**Cleaner UITableViewControllers**



When Xcode generates a default UITableViewController, and it encourages everything to go into this single view controller. This can mean a large class that is difficult to maintain, and yet this is how production code is often shipped.

It is possible to separate out the dataSource from the view controller, and should be considered as an approach if the controller starts to get too large.

**Prerequisites:**

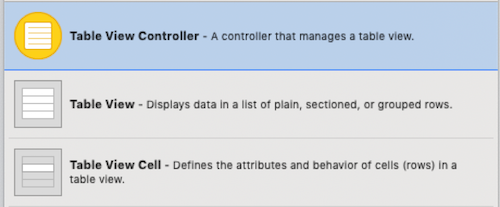
Creating view controllers Delegate pattern

A standard table view displays a list of items.

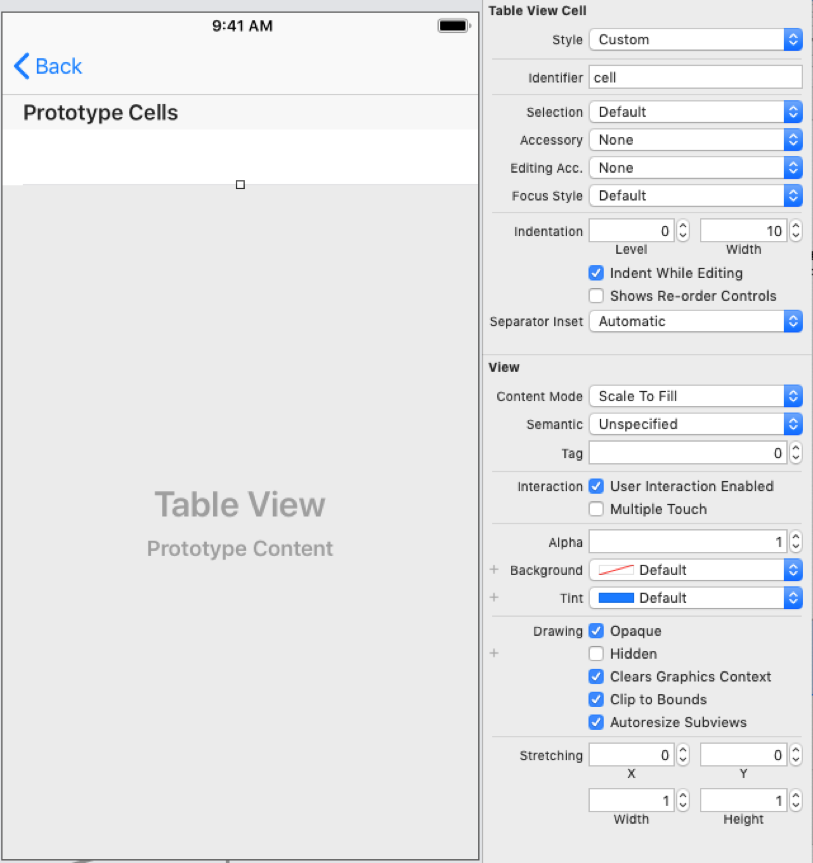
All of the following are presented on GitHub

**Standard table view**

You can simply add a TableViewController



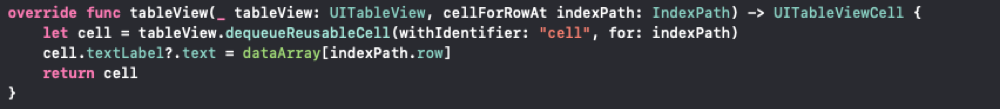
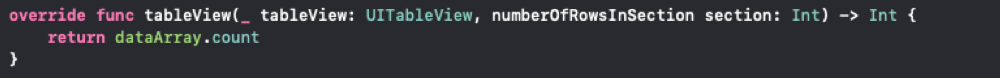
and it is essential for us to add the cell identifier. I call this “cell” for this example.



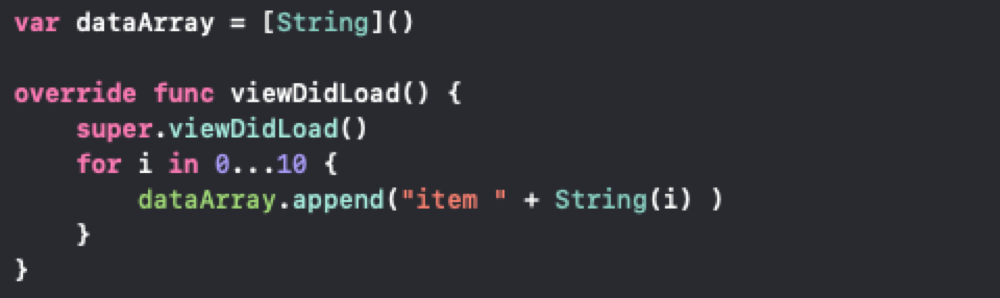
Setting up the view controller as a UITableViewController

https://cdn-images-1.medium.com/max/1600/1*coHBcjG9OSINr9CLAob1XQ.png

The datasource and delegate are hooked up from Interface Builder. The required methods set the number of rows for the table view and return the cell for each row. Since I named the embedded cell “cell” this has to match the one in cellForRowAt.



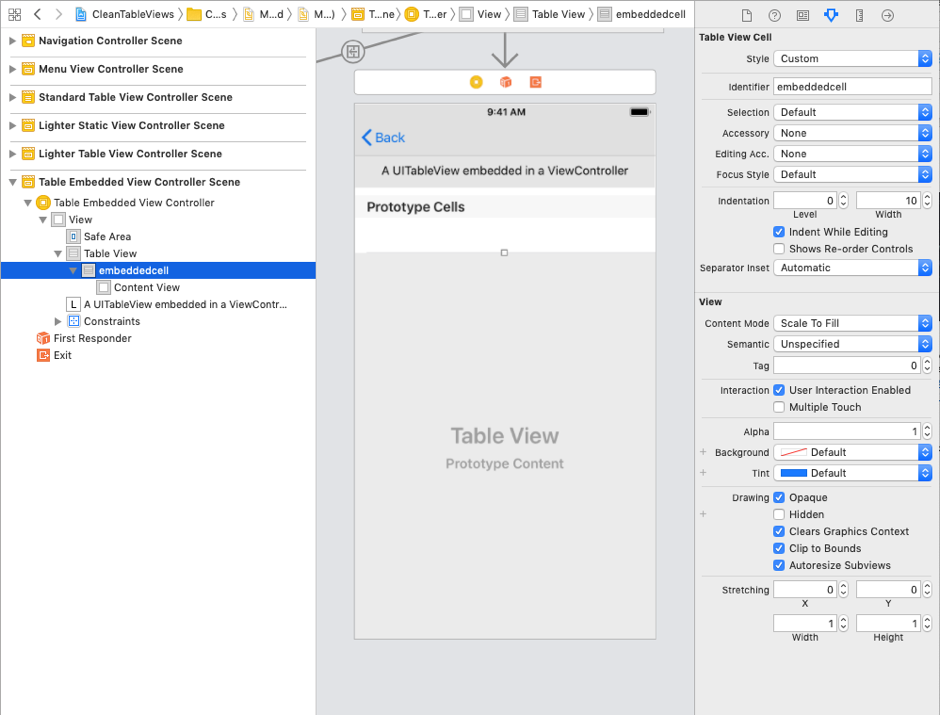
The data array is populated when the view loads



The whole view controller is here, I also implemented a headerView to maintain consistency in appearance with the other view controllers.

**Embedded table view**

An alternative approach is to embed the table view in a standard view controller. Note that I have to choose a new identifier for the embedded cell, and here I chose “embeddedcell”



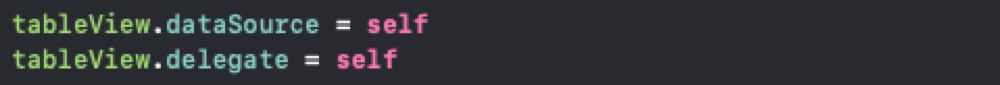
Which is set up view controller conforming to UITablveViewDataSource and as a delegate of UITableViewDelegate.

https://cdn-images-1.medium.com/max/1600/1*DVo6Olfy_-sA-uppRqsMRA.png

This means that we need to set up an outlet to hook Interface Builder into the view controller.

https://cdn-images-1.medium.com/max/1600/1*acT5s6RpJcNIocihAyX7Cg.png

and ensure that the data source and delegate are set.



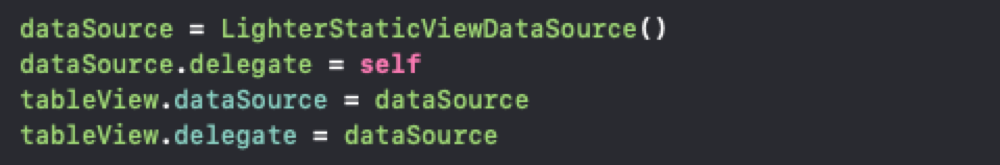
It is debatable how much more flexibility thus gives us over the header (and possibly footer) defined in an ordinary UITableViewController.

What is certain, however, is that we have a large class with things that arguably a view controller should not “know about”

Let us split that out

**A lighter view controller**

Similar to the embedded version above, but of course we need an extra file for the data source, and we require a reference to it from our view controller:

https://cdn-images-1.medium.com/max/1600/1*zXi-o4kED1vsolVqdIP1ZA.png

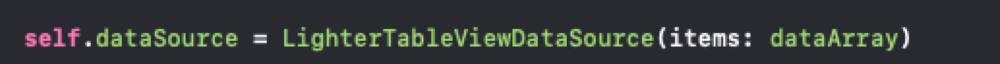
The data source will still require cellForRowAt and numberOfRowsInSection.

We have separated the data source into the separate file.

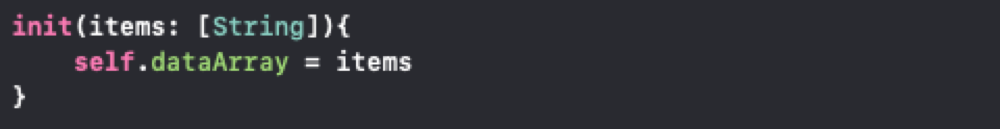
However, there might be an issue if you want to pass the data from the view controller to the data source. How would we do that?

**A lighter view controller with data in the view controller**

As an example, we can absolutely pass the data from the view controller to the data source. Within the view controller we will pass our data



which involves creating an initialiser for our data source:



**A Link to GitHub**

The completed code is here: <https://github.com/stevencurtis/cleantableviews>

**The Twitter contact:**

Any questions? You can get in touch with me [HERE](https://twitter.com/stevenpcurtis)