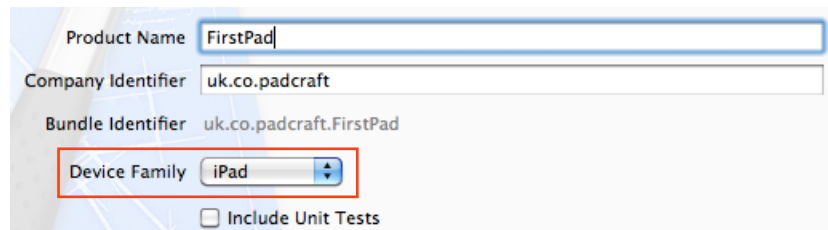


Workshop 10: Developing iPad applications

In this final workshop we look at the differences between iPhone and iPad apps.

Task 1: Creating an iPad app

There is only ONE thing extra you need to do to create an iPad app. When creating a new 'View-based Application', change the Device Family from 'iPhone' to 'iPad':



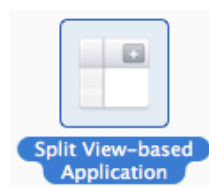
Create an iPad app called 'FirstPad'. Open the FirstPadViewController.xib and you will see it is bigger than before. Add a label, then run the app.

The iOS Simulator will run as an iPad. Everything else is the same!

Task 2: Split views

The only difference between iPhone and iPad is the screen size. But a bigger screen means that there are some different interface objects. Let's take a look at the 'Split View'.

Create a new project of type 'Split View-based Application':



Call your new project 'PhotosPad'.

Run the app. Try rotating the device left and right.

Task 3: Left side of the split view

Xcode automatically creates a RootViewController and a DetailViewController for you (like an iPhone Navigation project). The left side of the split view is the RootViewController. It is a UITableViewController (like you have seen many times before).

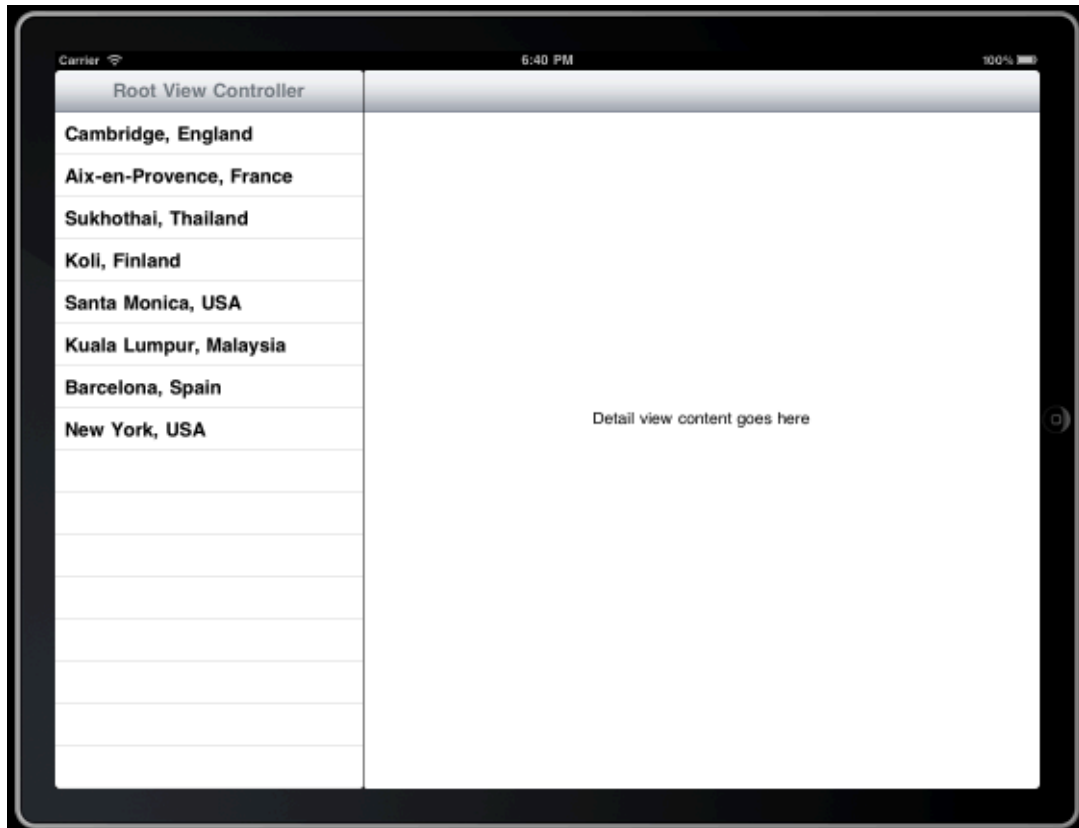
Grab the [workshop files](#) and add them to your project.

You can get an NSArray of Photo objects using:

```
yourArray = [CollectPhotos photos];
```

Set up your RootViewController to get all the Photo objects and show them in the table. (The Photo object has a property called 'title'.)

Run the app. Can you see your list of photo titles?



Task 4: Right side of split view

Next open the DetailViewController and add a UIImageView (remember 4 things!).

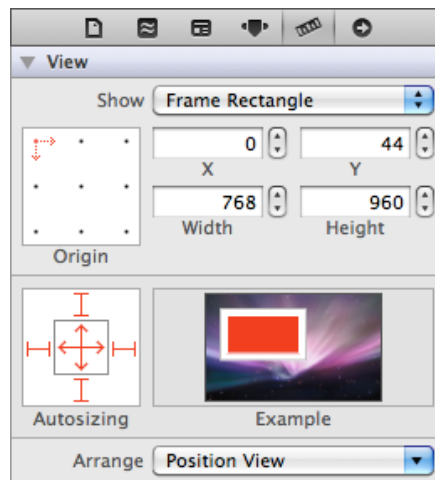
In the RootViewController.m, add the following code to the didSelectRowAtIndexPath method:

```
Photo *p = [items objectAtIndex:indexPath.row];
detailViewController.imageView.image =
    [UIImage imageNamed:p.imageName];
```

Note this is different to the iPhone: when we select a row in the table, we do not create a new view controller because it already exists. Instead, we just set the image property of the imageView.

Run the app. Can you see photos?

If you have a problem with the image size then try setting the 'Autosizing' properties on the Image View as follows:



Your final app should look like this:

