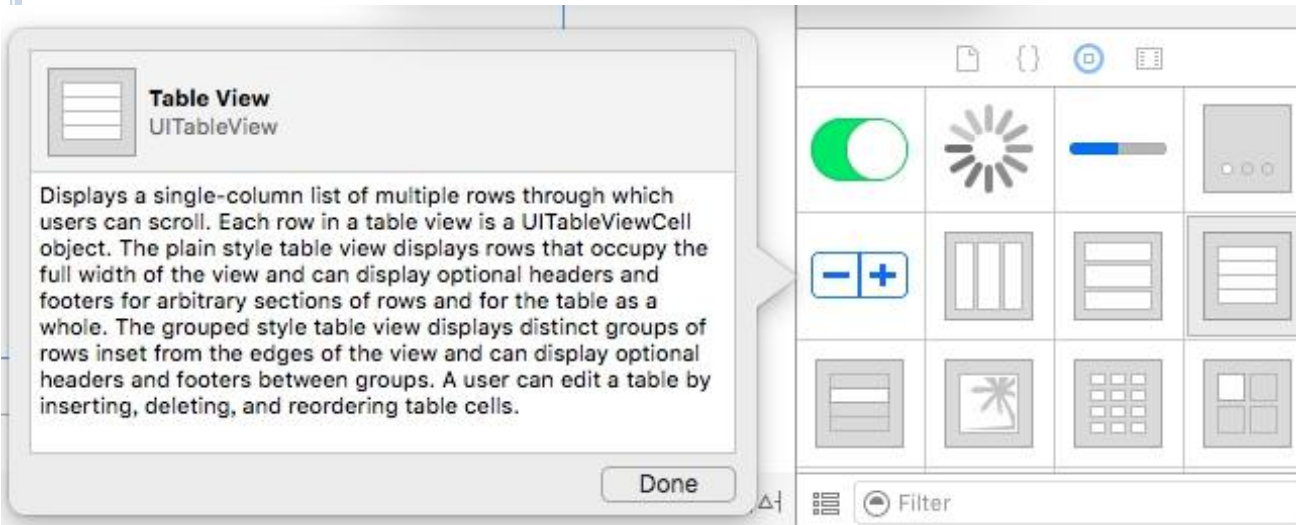
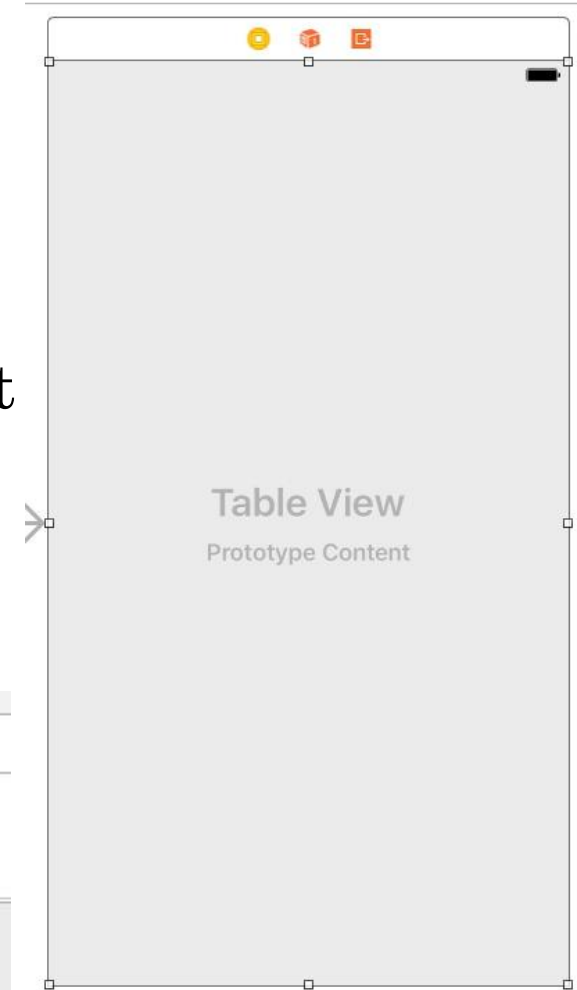


UITABLEVIEW APP

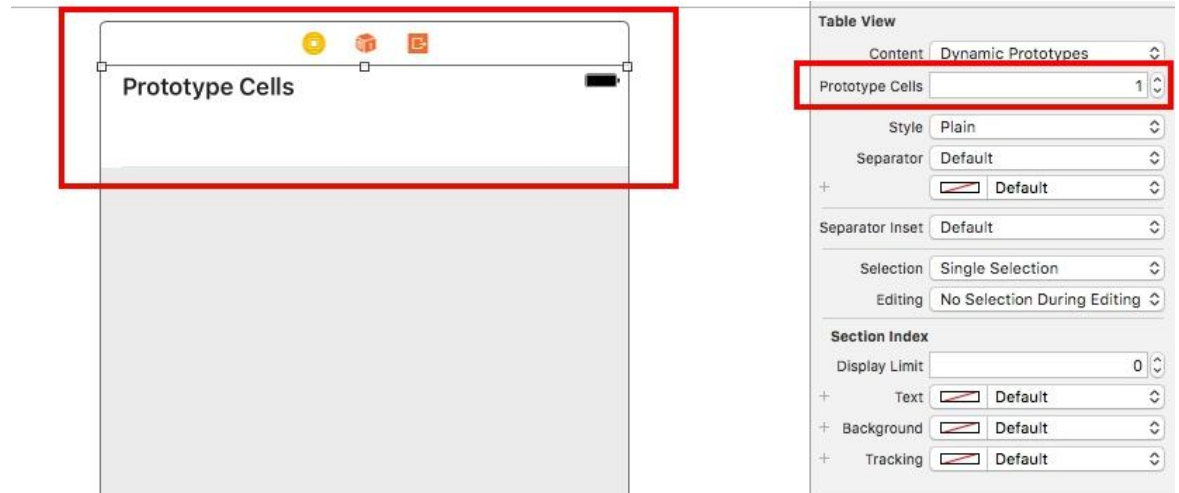
CONSTRUCT TABLEVIEW

- Select the TableView to be your layout



CHANGE TABLEVIEW'S PROPERTY

- Change Prototype Cell from 0 to 1
- You will see “Prototype Cell” shown on your layout



- If you didn't see that in the Xcode, open it in the way :
View > Utilities > Show Attributes Inspector



UITABLEVIEW AND PROTOCOLS

- Add “UITableViewDataSource” and “UITableViewDelegate” after “UIViewController” in your program to be protocol
- UITableViewDataSource & UITableViewDelegate are both protocol in Swift. To show the data in the table view


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

Type 'ViewController' does not conform to protocol 'UITableViewDataSource'



ERROR WARNING

- You would see the warning
“ViewController does not conform to protocol UITableViewDataSource”
- You must to tell UITableView what data you want to be shown.
- UITableViewDataSource is the key which connect your data and table view.
- To achieve that connect, you need to use

 tableView(_ tableView: numberOfRowsInSection)
tableView(_ tableView: cellForRowAtIndexPath)

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {}  
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->  
    UITableViewCell{}
```

USE ARRAY TO STORE DATA

- The data(label) you want to be shown on the table view, you can use array to store.

```
var items: [String] = ["#1", "#2", "#3", "#4", "#5", "#6", "#7", "#8"]
```



NUMBEROFROWINSECTION

- numberOfRowsInSection:

To show how many rows in each area.

You can use “count” to get rows in the array items(your array name)

```
//Set the number of rows
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return self.items.count;
}
```



CELLFORROWATINDEXPATH

- cellForRowAtIndexPath:

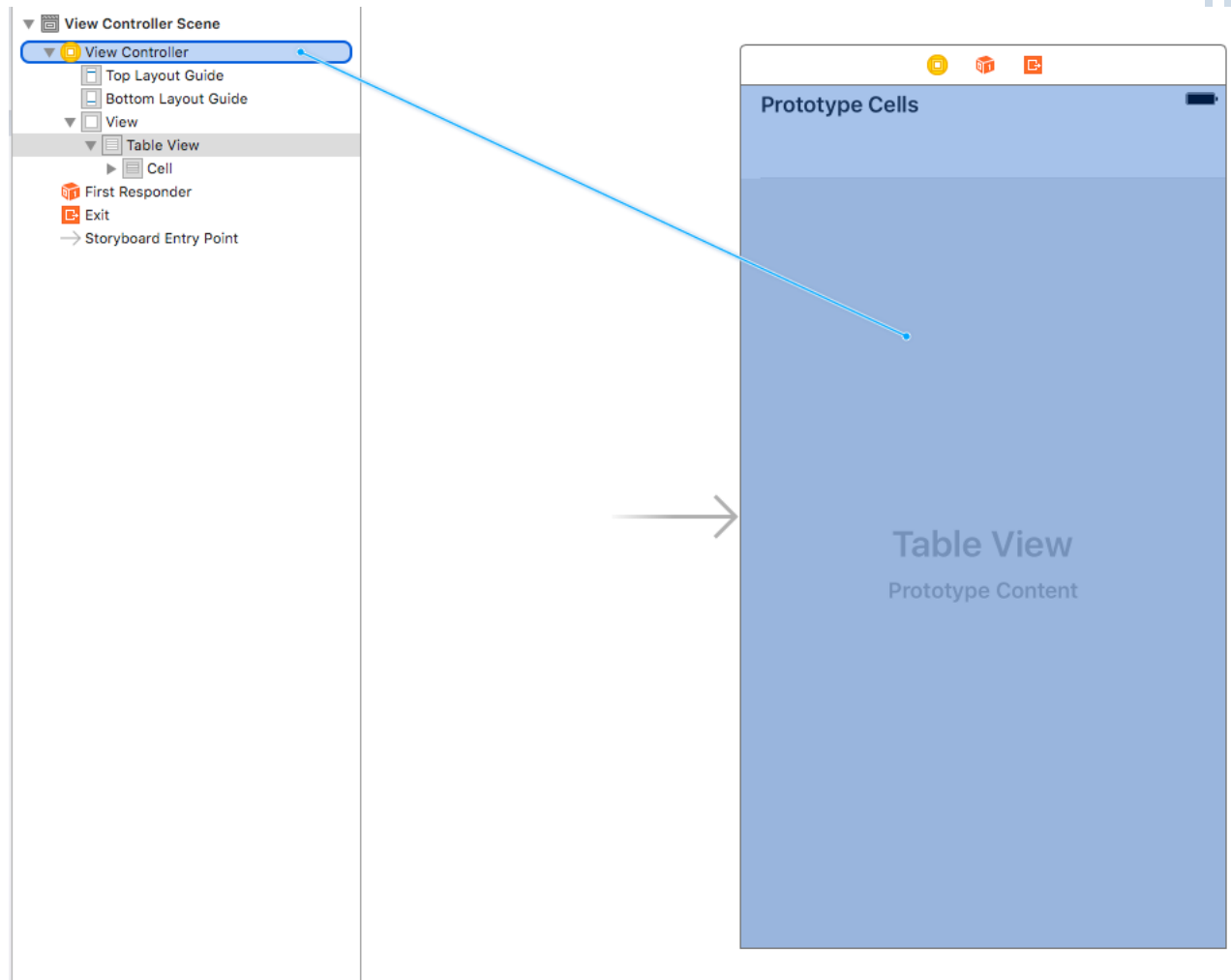
It would be called when each table views shown.
From indexPath we can get the present table view's row(indexPath.row)

```
//Create the cell
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell
{
}
}
```


CONNECT DATASOURCE AND DELEGATE

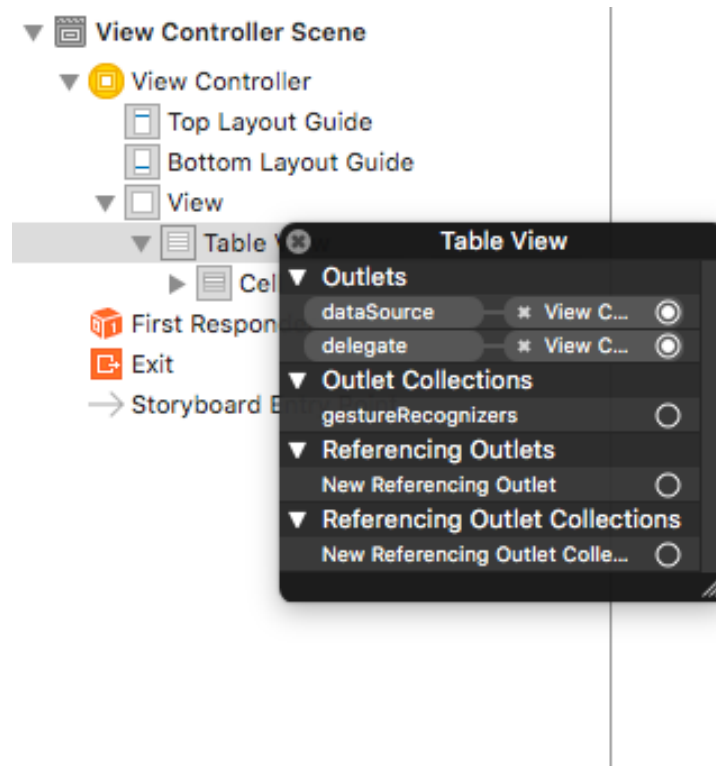
- Select View Controller and click “control” to connect Table View

Outlets
• dataSource
• delegate



CHECK CONNECTION

- Click table view to check your connection



STORE YOUR IMAGE

- Take your images into the file named “Assets.xcassets”

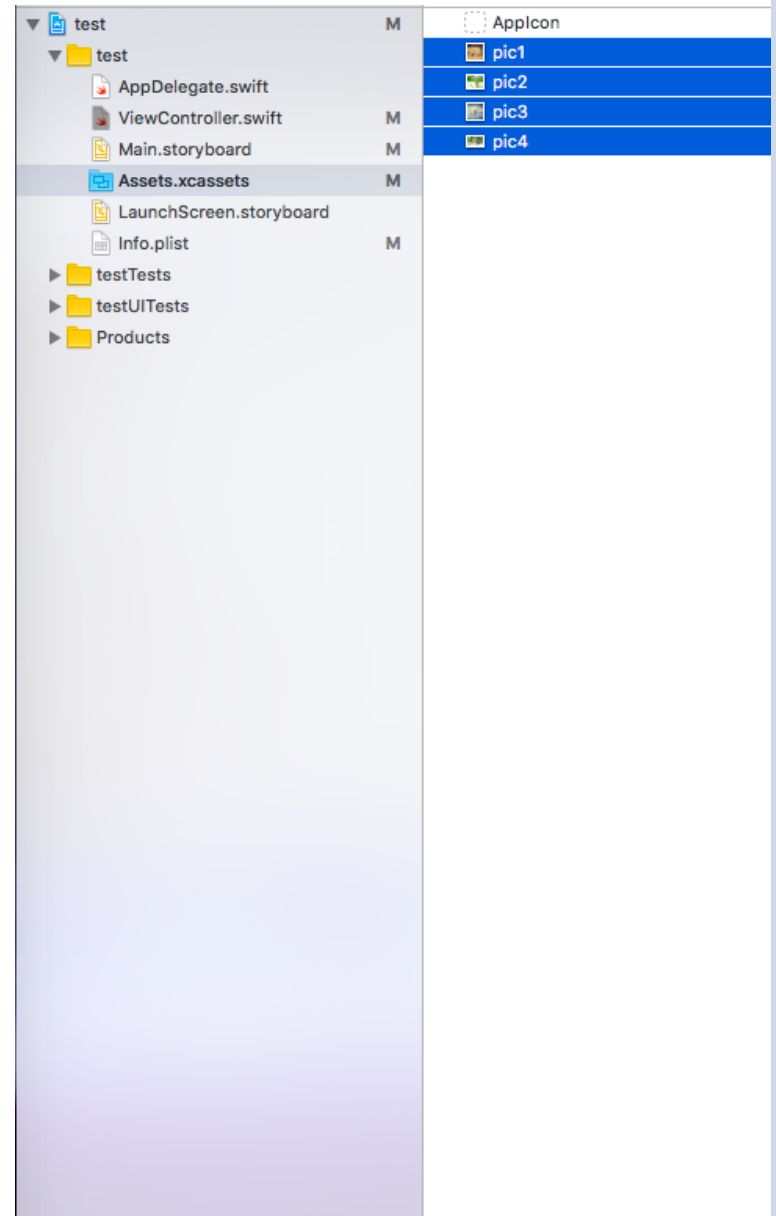


IMAGE NAMES

- Remember your image name need to be the same as image files in Assets.xcassets

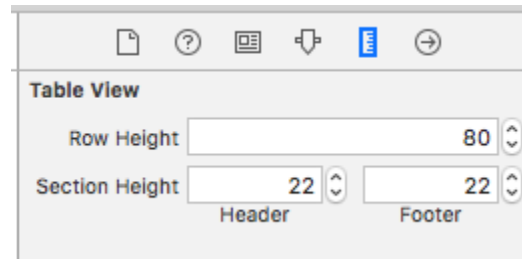
```
var items: [String] = ["#1", "#2", "#3", "#4", "#5", "#6", "#7", "#8"]
var itemdetail: [String] = ["Rainy Day", "Sunny Day", "Go Together", "Zone Out", "On The road", "Keep Balance", "Sleeping", "Hang Out"]
var itemtype: [String] = ["pic1", "pic2", "pic3", "pic4", "pic5", "pic6", "pic7", "pic8"]

var imagesname: [String] = ["pic1.jpg", "pic2.jpg", "pic3.jpg", "pic4.jpg", "pic5.jpg", "pic6.jpg", "pic7.jpg", "pic8.jpg"]
```

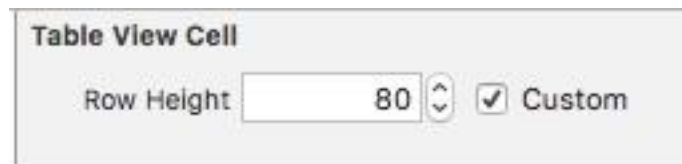


CHANGE ROW HIGH

- To change your each cell's high, you should change Table View's height first.

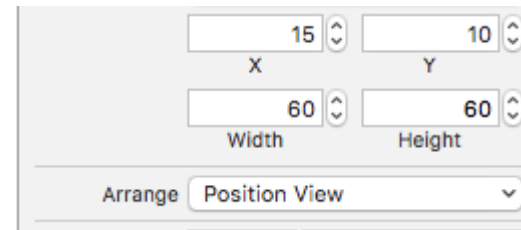
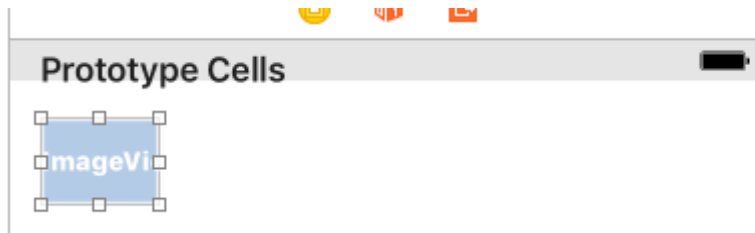


- Next click the cell, to click “Custom”
- Then your cell's height would be the same as your Table View's height



ADD THREE ELEMENT IN CONTENT VIEW

- Image View:
Chang “X”, ”Y”, ”Width”, ”Height”

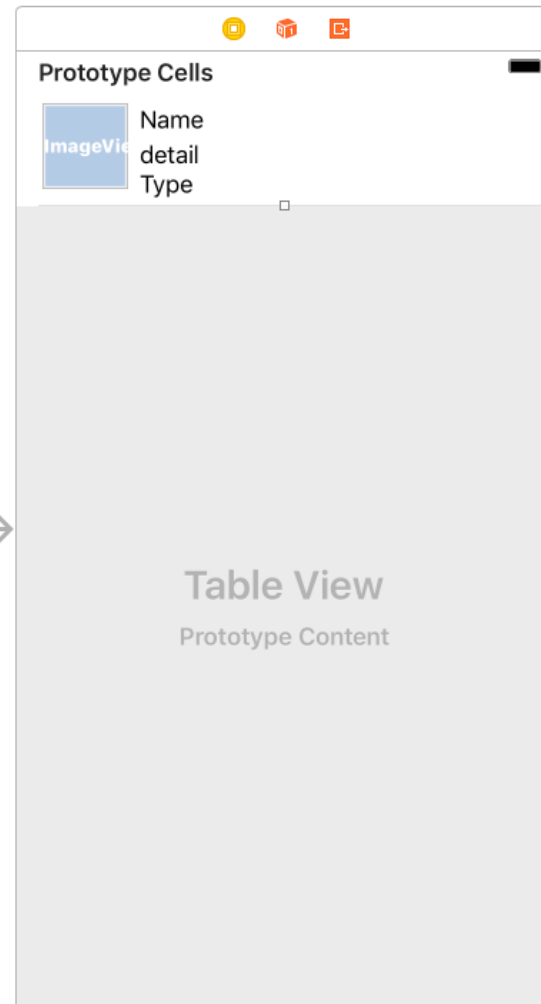
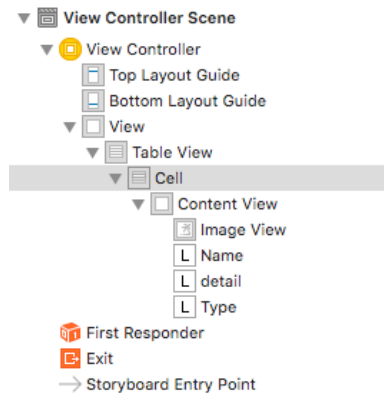


- Three labels: (Name, Detail, Type)

	X	Y	Width	Height
Name	86	9	205	21
Detail	86	35	205	18
Type	86	54	205	21

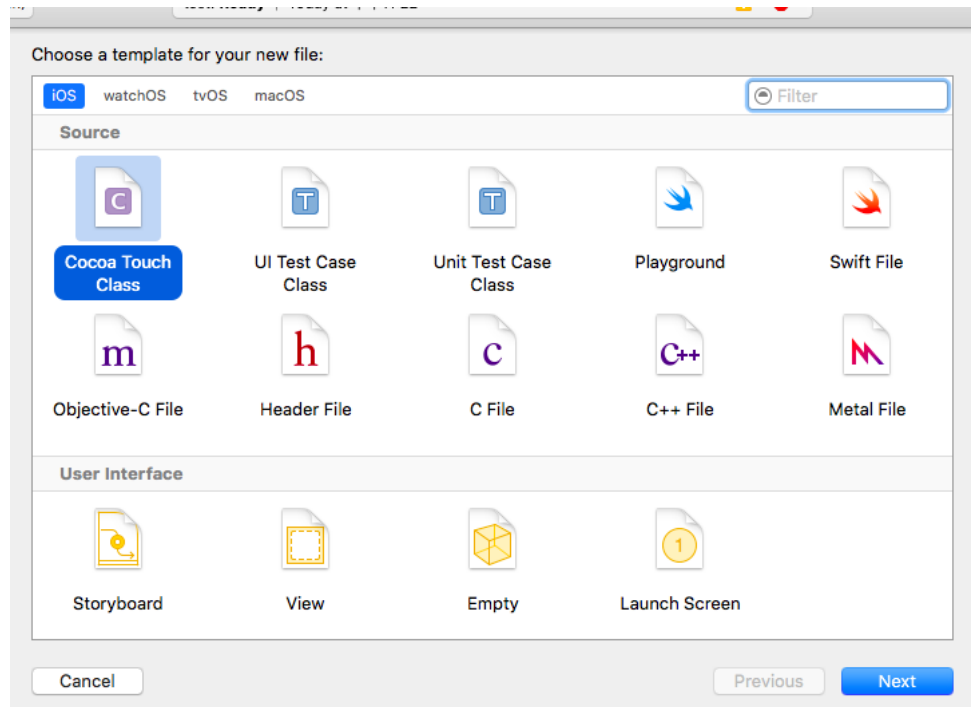


AFTER ADDING ELEMENTS



USE TABLEVIEWCELL TO INHERIT UITABLE-1

- To click your file name and select New file to build TableViewCell
- Select “Cocoa Touch Class” and click “Next”



USE TABLEVIEWCELL TO INHERIT UITABLE-2

- Enter your Class name and please remember it
- The next row “Subclass of ” select “UITableViewCell”

Choose options for your new file:

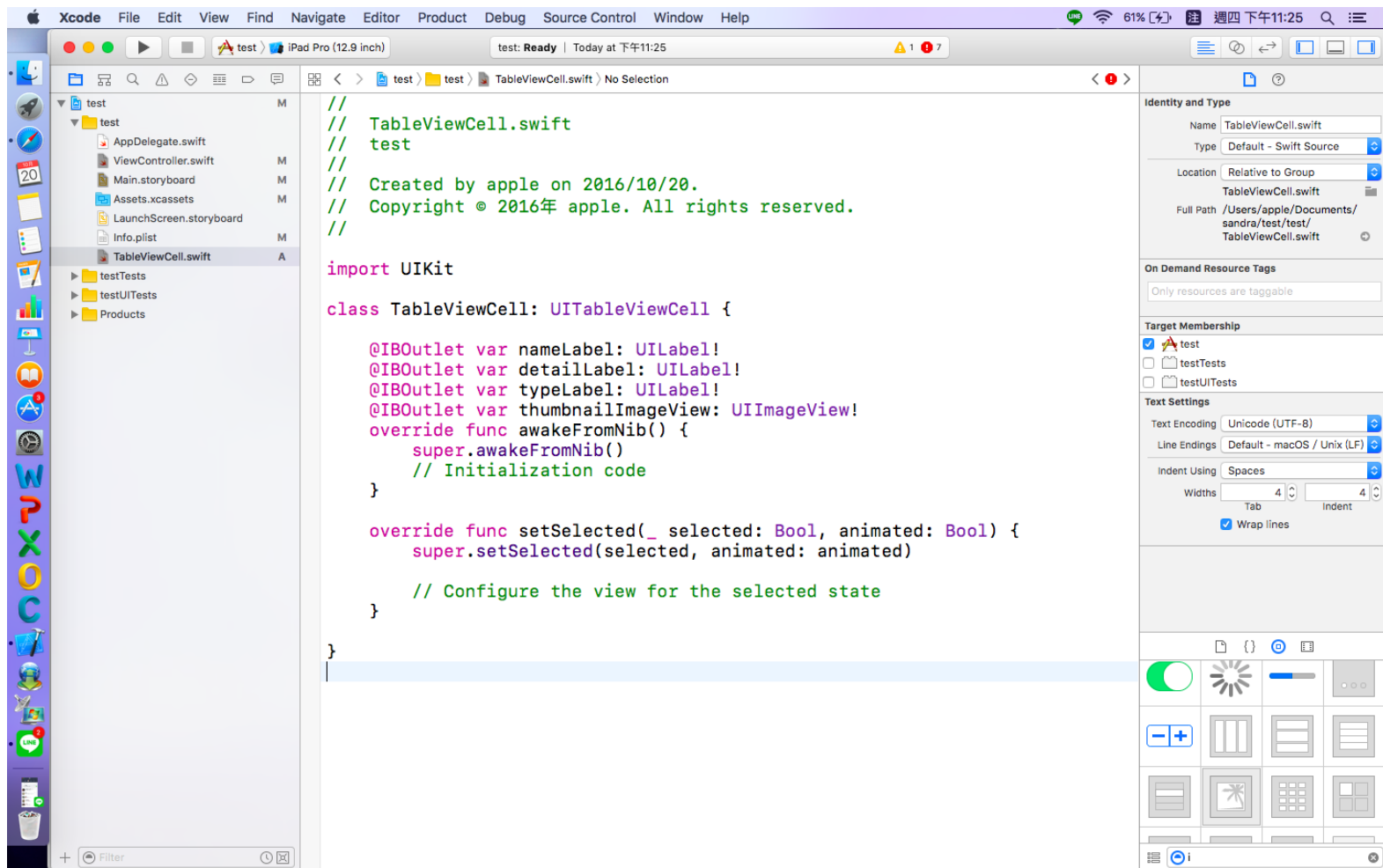
Class:

Subclass...

Also create XIB file

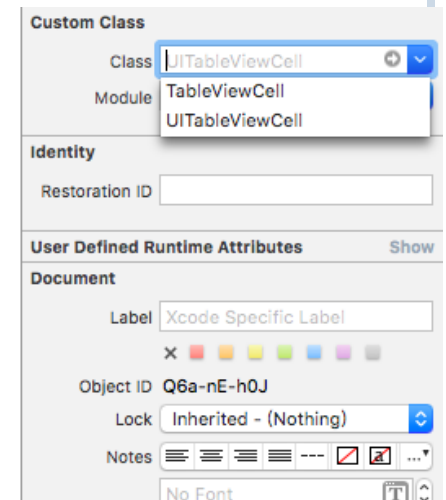
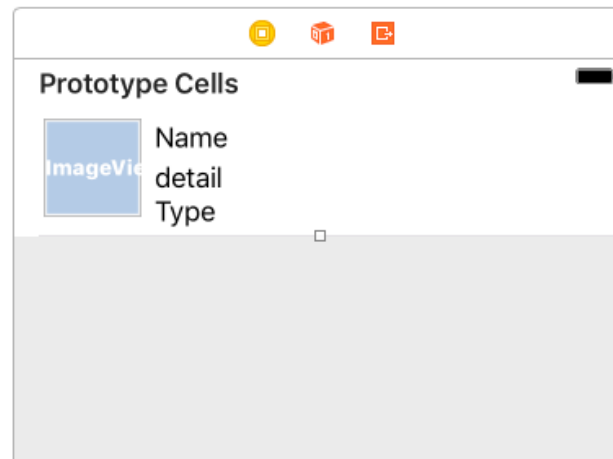
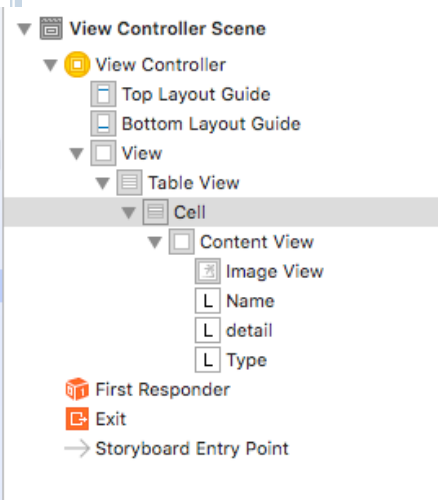
Language:

CLAIM THE OUTLET VARIANCE



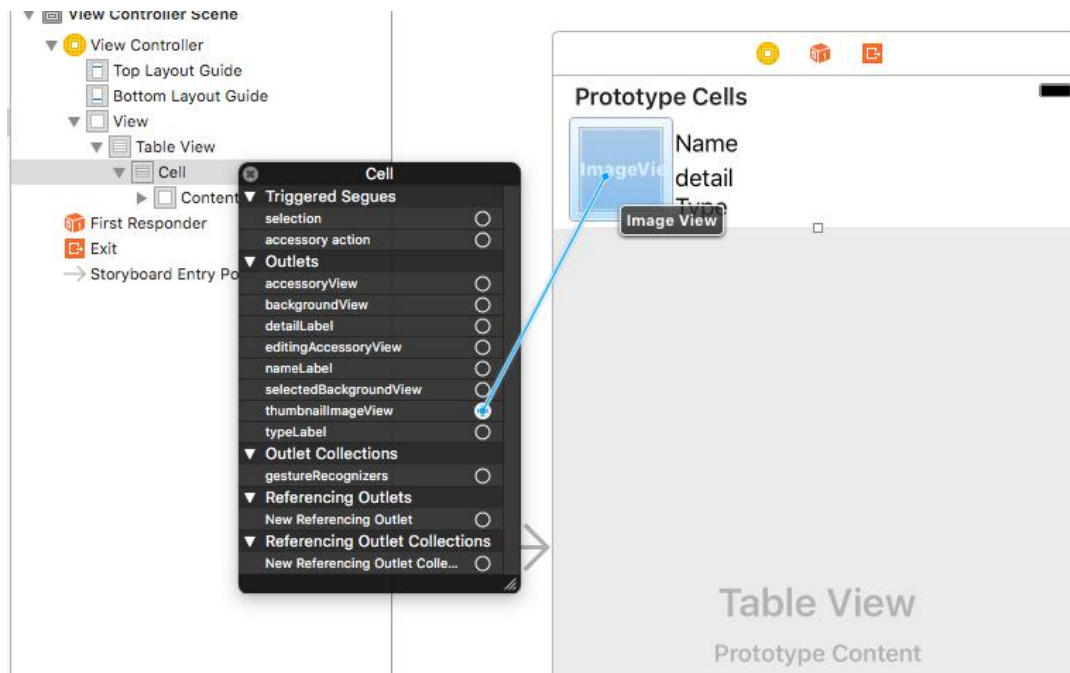
CONNECT PROGRAM AND UI-1

- Before connecting, you need to change Cell's "Custom Class"
- To select your file name which you build on the step above



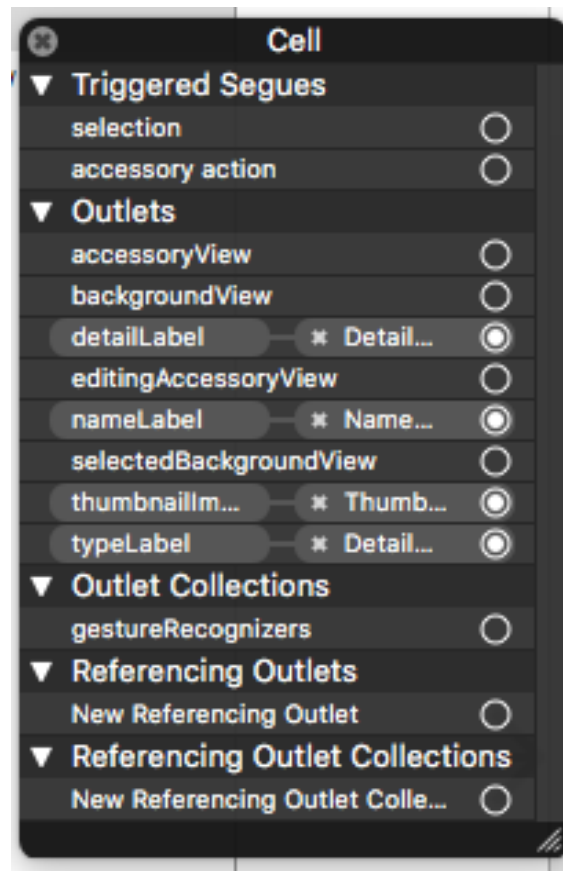
CONNECT PROGRAM AND UI-2

- Click right button on Cell
- It would show the Outlet, connect thumbnailImageView and Image View



CONNECT PROGRAM AND UI-3

- Repeat the step above
- To connect each Label and Cell Label



FINISH CELLFORROWATINDEXPATH

```
let cell = tableView.dequeueReusableCell(withIdentifier: "Cell",  
    for: indexPath) as! TableViewCell
```

```
cell.nameLabel.text = self.items[indexPath.row]  
cell.detailLabel.text = self.itemdetail[indexPath.row]  
cell.typeLabel.text = self.itemtype[indexPath.row]  
cell.thumbnailImageView.image = UIImage(named: imagesname  
    [indexPath.row])  
cell.thumbnailImageView.layer.cornerRadius =  
    cell.thumbnailImageView.bounds.size.width / 2  
//cell.thumbnailImageView.layer.cornerRadius = 10.0  
cell.thumbnailImageView.clipsToBounds = true  
return cell
```



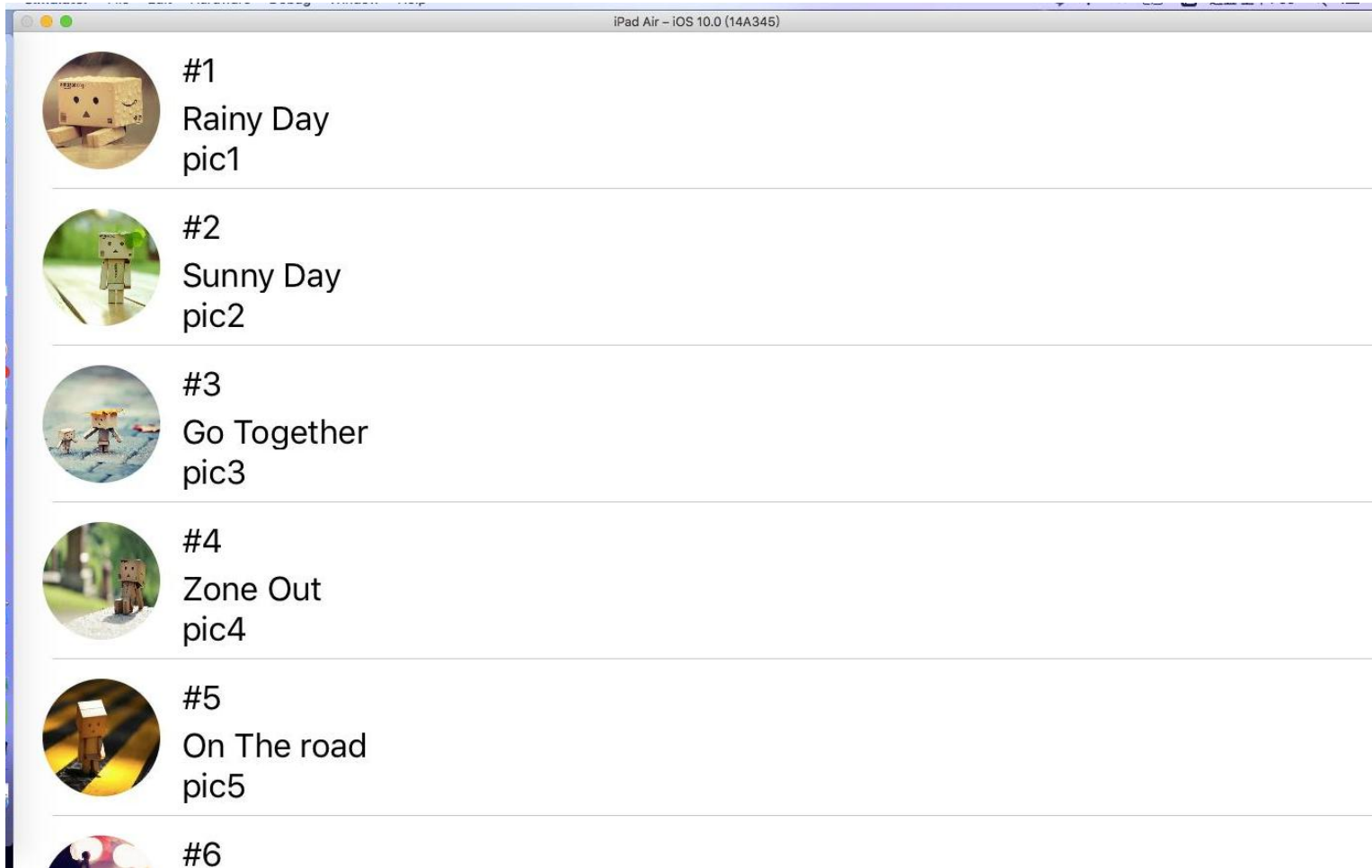
CELL SELECTION ACTION

```
//Cell selection
func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
    IndexPath) {
    //Deselects the selected cell.
    tableView.deselectRow(at: indexPath as IndexPath, animated: true)

    let alertView = UIAlertController()
    alertView.delegate = self
    alertView.title = "Message"
    alertView.message = "You choose Picture \(items[indexPath.row])!"
    alertView.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: nil))
    alertView.addAction(UIAlertAction(title: "Okay", style: .default, handler: nil))
    alertView.show()
    //print("You selected \(itemtype[indexPath.row])!")
}
}
```



RESULT-LAYOUT



RESULT-CELL SELECTION

