

iOS

# Use Storyboards to Build Navigation Controller and Table View

SIMON NG 15TH JUN '12 86

34 Shares

- f 1
- t 8
- g+ 19
- in 6

By now, if you've followed our tutorials, you should have a basic understanding about UITableView and how to build a simple app. This week, we'll talk about something new – **Storyboards**. This is one of the most exciting features introduced in Xcode 4.2 and iOS 5 SDK. It makes your life, as an iOS developer, simpler and lets you easily design the user interface of your iOS app.

In this tutorial, we'll show you how to use Storyboards to build a Navigation interface and integrate it with UITableView. We try to keep thing simple and focus on explaining the concept. So no fancy interface or pretty graphic. We'll leave the artwork to future tutorials.

Okay, let's get started.

## What's Navigation Controller?

Before we move onto the coding part, as usual, let's have a brief introduction about Navigation Controller and Storyboards.

Like table view, navigation controller is another UI element you commonly find in iOS app. It provides a drill-down interface for hierarchical content. Take a look at the built-in Photos app, YouTube, and Contacts. They're to display hierarchical content. Usually table view and navigation controller work hand in hand. It doesn't mean you have to use both together, however.



An example of Navigation Controller – Photos App

### Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

Storyboards Overview

PREVIOUS POST

Delete a Row from UITableView and Model-View-Controller

NEXT POST

Storyboards Segue Tutorial: Pass Data Between View Controllers

AppCoda is one of the leading iOS programming communities. Our aim is to teach everyone how to build apps with high quality and easy-to-read tutorials. Learn by doing is the heart of our learning materials.

### OUR PRODUCTS

#### RSS App Template

Save you thousands of dollars. Simply plug your own RSS feeds and turn the Xcode template into a RSS reader or a Blog reader app.

OUR

### COURSE

#### Beginning iOS 11 Programming with Swift

Learn how to code in Swift and build a real world app from scratch. Now supports Xcode 9, Swift 4 and iOS 11.

34 Shares

f 1

t 8

g+ 19

in 6

### MEET APPCODA

- About Our Team
- Write for Us
- Advertise

Copyright © AppCoda. 2018 • All rights reserved.  
Terms of Service | Privacy Policy | RSS Feed | Contact Us

TWITTER FACEBOOK GITHUB



### Scene and Segues

When working with Storyboards, Scene and Segues are two terms you always come across. Within a Storyboard, a scene refers to a single view controller and its view. Each scene has a dock, which is used primarily to make transitions between the view controller and its views.

Segue sits between two scenes and manages the transition between two scenes. Push and Modal Segues are the most common types of segue. A segue is a transition between two scenes. Push and Modal Segues are the most common types of segue.

### Creating Navigation Controller in Storyboards

Now let's get our hands dirty and create our own Storyboards. In this tutorial, we'll build a simple app that displays a list of recipes. We use the table view to display a list of recipes. When the app navigates to the next screen showing the details. It'll be easy.

First, fire up Xcode (make sure you're using 4.2 or up) and create a new project using "Single View Application".



### Want to Learn Swift?

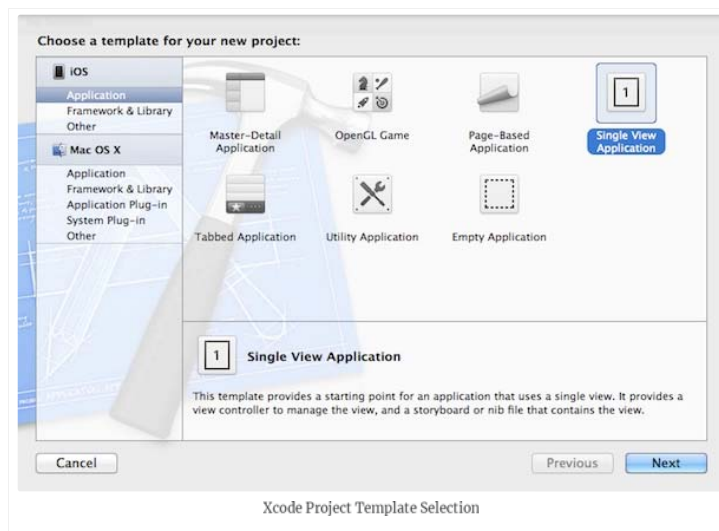
Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

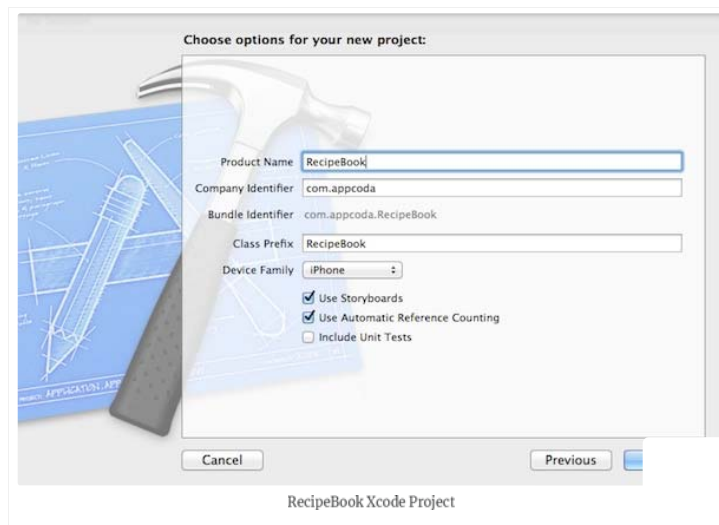
This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

34 Shares  
f 1  
t 8  
g+ 19  
in 6



Click "Next" to continue. In reference to the below figure, fill in all the required values for the Xcode project. Make sure you enables the "Use Storyboards" option.



Click "Next" to continue. Xcode then asks you where you saves the "SimpleTable" project. F your project.

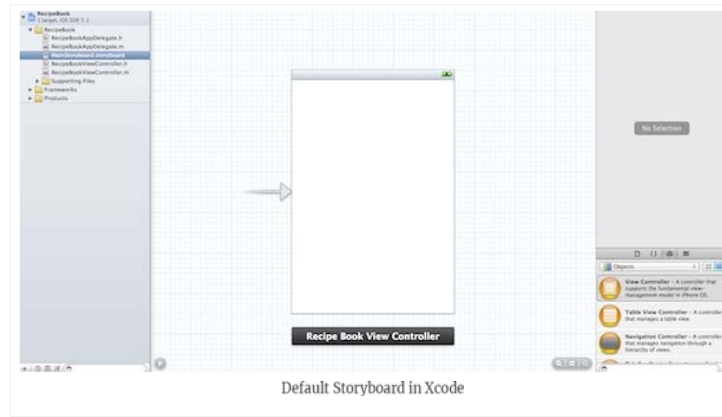
You may notice there is a minor difference in the Xcode project, as compared with those you The .xib file (interface builder) is replaced with the MainStoryboard.storyboard file.

### Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required. **DOWNLOAD NOW**

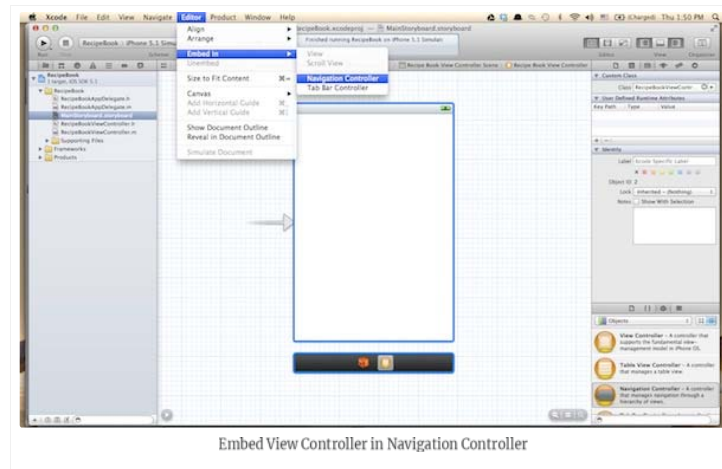
This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook



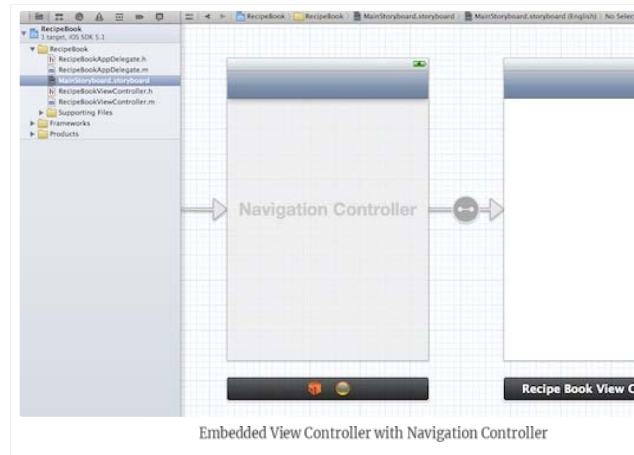
Default Storyboard in Xcode

By default, Xcode creates a standard view controller. As we'll use navigation controller to control the screen navigation, we first change the view controller to navigation controller. Select the Simply select "Editor" in the menu and select "Embed in", followed by "Navigation Controller".



Embed View Controller in Navigation Controller

Xcode automatically embeds the RecipeBook View Controller with Navigation Controller. Yo



Embedded View Controller with Navigation Controller







**Want to Learn Swift?**  
Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.  
**DOWNLOAD NOW**

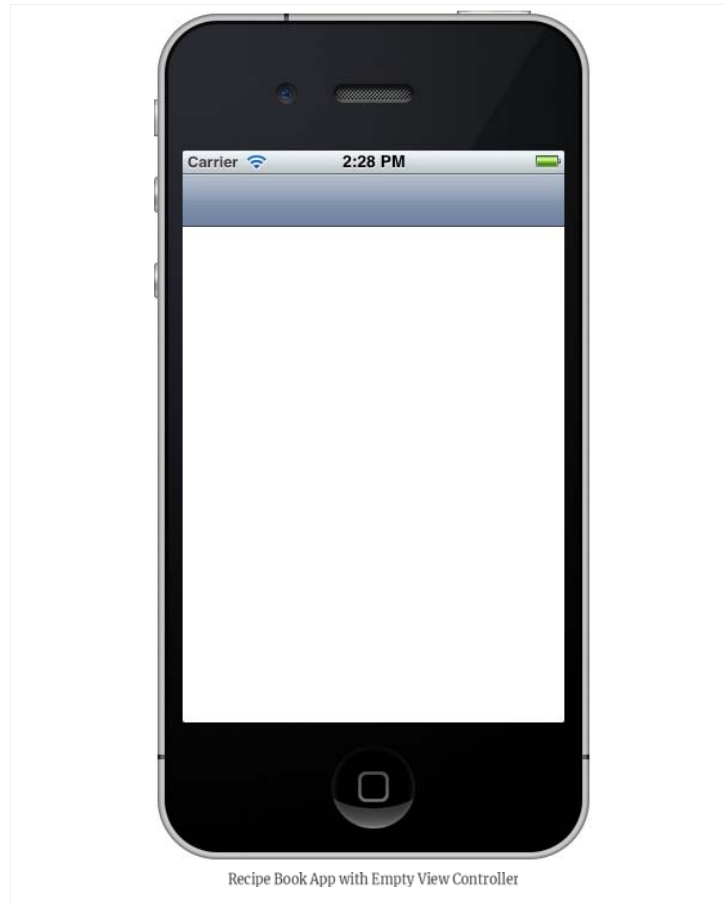
This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

34 Shares  
f  
1  
8  
g+  
19  
in  
6

34 Shares

-  1
-  8
-  19
-  6



## Adding Table View for Your Data

Next, we'll add a table view for displaying our recipes. Select "Table View" in Object Library and drag it into "Recipe Book View Controller".

Please note you can't drag stuff when the editor is zoomed out. If you can't drag the table view and try again.



**Want to Learn Swift?**  
 Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.  
**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

[Send me the eBook](#)

The next thing we have to do is to write code to populate the table data (i.e. recipes). In Project "RecipeBookViewController.h". Append "" after "UIViewController". Your code should look

```
#import <UIKit/UIKit.h>

@interface RecipeBookViewController : UIViewController <UITableViewDelegate, UITableViewDataSource>

@end
```

If you've read our Simple Table tutorial, you should be very familiar with the code. I'll not explain the code in details. If you find it difficult to follow, check out [our earlier tutorial](#).

Next, select "RecipeBookViewController.m" and define an instance variable (i.e. recipes array) for holding the table data.

```
@implementation RecipeBookViewController {
    NSMutableArray *recipes;
}
```

In the "viewDidLoad" method, add the following code to initialize the "recipes" array:

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Initialize table data
    recipes = [NSMutableArray arrayWithObjects:@"Egg Benedict", @"Mushroom Risotto", @"Full Breakfast", @"Hamburger", @"Ha
}
```

Lastly, we have to implement two datasource methods to populate the table data: "tableView:numberOfRowsInSection" and "tableView:cellForRowAtIndexPath". Recalled that these two methods are part of the UITableViewDataSource protocol, it's mandatory to implement the methods when configuring a UITableView. The first method is used to inform the table view how many rows are in the section, while the second method is used to fill the cell data. So let's add the below code.

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [recipes count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *simpleTableIdentifier = @"RecipeCell";

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:simpleTableIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:simpleTableIdentifier];
    }

    cell.textLabel.text = [recipes objectAtIndex:indexPath.row];
    return cell;
}
```



## Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

For your reference, this is the complete source code of "RecipeBookViewController.m".

```
//
// RecipeBookViewController.m
// RecipeBook
```

Send me the eBook

34  
Shares



1



8



19



6



```

//
// Created by Simon Ng on 14/6/12.
// Copyright (c) 2012 Appcoda. All rights reserved.
//

#import "RecipeBookViewController.h"

@interface RecipeBookViewController ()

@end

@implementation RecipeBookViewController {
    NSArray *recipes;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Initialize table data
    recipes = [NSArray arrayWithObjects:@"Egg Benedict", @"Mushroom Risotto", @"Full Breakfast", @"Hamburger", @"Ha

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [recipes count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *simpleTableIdentifier = @"RecipeCell";

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:simpleTableIdentifier];

    if (cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault r

        cell.textLabel.text = [recipes objectAtIndex:indexPath.row];
        return cell;
    }
}

@end

```

34

Shares

f

1

t

8

g+

19

in

6

x



## Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

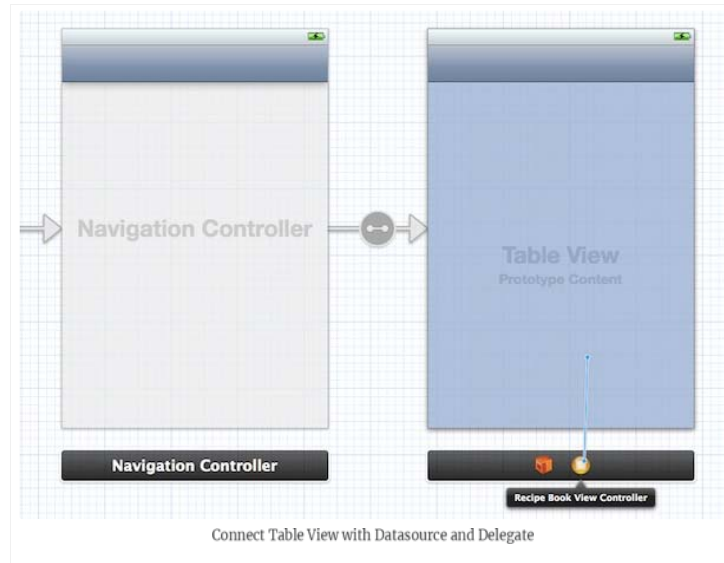
**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

Lastly, we have to establish the connection between the Table View and the two methods we Storyboard. Press and hold the Control key on your keyboard, select the “Table View” and the screen should look like this:

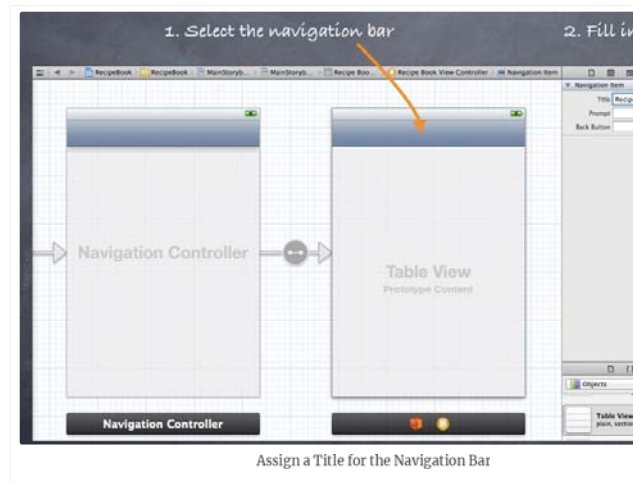
34 Shares  
 f 1  
 t 8  
 g+ 19  
 in 6



Release both buttons and a pop-up shows both “dataSource” & “delegate”. Select “dataSource” to make a connection between the Table View and its data source. Repeat the above steps and make a connection with the delegate.



Before we test out the app, one last thing to do is add a title for the navigation bar. Simply select the navigation bar of “Recipe Book View Controller” and fill in the “Title” under “Attributes Inspector”. Remember to hit ENTER after keying in the title to effectuate the change.



### Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required. **DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

[Send me the eBook](#)



34  
Shares

1



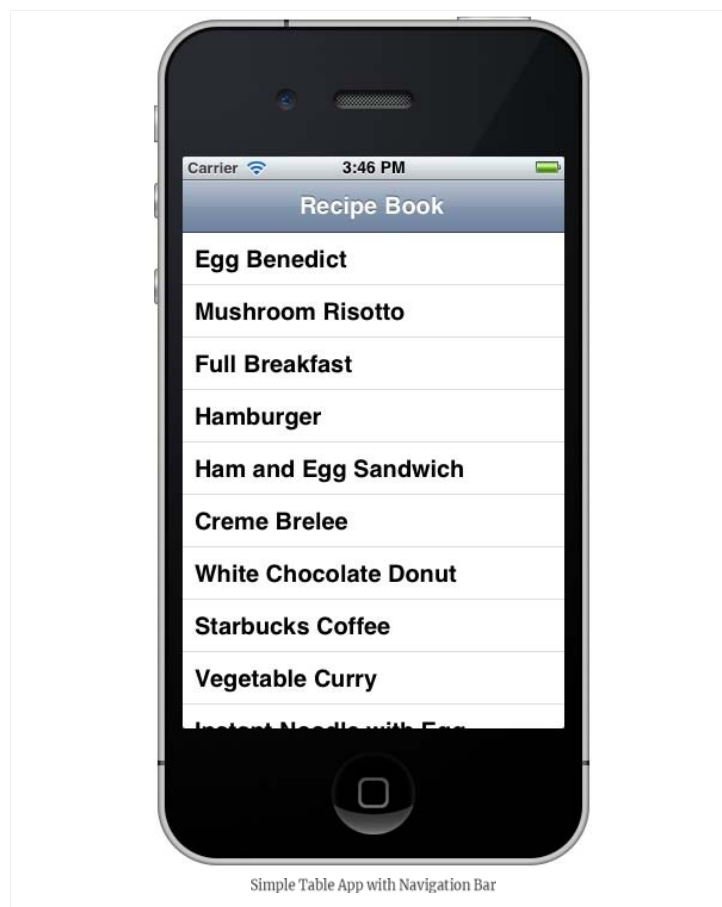
8



19



6



## Introducing Prototype Cell

Do you remember how we [customize the table cell](#)? Several weeks ago, we showed you how to design your own custom table cell using Interface Builder. In brief, you need to create a separate nib for the cell and programmatically load it in the table. With the introduction of Prototype Cell in Storyboard, it's much simpler to create a custom cell. Prototype cell allows you to easily design the layout of a table cell right in the Storyboard editor. ✕

We will not go into the details of the customization in this tutorial but just simply add “Disc

To add a prototype cell, select the Table View. Under “Attributes Inspector”, change the “Pr  
“1”. As soon as you change the value, Xcode automatically shows you a prototype cell. In ord  
let's also change the “Style” option from “Plain” to “Group”.



## Want to Learn Swift?

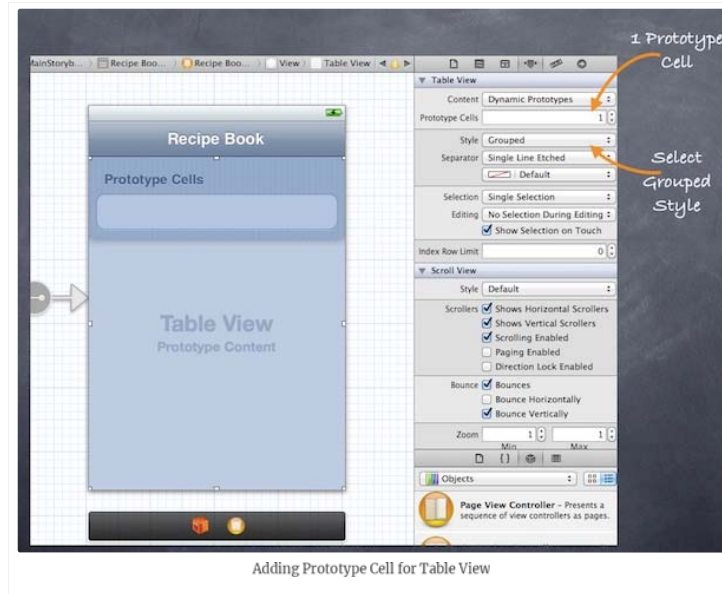
Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

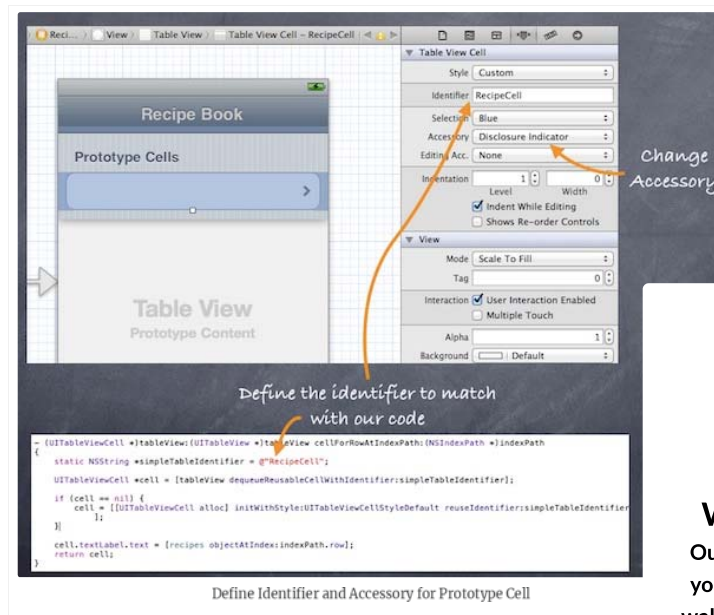
This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

- 34 Shares
- f 1
- t 8
- g+ 19
- in 6



Next, select the “Prototype Cell”. You should be able to customize the options of the cell. To display a disclosure indication for each cell, change the “Accessory” to “Disclosure Indicator”. It’s important to define the Reuse identifier. You can think of this identifier as the cell’s ID. We can use it to refer to a particular prototype cell. Here, we define the identifier as “RecipeCell” that matches with our code.



Now, run the app again. It looks a bit difference and we’re making progress. We’ve changed and added the disclosure indicator.



**Want to Learn Swift?**  
 Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.  
**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

34 Shares



1



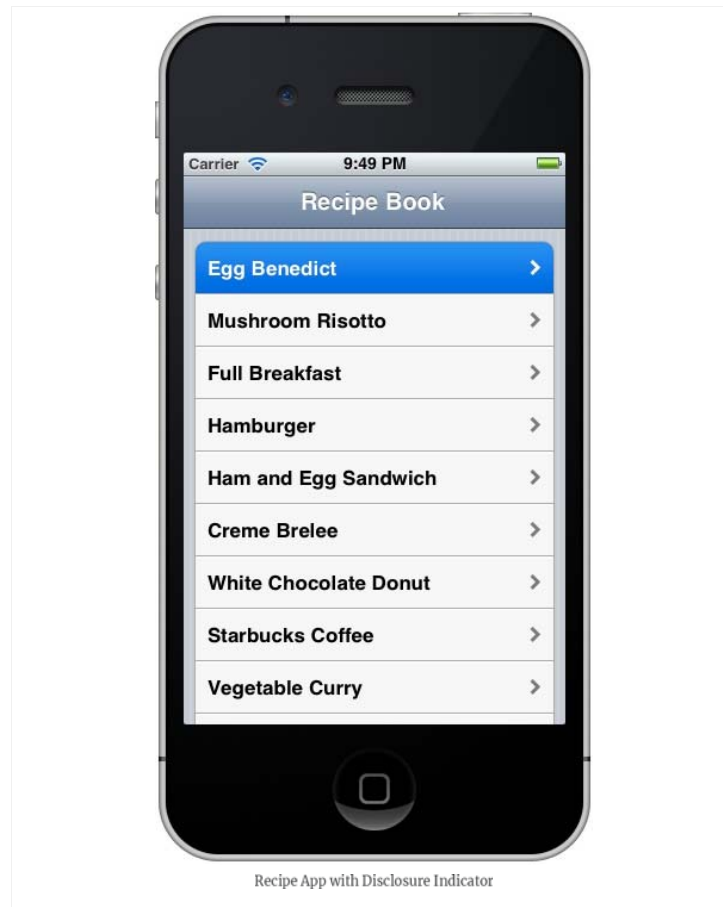
8



19



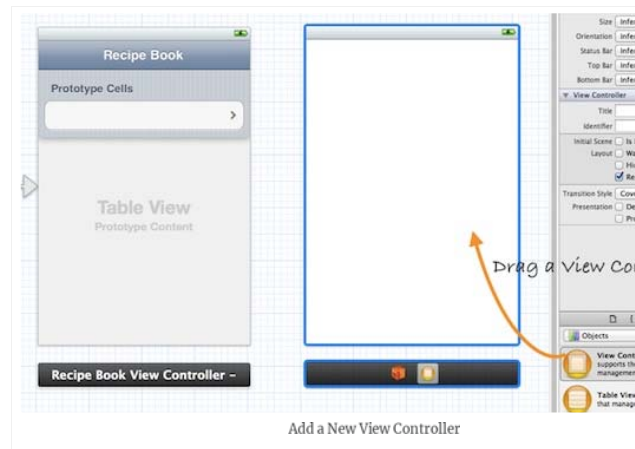
6



## Adding Detail View Controller

Finally it comes to the last part of the tutorial. What's missing is the detail view controller that shows the details of recipe. The detail view controller should be displayed when user taps on any of the recipes.

Okay, let's add a new View Controller as the detail view controller.



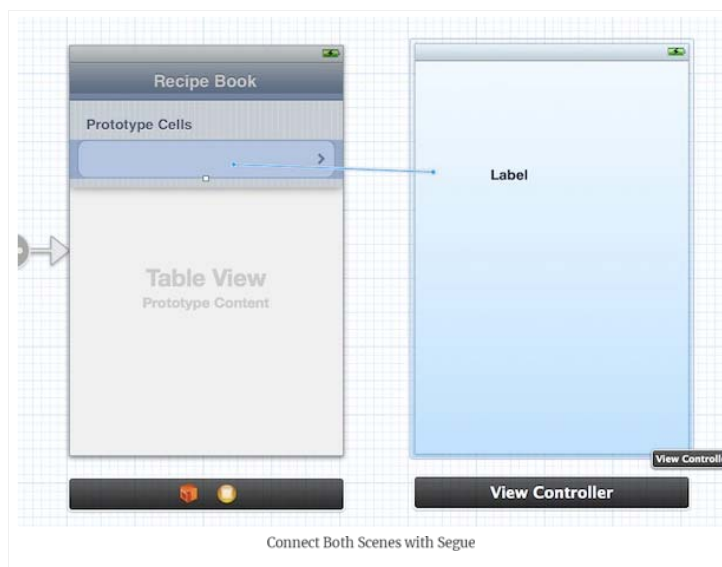
**Want to Learn Swift?**  
 Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.  
**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

The primary purpose of this tutorial is to show you how to implement navigation controller as possible. Let's just a label displaying the recipe name. Drag the label from Object library a You may change the font size or type to make the label look better.

Next, we'll add a segue to connect the prototype cell and the new View Controller. It's very straightforward to add a segue object. Press and hold the control key, click on the prototype cell and drag to the View Controller.

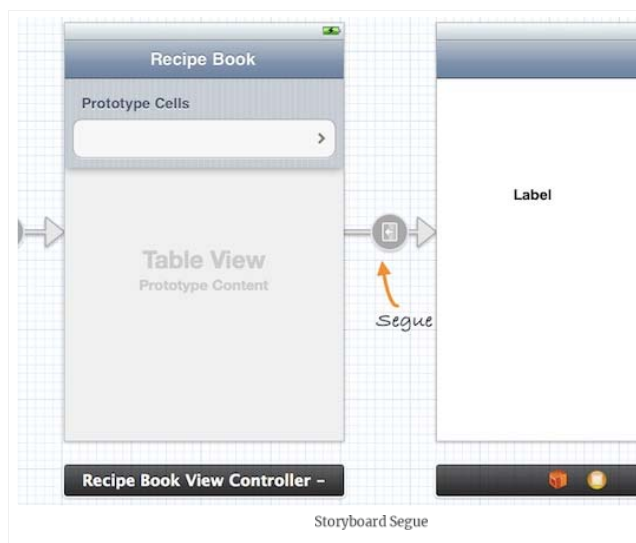


34  
Shares  
f  
1  
t  
8  
g+  
19  
in  
6

Release both buttons and a pop-up shows three types of Segues (push, modal and custom).



As explained before, segue defines the type of transition between scenes. For standard navigation, we use "Push". Once chosen, Xcode automatically connects both scenes with Push segue. Your screen should look like this:



**Want to Learn Swift?**  
Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.  
**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

Now, let's run the app again. As you select any of the recipes, the app shows the detail view controller just shows a label, you already make the navigation work.



- 34 Shares
- f 1
- t 8
- g+ 19
- in 6

## What's Coming Next?

This is a lengthy tutorial and finally it comes to an end. I hope you have a better understanding about Storyboards and know how to design your own navigation controller. However, there is still one thing left: *how can you pass the recipe name from the "Recipe Book View Controller" to the "Detail View Controller"?* I'll leave this to the next tutorial which should be published by the end of this week.

Storyboards, UITableView and Navigation Controller are the fundamental UI elements and commonly used when building iOS apps. So take some time to go through the tutorial and make sure you have a thorough understanding of the material. As always, if you have any questions, leave me comment or ask it at [our forum](#).

**Update #1:** The [next tutorial about data passing](#) is now available!

**Update #2:** You can now [download the full source code](#) for your reference.



- BEGINNER
- IOS 5
- IOS PROGRAMMING
- IOS SDK
- OBJECTIVE C
- STORYBOARDS
- UINavigationController



### Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.



@simonng  
Simon Ng

Founder of AppCoda. Author of multiple iOS programming books including Programming with Swift and Intermediate iOS 11 Programming with Swift. iOS Blogger. Follow me at Facebook, Twitter and Google+.

- 
- 
- 
- 
- 

Send me the eBook

📖 RELATED ARTICLES

SWIFT

Mastering Swift:  
Enumerations, Closures,  
Generics, Protocols and  
High Order Functions

IOS

Customizing Navigation  
Bar and Status Bar in iOS 7

34  
Shares



1



8



19



6



### Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

71 Comments Appcoda 1 Login

Recommend 8 Share Sort by Best

Join the discussion...

LOG IN WITH OR SIGN UP WITH DISQUS Name

34 Shares

- f 1
t 8
g+ 19
in 6

Roy Jay • 5 years ago
Hum...I've done the other tutorials successfully up to this point but when I set this one up, I don't get the Recipe list. I've stopped right after connecting the Table View Controller to the dataSource and the delegate. When I run the app, I still get a blank Recipe Book app. Any ideas? One strange thing was that when I went to connect to TVC to the two Outlets, they appeared to already be connected (almost by default). I've disconnected them and reconnected them a few times and its still not working. Thanks!
18 ^ | v . Reply . Share >

Umair Ali • 5 years ago
I can't drop table view on the recipe book view controller, i try to drop by using zoom out and zoom in to recipe book view controller but i can't do this...whats a problem whith it.???
11 ^ | v . Reply . Share >

James G • 5 years ago
Great tutorial. I got to the very end but how do I create different detail view controllers to correspond with each of the recipes in this tutorial? I can't seem to find that in this tutorial. Thank you!
7 ^ | v . Reply . Share >

Kika • 5 years ago
Hi, I did everithing as you said. App builds without issues, but when I try to scrool table view, app stops working with EXC\_BAD\_ACCESS pointing on this line of thext:
cell.textLabel.text = [recipes objectAtIndex:indexPath.row];
What am I doing wrong?
7 ^ | v . Reply . Share >

Taras -> Kika • 5 years ago
I'm getting the same thing. Must be the iOS6 / iPhone 6.0 sim issue. Classic.
5 ^ | v . Reply . Share >

Darkfire -> Taras • 5 years ago
I am having the same issue.
^ | v . Reply . Share >

Darkfire -> Darkfire • 5 years ago
Nevermind, I redid the tutorial from scratch and it's all working now. Not sure why.
^ | v . Reply . Share >

Marky Boy -> Darkfire • 5 years ago
Hi there i am struggling with this problem and have no clue on how to solve it. has anyone found out how to fix it or what the cause of it is ? i would rather learn how to fix it than just restart the tutorial. Btw these are Ace tutorials :-)

eddy -> Marky Boy • 5 years ago
make sure to use automatic reference counting
^ | v . Reply . Share >

Alex • 5 years ago
Hi guys, everything works fine until I press on a recipe, then nothing happens, It doesnt redirect me to the DetailedViewController... and yes Ive connected the cell to the ViewController and chose Push... :(
6 ^ | v . Reply . Share >

Frank -> Alex • 4 years ago
Hey not sure if you've found the solution, but I had the same issue so hopefully this helps anyone else with this problem. Make sure that whatever you name your identifier in the storyboard for the prototype cells matches the variable you



Want to Learn Swift?
Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.
DOWNLOAD NOW

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook

34  
Shares



1



8



19



6



### Want to Learn Swift?

Our free 150-page eBook will help you learn the basics of Swift 4 and walk you through building your first iOS app using Xcode 9. No prior programming experience is required.

**DOWNLOAD NOW**

This FREE information will help you learn Swift and iOS programming, plus receive programming tips and freebies.

Send me the eBook