# Basic Exercises Part 9.2. Objective-C. TableView

## Table Views

* Table Views is one of the common UI elements in iOS apps. Most apps, make use of TableViews to display list of data. For example, the built-in iPhone contacts list displayed, or the Mail app.
* For more information visit:

**https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/TableView\_iPhone/CreateConfigureTableView/CreateConfigureTableView.html**

And this one:

**https://developer.apple.com/documentation/uikit/uitableview?language=objc**

### **1.1 Create a Simple project**

### Create a basic Single View. We will create a new project on each approach, so we only going to write the steps once. Use Objective C as language.

### A screenshot of a cell phone Description automatically generated

### Create a new file. Select a Cocoa Touch Class. Name it: RootTableView. Subclass of: UIViewController.

### As a result, you would get the .h file and the .m file.

### **1.2** **Storyboard**

In the storyboard, drag and drop a UITableViewController. Be sure to take all the space available. In the class identity inspector set the class name for your ViewController with the name you previously created.

A screenshot of a social media post

Description automatically generated

Remember to add a “Table View Cell” in order to display the rows. Select the property cell and make it a little bigger than the original size. You should see something similar to below:

A screenshot of a cell phone

Description automatically generated

### **1.3 Mock the data source.**

### In the .m file, create a property to mock the information to populate into the TableView. In the viewDidLoad method, add some strings into the “dataArray” property.

#import "RootViewController.h"

**@interface** RootViewController ()

**@property** NSMutableArray\* dataArray;

**@end**

**@implementation** RootViewController

- (**void**)viewDidLoad {

[**super** viewDidLoad];

**self**.dataArray = [[NSMutableArray alloc] init];

[**self**.dataArray addObject:@"TV monitor"];

[**self**.dataArray addObject:@"CD player"];

[**self**.dataArray addObject:@"Mp3 music"];

}

**@end**

### **1.4 Create the outlets.**

### Drag and drop the tableView outlet into the .h file.

A screenshot of a cell phone

Description automatically generated

Now the tableView is accessible in the .m file.

### **1.5 Protocols**

Conform the TableView delegate and datasource protocols within the .m file. The required methods are ‘numberOfRowsInSection’ and ‘cellForRowAtIndexPath’.

Link the delegate and datasource to self in the viewDidLoad.

### **1.6 Create the cell file**

Create a new tableView cell file. Select Cocoa Touch File. Name it as TableViewCell subclass of UITableViewCell. No code is required inside.

Link the new file Table View Cell with the cell inside the storyboard (use the identity inspector). Set the identifier for the cell in the attributes inspector; name it “cell”.

Import the TableViewCell.h file inside the RootViewController.m file.

* Now you can set the cell inside the row using dequeue reusable cell.
* Set the property ‘textLabel.text’ equals to the corresponding position of the array given the indexpath.

Built and run.

A screenshot of a cell phone

Description automatically generated

### **1.7 To-Do app**

Try to recreate the app of week 1. This time we don't need the TabBar, nor save the information, let's make the problem easier and the only thing to do is mock the information, and show a detailView when the user clicks in a row.