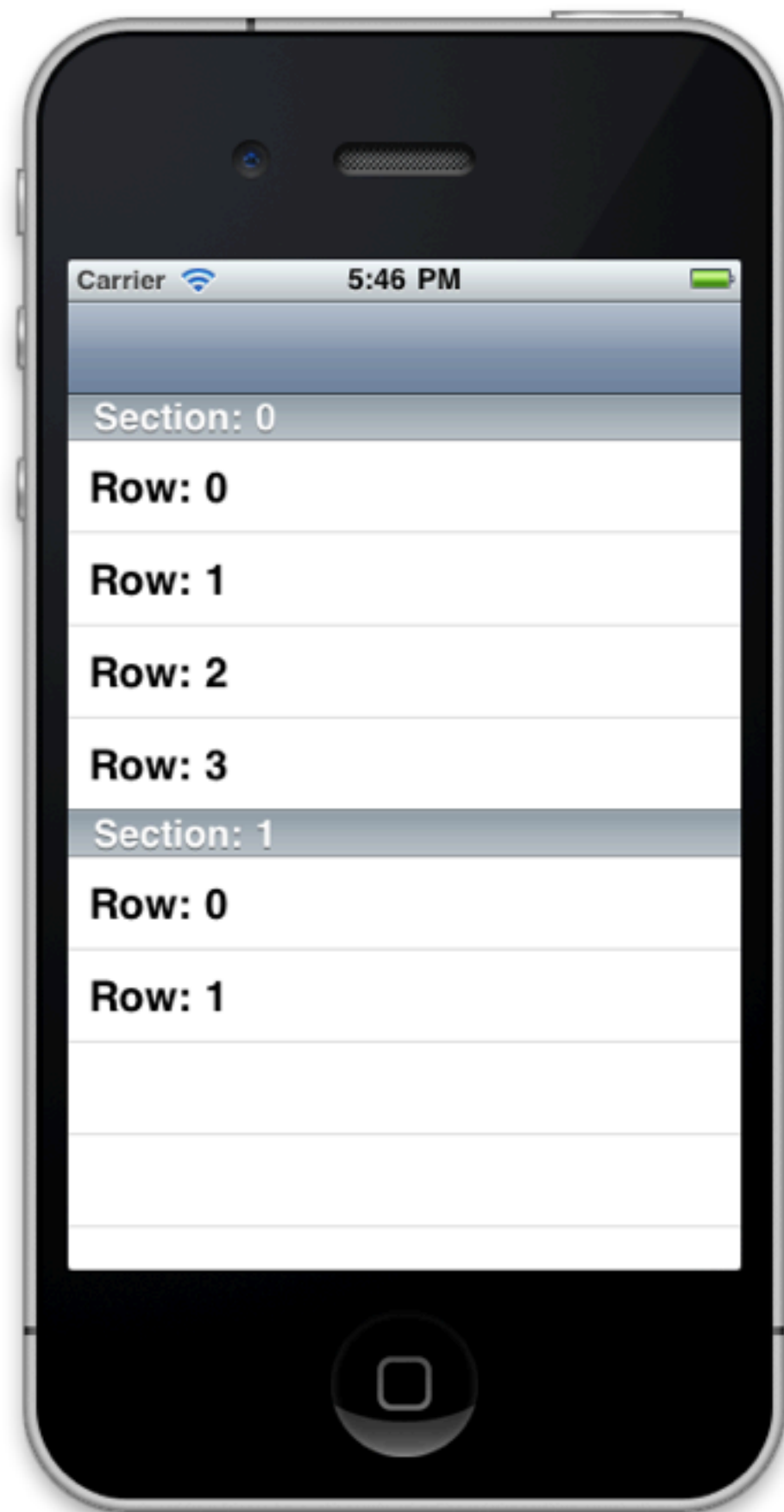
- `cell.textLabel.text = [NSString
stringWithFormat:@"Row: %d", indexPath.row];`



Lets Give Our Sections a Title

- Add this Method to our RootViewController interface

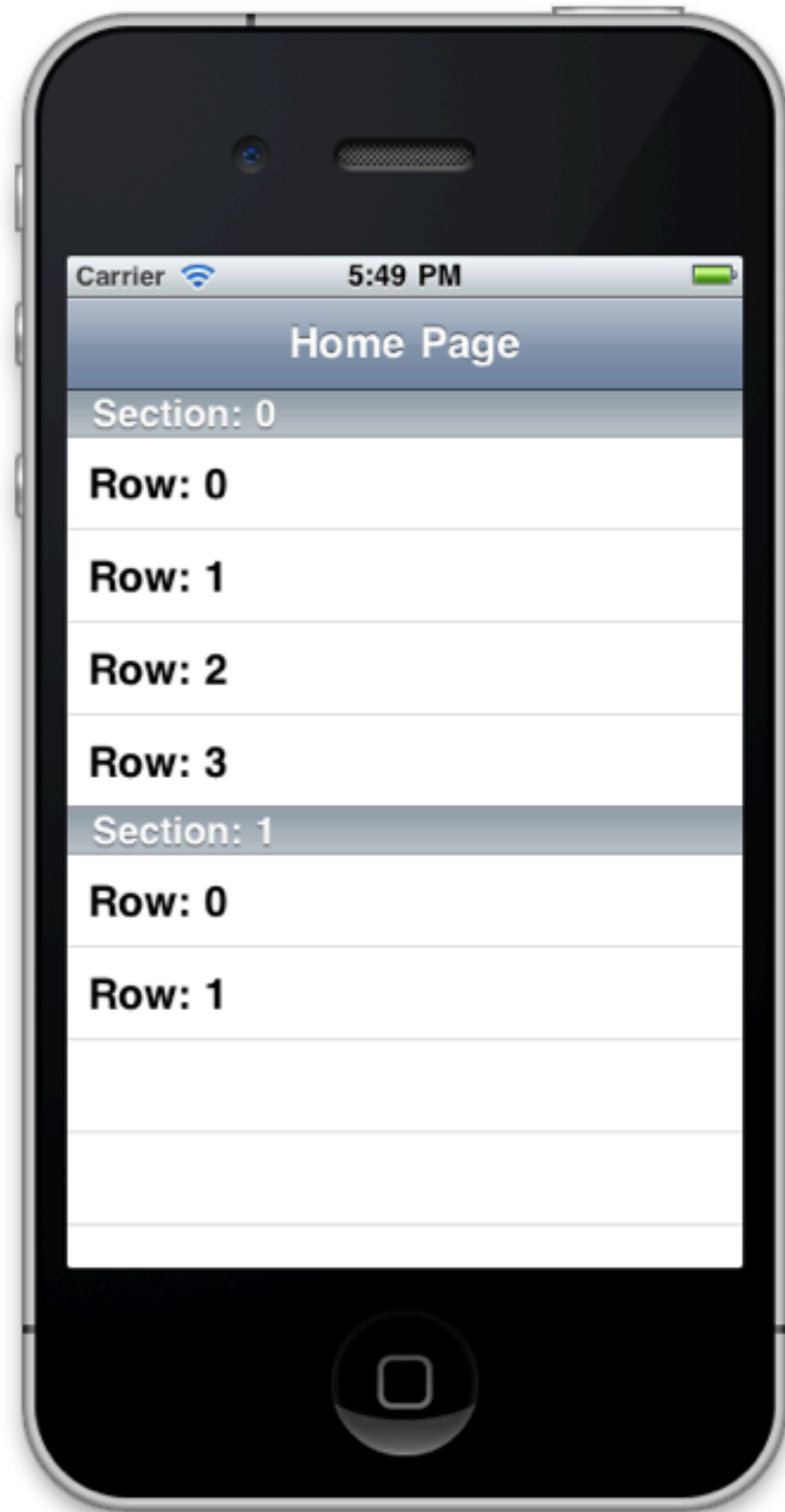
```
-(NSString *)tableView:(UITableView *)tableView  
titleForHeaderInSection:(NSInteger)section {  
  
    return [NSString stringWithFormat:@"Section:  
%d", section];  
  
}
```



Give Our Page a Title

- Let's override the `viewDidLoad` method and set our title

```
-(void)viewDidLoad {  
    [super viewDidLoad];  
  
    self.title = @"Home Page";  
}
```

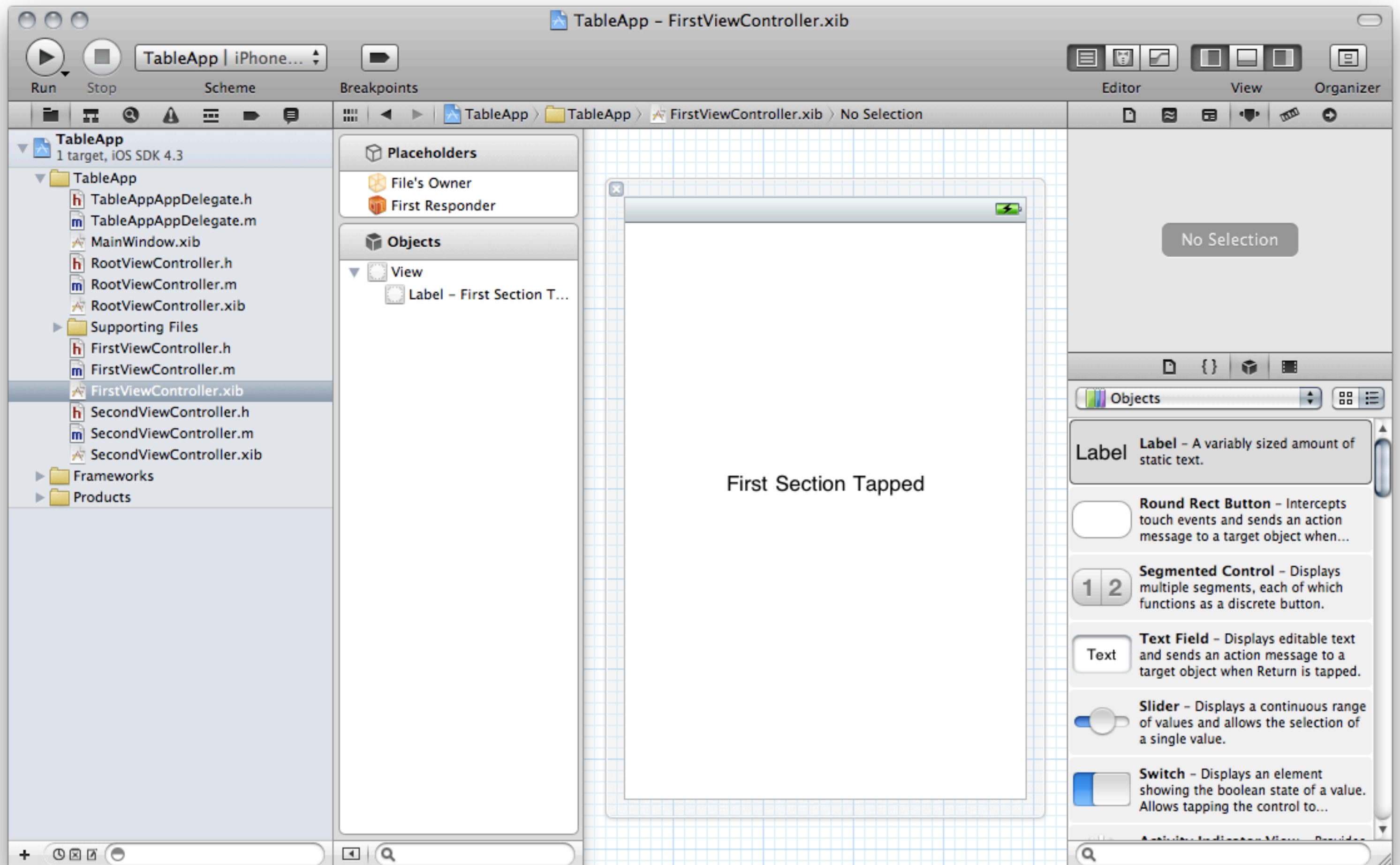


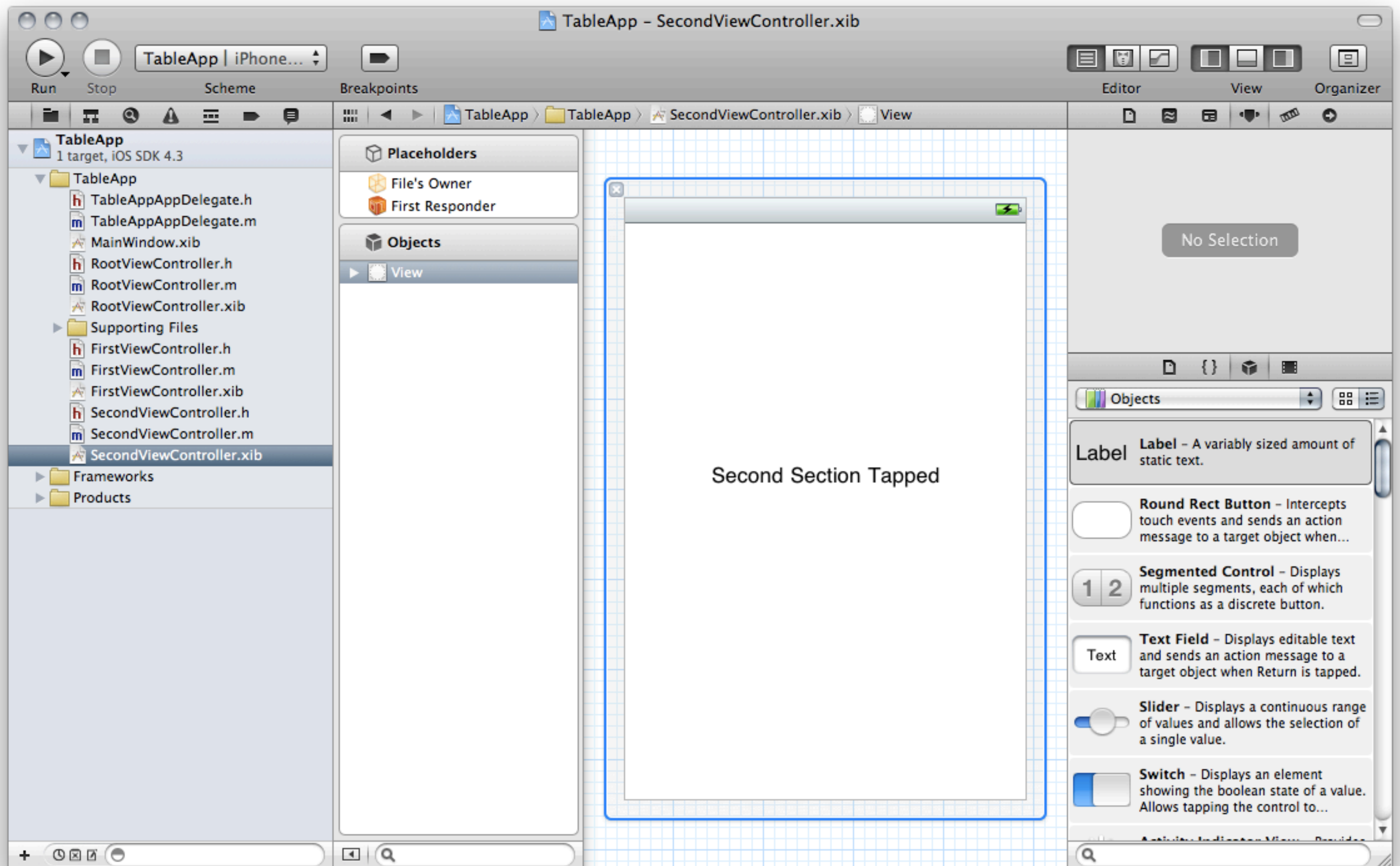
Navigation

- Let's create pages to navigate to
- To keep this short we will create 2 pages, one for each section

Navigation

- Create our 2 new View Controllers
 - File > New > New File
 - Select UIViewController (Subclass of UIViewController)
 - FirstViewController
 - SecondViewController





Navigation

- Let's setup our Navigation code
 - Open back up RootViewController
 - Add imports at top of file
 - `#import "FirstViewController.h"`
 - `#import "SecondViewController.h"`
 - Look at
 - `– (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath`

```
- (void)tableView:(UITableView *)tableView  
didSelectRowAtIndexPath:(NSIndexPath *)indexPath {  
    if (indexPath.section == 0) {  
    } else if (indexPath.section == 1) {  
    }  
}
```

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {  
  
    if (indexPath.section == 0) {  
  
        FirstViewController *first = [[FirstViewController alloc]  
initWithNibName:@"FirstViewController" bundle:nil];  
  
        [self.navigationController pushViewController:first animated:YES];  
  
        [first release];  
  
    } else if (indexPath.section == 1) {  
  
    }  
  
}
```

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {  
  
    if (indexPath.section == 0) {  
  
        FirstViewController *first = [[FirstViewController alloc]  
initWithNibName:@"FirstViewController" bundle:nil];  
  
        [self.navigationController pushViewController:first animated:YES];  
  
        [first release];  
  
    } else if (indexPath.section == 1) {  
        SecondViewController *second = [[SecondViewController alloc]  
initWithNibName:@"SecondViewController" bundle:nil];  
  
        [self.navigationController pushViewController:second animated:YES];  
  
        [second release];  
    }  
  
}
```

Niceties

- Add an accessory view to our Cells
- Lets configure our cell again
 - `cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;`

