# Assignment week 4

# Table view and loading JSON Data

(More) Sources:

Alamofire (SWIFT-versie van AFNetworking)

https://github.com/Alamofire/Alamofire

Tutorial JSON at wenderlich’s site:

http://www.raywenderlich.com/82706/working-with-json-in-swift-tutorial

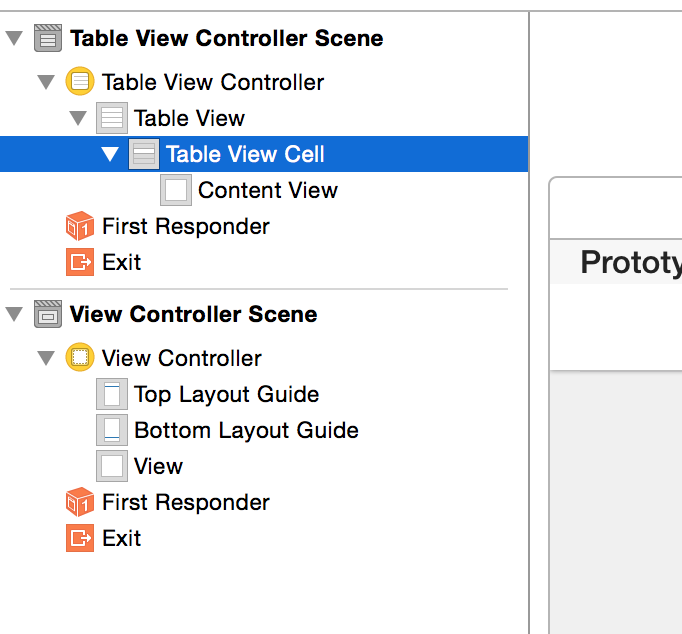
## Step 1: Creating the project and the storyboard.

Create a new Single View Application project and give it a good name.

You can safely delete the View Controller from main storyboard since we will not use it. Also delete the ViewController.swift.

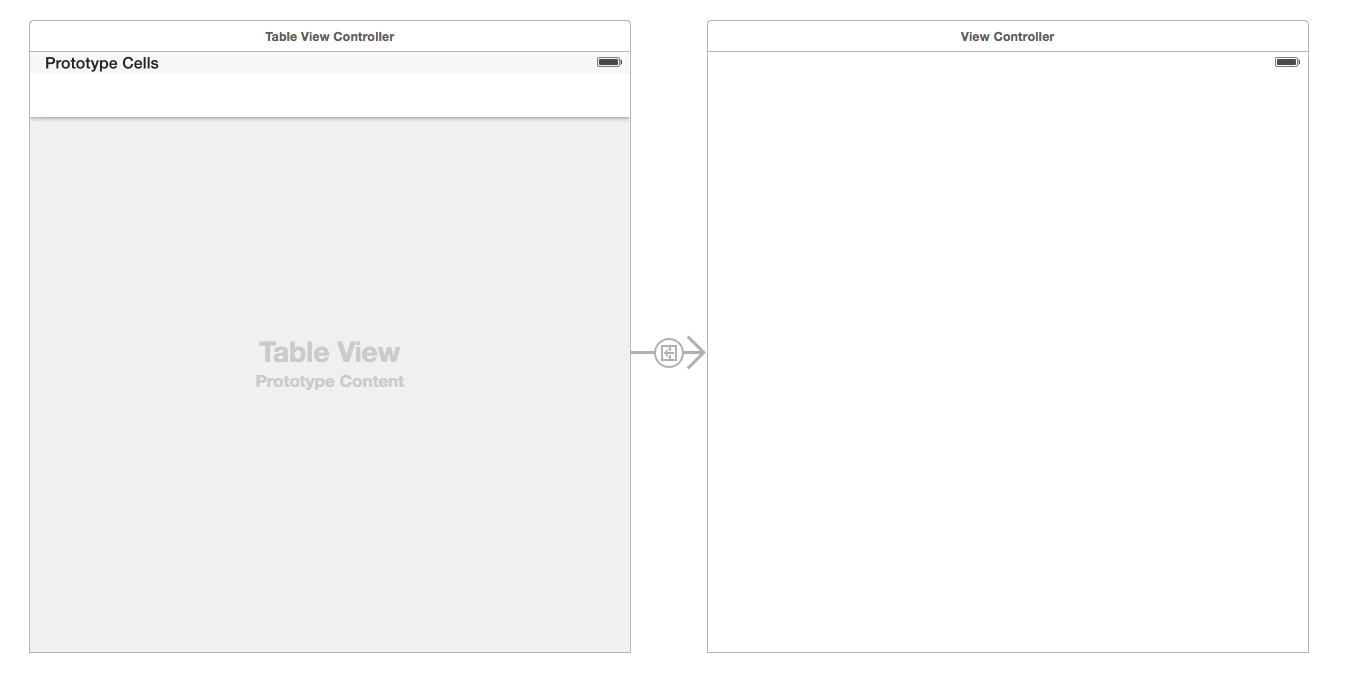
Now add a TableViewController to the storyboard.

Add a ViewController to the storyboard at the right side of the TableViewController. Now select the TableViewCell on the TableViewController, or in the Tree View left to it:



Now *ctrl+drag* the TableViewCell to the ViewController (you can choose to do this in the Storyboard or in the Tree View, whichever you think is most convenient) on the right to create a segue. From the popup menu choose ‘*Show’*.

You’ll end up with something like this:

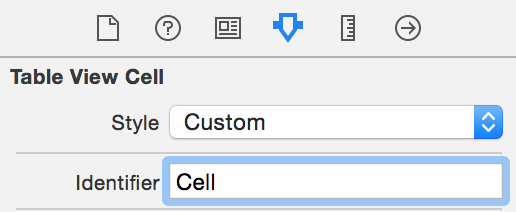


Set the Table View Controller to be the ‘Initial View Controller’.

The table view is about to show a list of pirate names and the Viewcontroller will show details about the selected pirate.

Select the TableViewCell again and set the *identifier* to ‘Cell’ in the *attributes*

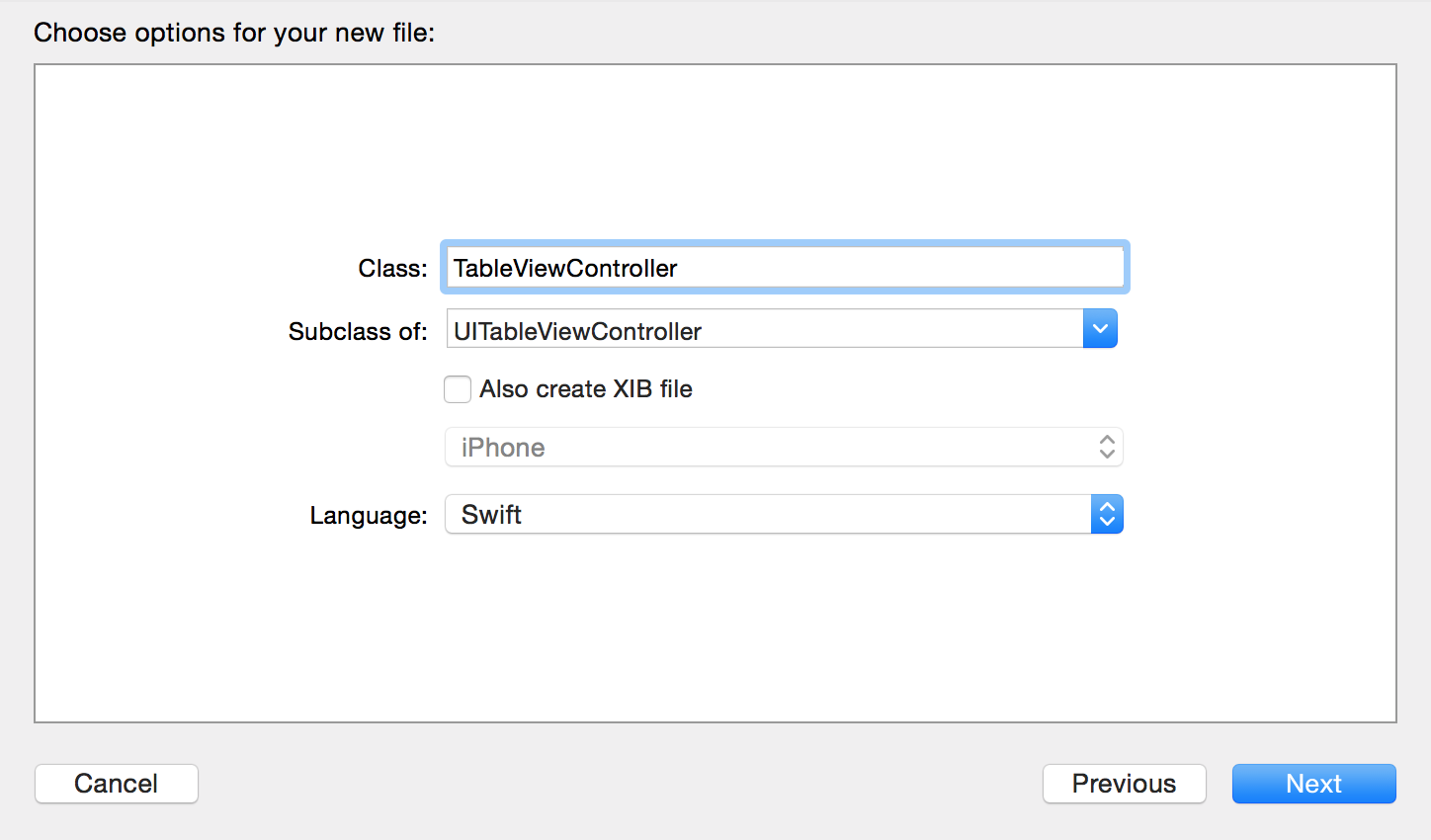
*inspector*.



## Step 2: Create the code behind the controllers

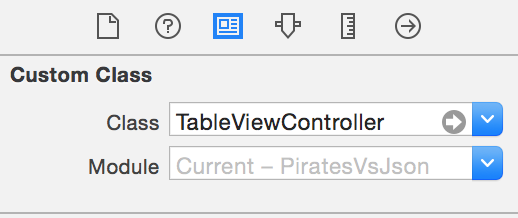
Create a new Cocoa Touch Class (file -> new file -> Cocoa Touch Class)

Make it a *subclass* of *UITableViewController* and give it a good name like ‘TableViewController’



Create another Cocoa Touch Class. This time make sure it’s a *subclass* of *UIViewController*. Give it a good name like ‘DetailsViewController’.

Select the TableViewController in the Storyboard and set the custom class.

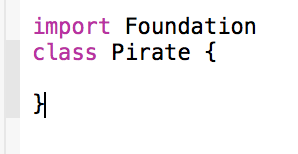


Do the same for the ViewController.

Finally create a new Swift file, call it ‘Pirate.swift’.



Add a class ‘Pirate’ to this file.

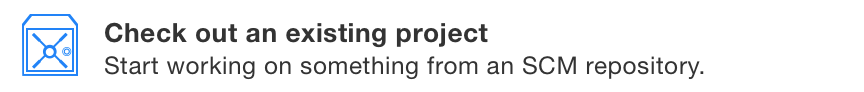


### Step 3: Add Alamofire to your project.

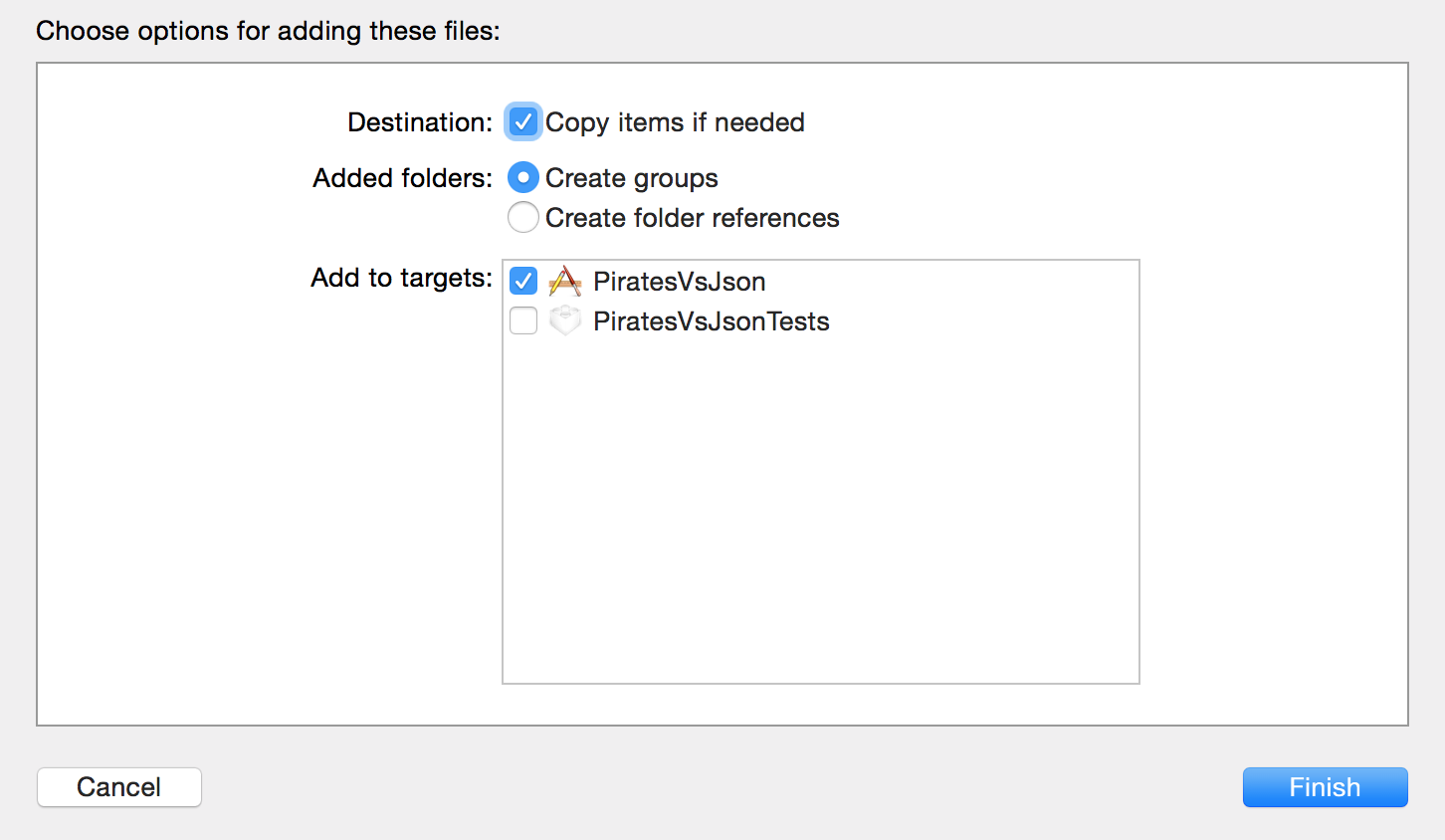
For the networking we are going to use a library called Alamofire, which is a Swift-version of the Objective-C AFNetworking library. Alamofire and its documentation can be found in GitHub:

<https://github.com/Alamofire/Alamofire>

All information you need to get it working in several ways is there, but for now it’s enough to either download the zip of this project (when you open the link above you find it in the menu on the right side in your browser), or if you know how to work with ‘*git*’ you can ‘*checkout*’ the project from the xcode startup screen:



For this tutorial we include a copy of the *Alamo.swift* file (you find it in the downloaded project) in our own project: you can do this by dragging it from the finder right to the place in xcode where you want it to be. Doing so you get an option form where you should make sure ‘Copy items if needed’ is clicked.



After clicking Finish the file is added to your project and where ready to use it:

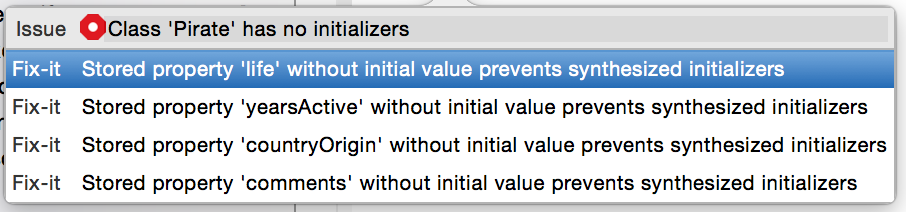


## Step 4: The Pirate class

Take a look at this week’s slides and add some properties to the Pirate class:

* name
* Life
* Years active
* Country of origin
* Comments

Having done this xcode will propose to fix issues for you,

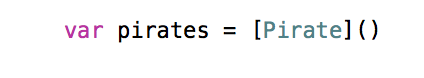


but let’s create an init method instead that fills all values:

## Macintosh HD:Users:fhict:Dropbox:Screenshots:Screenshot 2015-03-14 15.51.17.png

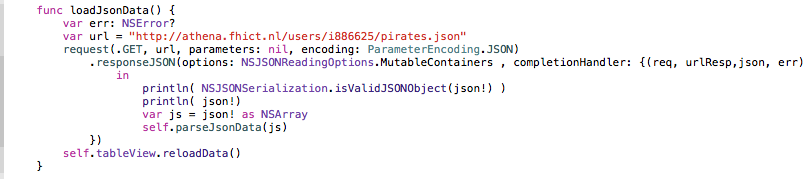
## Step 5: Initializing the array

In the TableViewController we are going to create a collection property to contain the pirates.

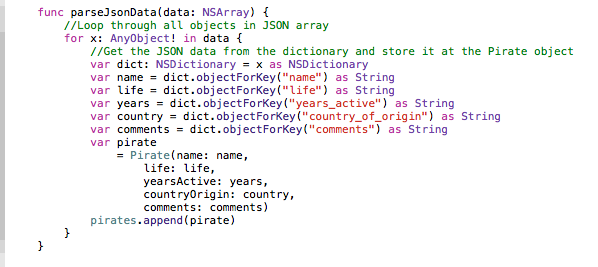


## Step 6: Load the JSON data

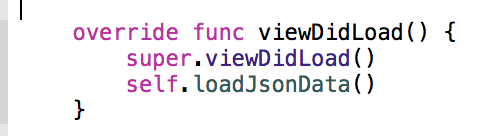
In this week’s slides you find a method called ‘*loadJsonData*()’ which we will add to our TableViewController implementation.



Now we are going to create the method ‘*parseJsonData*’ method that’s called from the method above. Compare it to the ‘*parseJsonData*’-method in the slides. When done parsing, the table should be reloaded. It looks like:

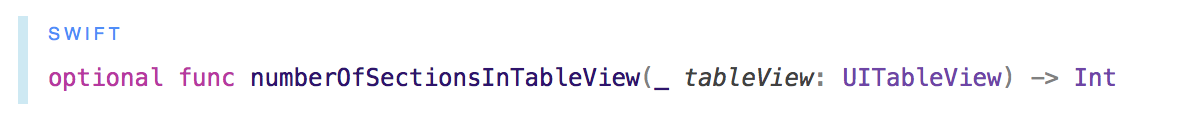


Finally call the loadJsonMethod in the ViewDidLoad method:

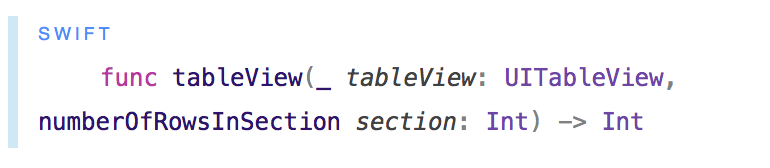


## Step 7: Show the Pirate Names

To fill the TableView we have to implement some methods. Go to the class declaration of your TableViewController class: it states that it is a subclass of UITableViewController: *alt+click* on the name of this superclass to see that this implements the *protocol* (compare to *interface* in C# or java) UITableViewDataSource, amongst others. Click on this name in the pop up to see the documentation. When the Table is displayed it will call methods of the data source to know what to display. For example the number of sections (~columns) of the table is determined by calling the method *numberOfSectionsInTableView*.



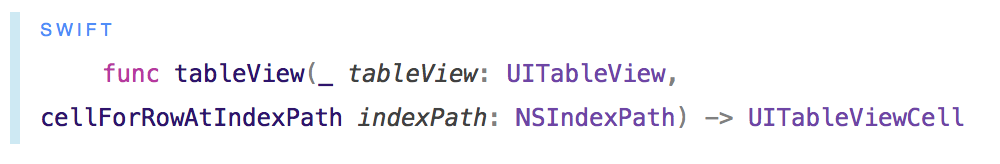
The default value returned by this method is 1. If you want more sections you have to override this method and make it return another number. Next important method from this protocol is numberOfRowsInSection, which is called when the view needs to know how many rows have to be displayed.



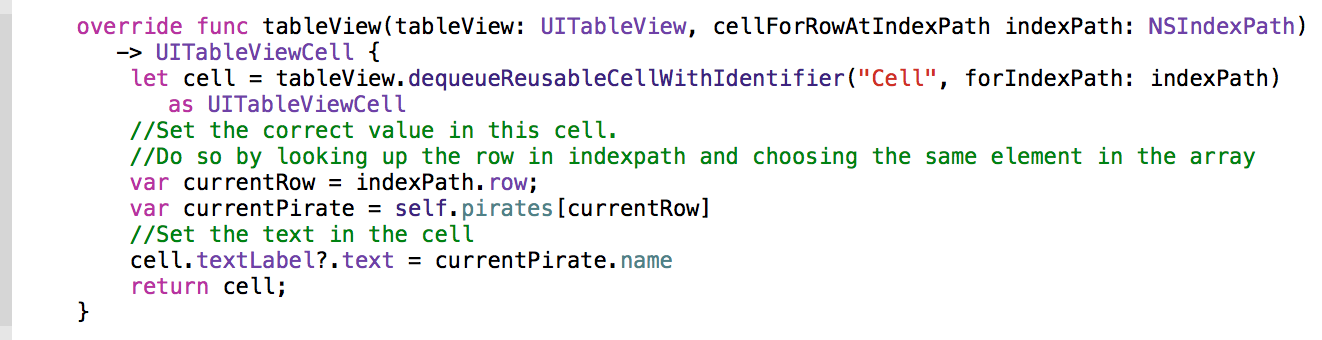
This methods needs to return the number of elements we want to show, so in this case we want to return the number of elements in the *pirates* collection.

Write the code to do so.

So now the TableView knows how many sections and rows we want to be displayed. The third and last important method of the data source we have to implement in our TableViewController is one that defines one cell of the Table. The TableView knows which cells will be visible to the user (so which cells have to be drawn on the screen), or these it will call this method:



This method should like:



## Step 8: Run the application

It should now look like this:



## Step 9: Show the details

Open the DetailViewController in the Storyboard and add the following UI elements: Add 5 labels on the left side, 4 labels and a TextView on the right side.

The left side labels should read:

- Name

- Life

- Active years

- Country of birth.

- Comments

The right labels and the TextView should display some text later on, so make sure that you create an IBOutlet to the controller code. Choose some good names.

## Step 10: Which details?

In the DetailViewController add a property to store a Pirate object.

Macintosh HD:Users:fhict:Dropbox:Screenshots:Screenshot 2015-03-15 21.47.51.png

As you probably know, the question mark means that this variable can have value nil.

## Step 10: Set the details.

Add and finish the following code in the DetailViewController by creating outlets for the labels and settings the text:

## Macintosh HD:Users:fhict:Dropbox:Screenshots:Screenshot 2015-03-15 22.31.52.png

## Step 11: Fix the Segui

Open TableViewController and add the following code:

