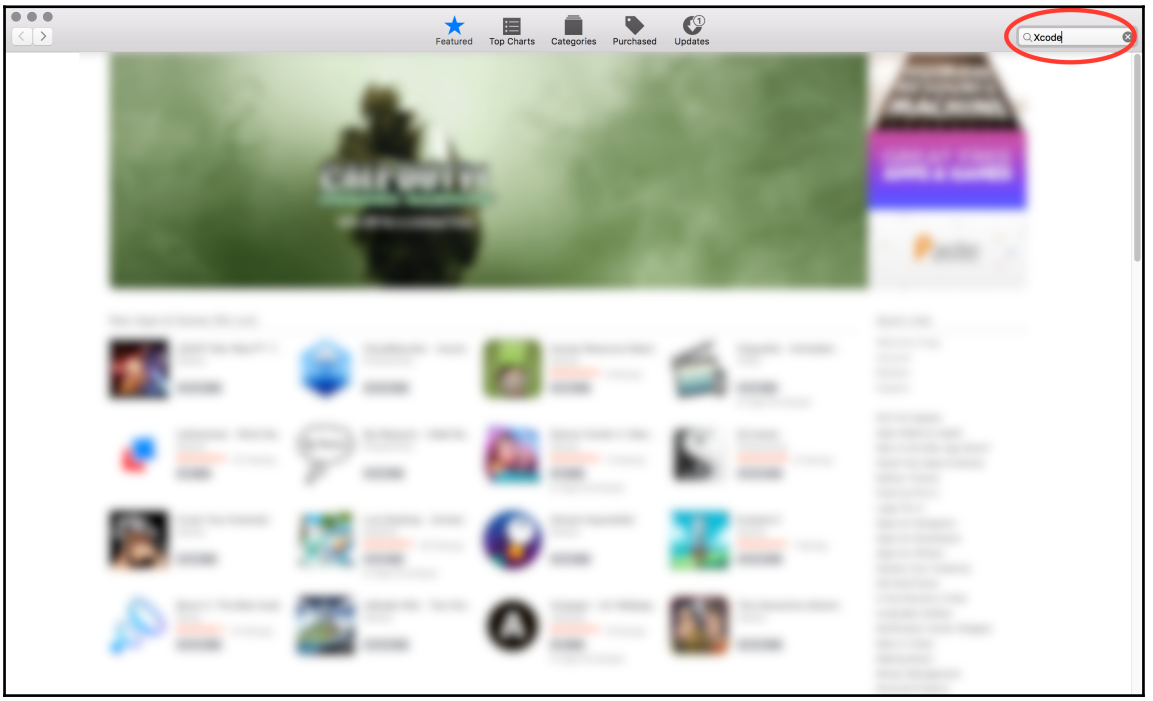
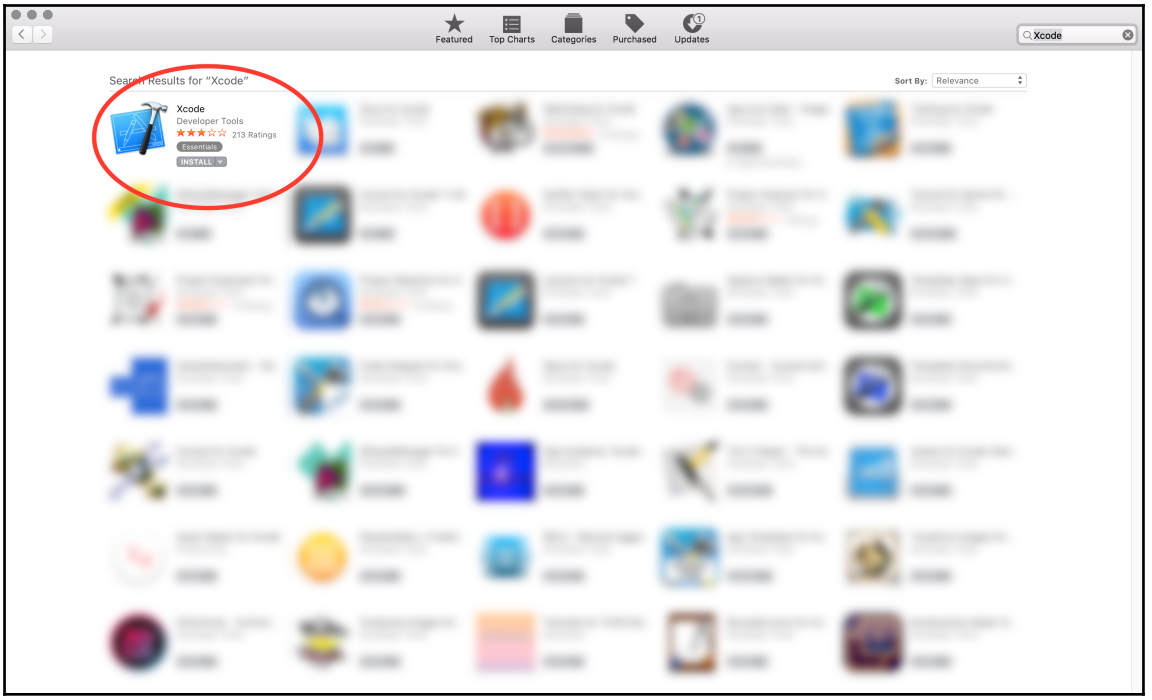
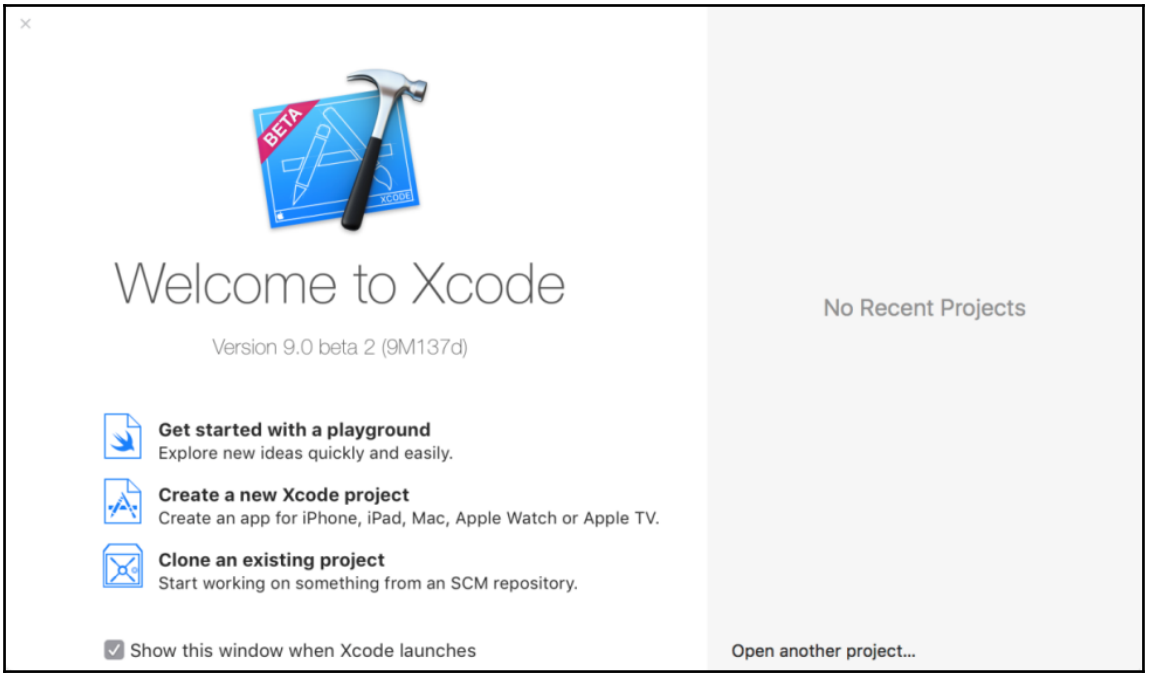
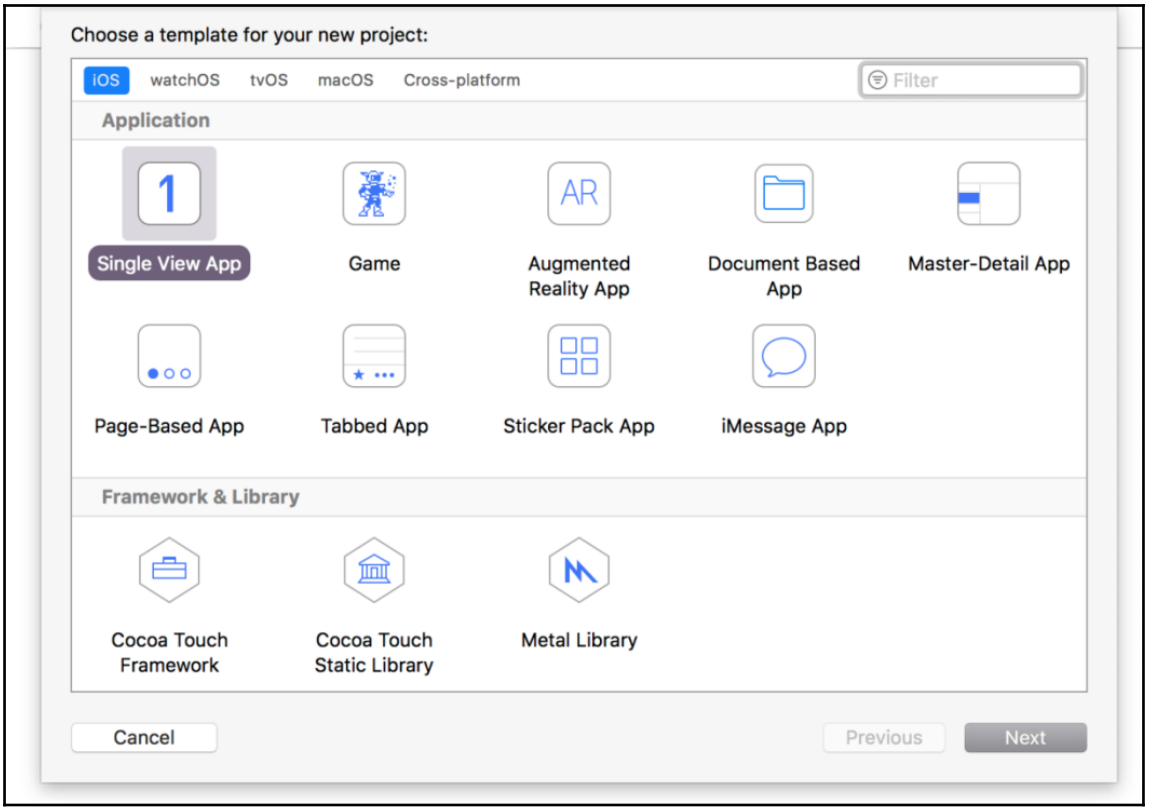
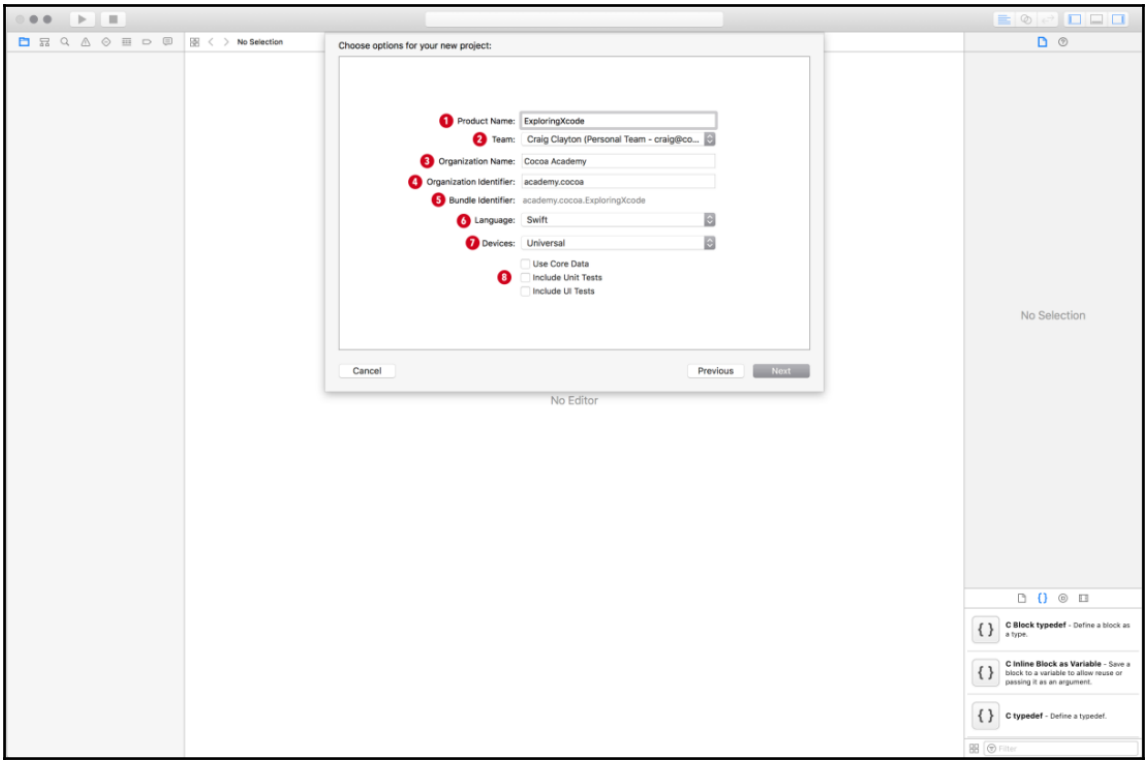
Chapter 1: Getting Familiar with Xcode

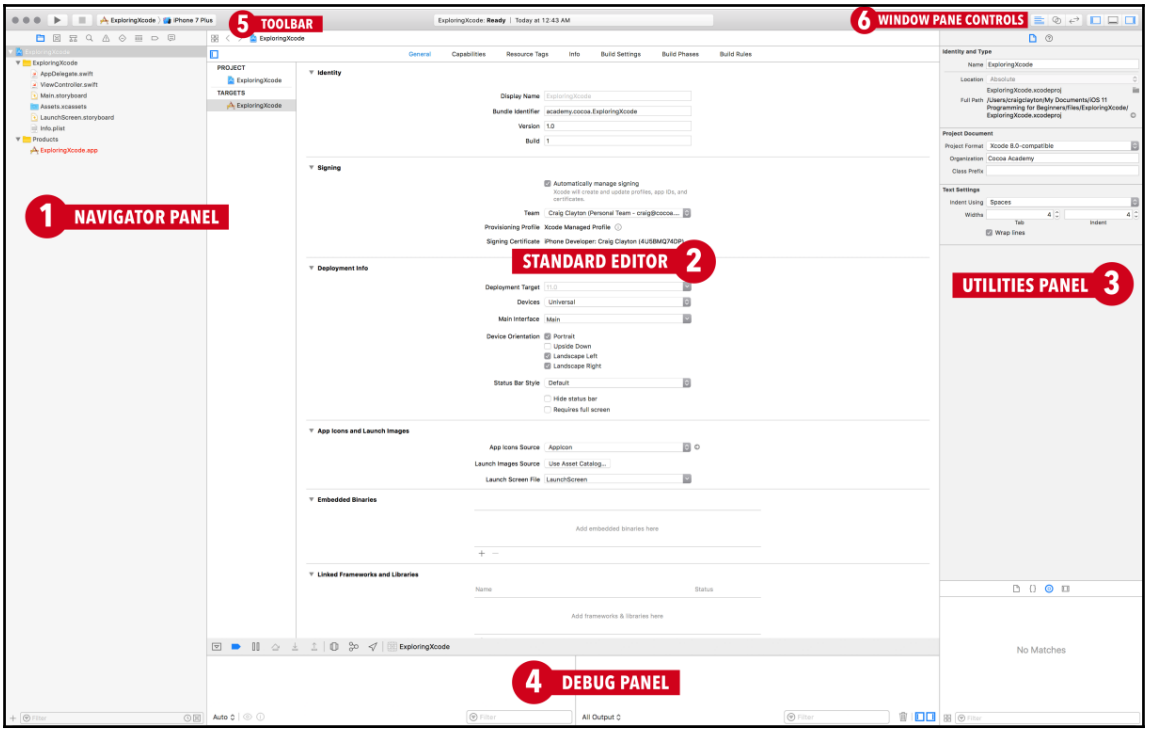


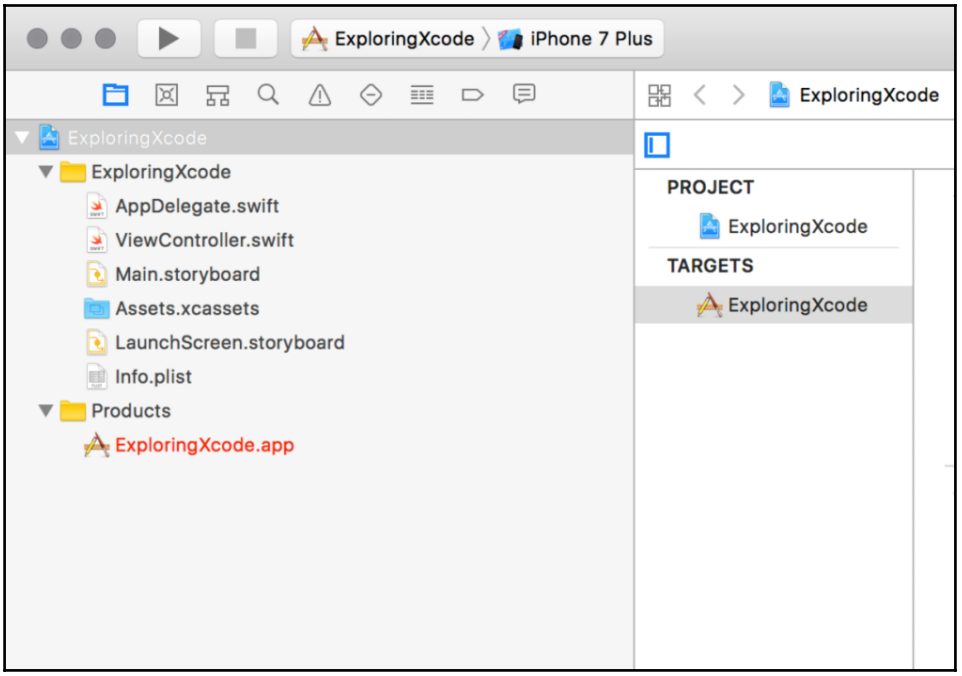

Featured
Top Charts
Categories
Purchased
Updates
Xcode
Search Results for "Xcode"
Sort By:
Relevance
Xcode
Developer Tools
213 Ratings
Essentials
INSTALL


Welcome to Xcode
Version 9.0 beta 2 (9M137d)
Get started with a playground
Explore new ideas quickly and easily.
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.
Clone an existing project
Start working on something from an SCM repository.
Show this window when Xcode launches
No Recent Projects
Open another project...

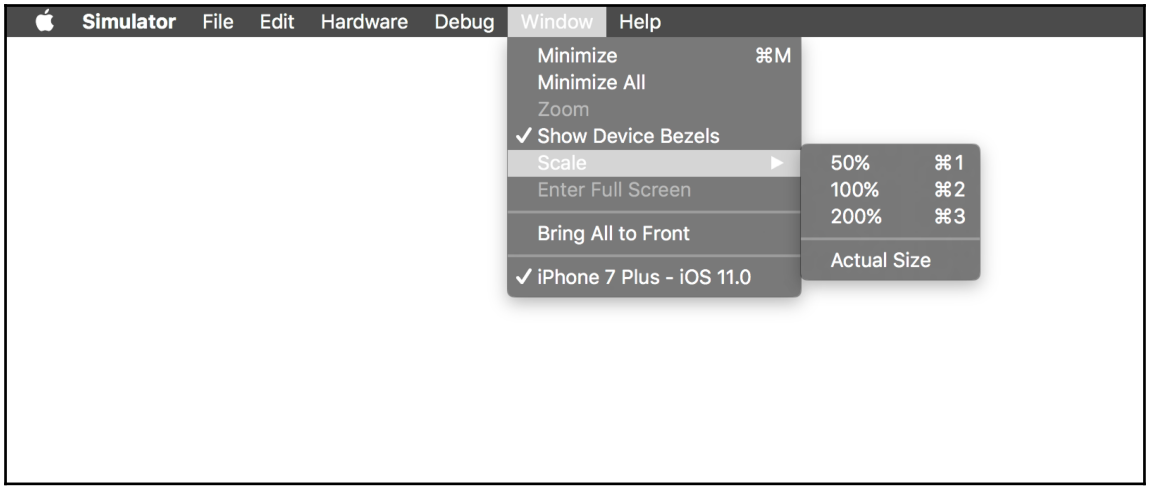

Choose a template for your new project:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application
Single View App
Game
Augmented Reality App
Document Based App
Master-Detail App
Page-Based App
Tabbed App
Sticker Pack App
iMessage App
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Metal Library
Cancel
Previous
Next

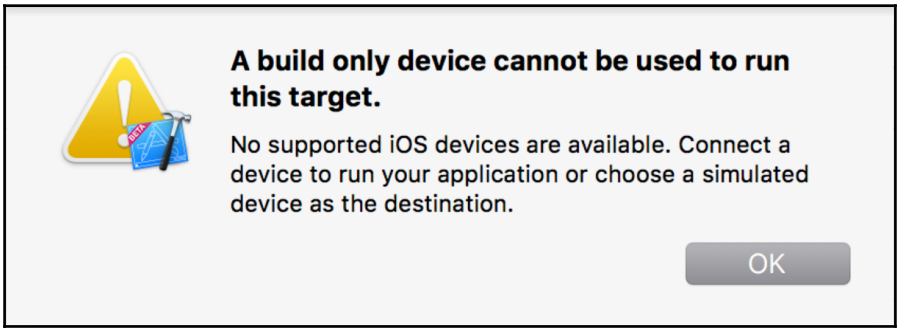

No Selection
Choose options for your new project:
1 Product Name: ExploringXcode
2 Team: Craig Clayton (Personal Team - craig@co...
3 Organization Name: Cocoa Academy
4 Organization Identifier: academy.cocoa
5 Bundle Identifier: academy.cocoa.ExploringXcode
6 Language: Swift
7 Devices: Universal
8 Use Core Data
Include Unit Tests
Include UI Tests
Cancel
Previous
Next
No Editor
No Selection
C Block typedef - Define a block as a type.
C Inline Block as Variable - Save a block to a variable to allow reuse or passing it as an argument.
C typedef - Define a typedef.


5 TOOLBAR
6 WINDOW PANE CONTROLS
1 NAVIGATOR PANEL
STANDARD EDITOR 2
UTILITIES PANEL 3
4 DEBUG PANEL

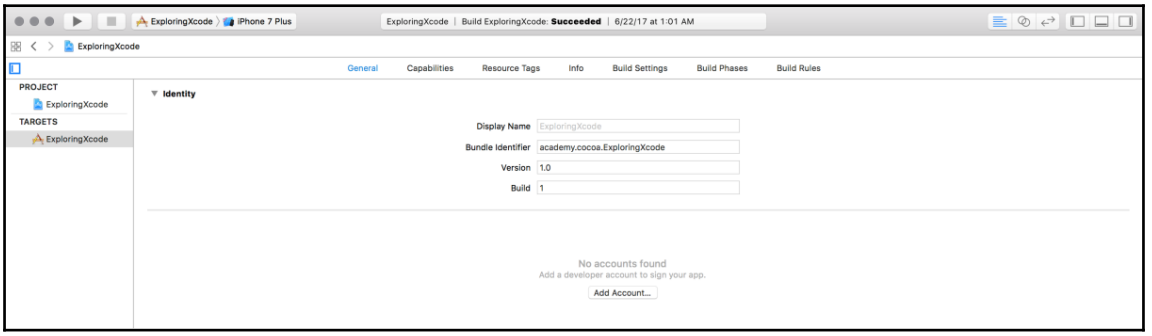

ExploringXcode
iPhone 7 Plus
ExploringXcode
ExploringXcode
ExploringXcode
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
ExploringXcode.app
PROJECT
ExploringXcode
TARGETS
ExploringXcode

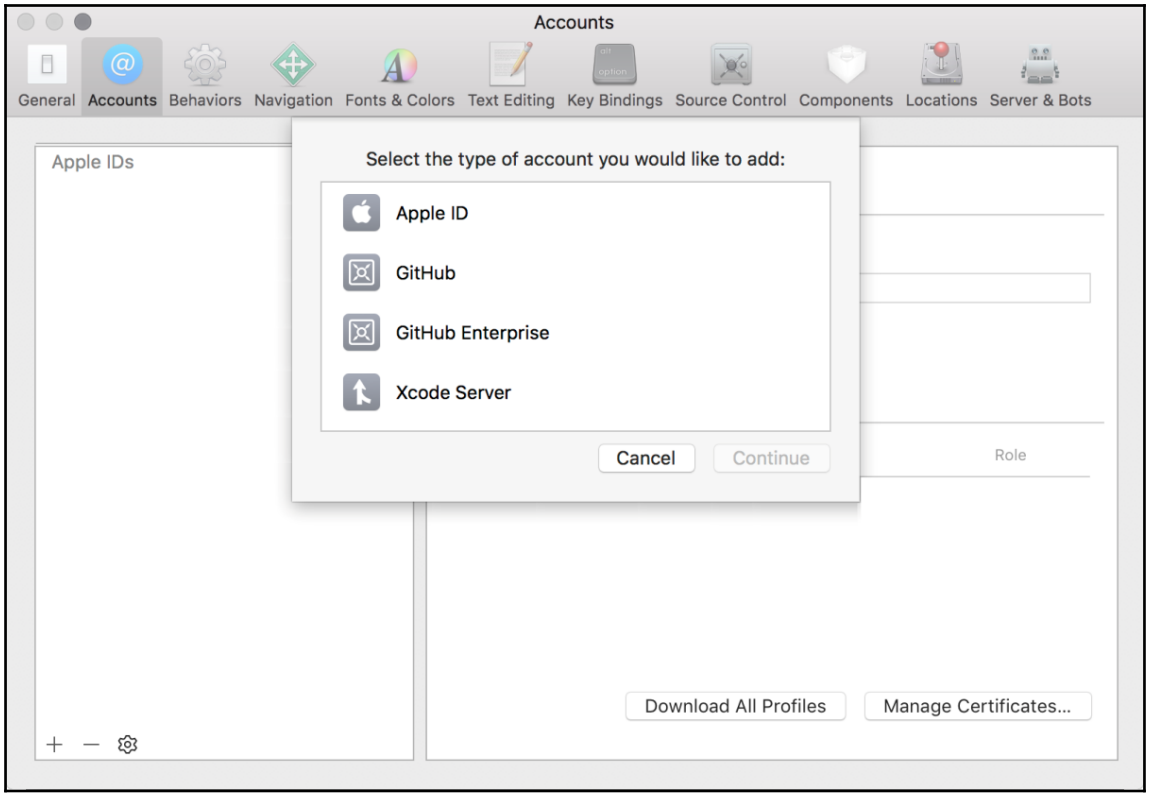

Device
No devices connected to 'My Mac'...
Build Only Device
Generic iOS Device
iOS Simulators
iPad (5th generation)
iPad Air
iPad Air 2
iPad Pro (9.7 inch)
iPad Pro (10.5-inch)
iPad Pro (12.9 inch)
iPad Pro (12.9-inch) (2nd generation)
iPhone 5s
iPhone 6
iPhone 6 Plus
iPhone 6s
iPhone 6s Plus
iPhone 7
iPhone 7 Plus
iPhone SE
Add Additional Simulators...
Download Simulators...

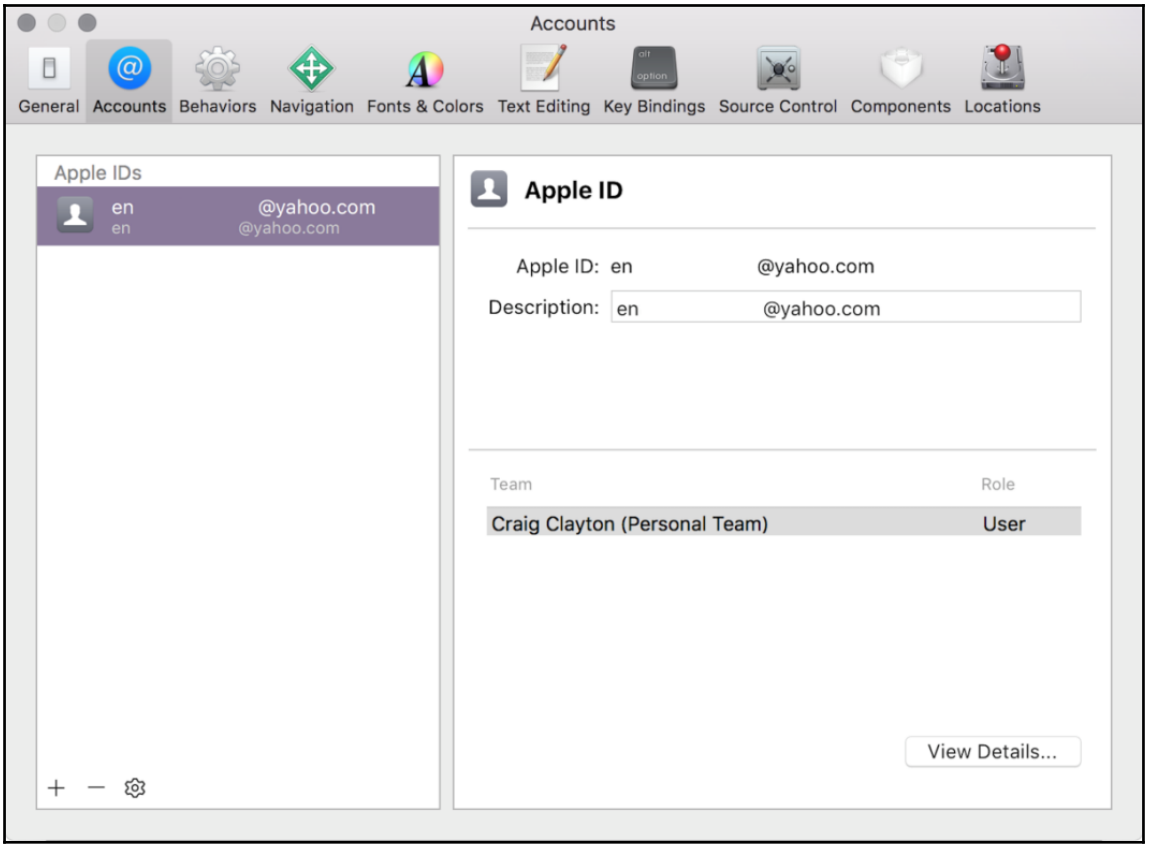

Simulator
File
Edit
Hardware
Debug
Window
Help
Minimize ⌘M
Minimize All
Zoom
✓ Show Device Bezels
Scale
Enter Full Screen
Bring All to Front
✓ iPhone 7 Plus - iOS 11.0
50% ⌘1
100% ⌘2
200% ⌘3
Actual Size

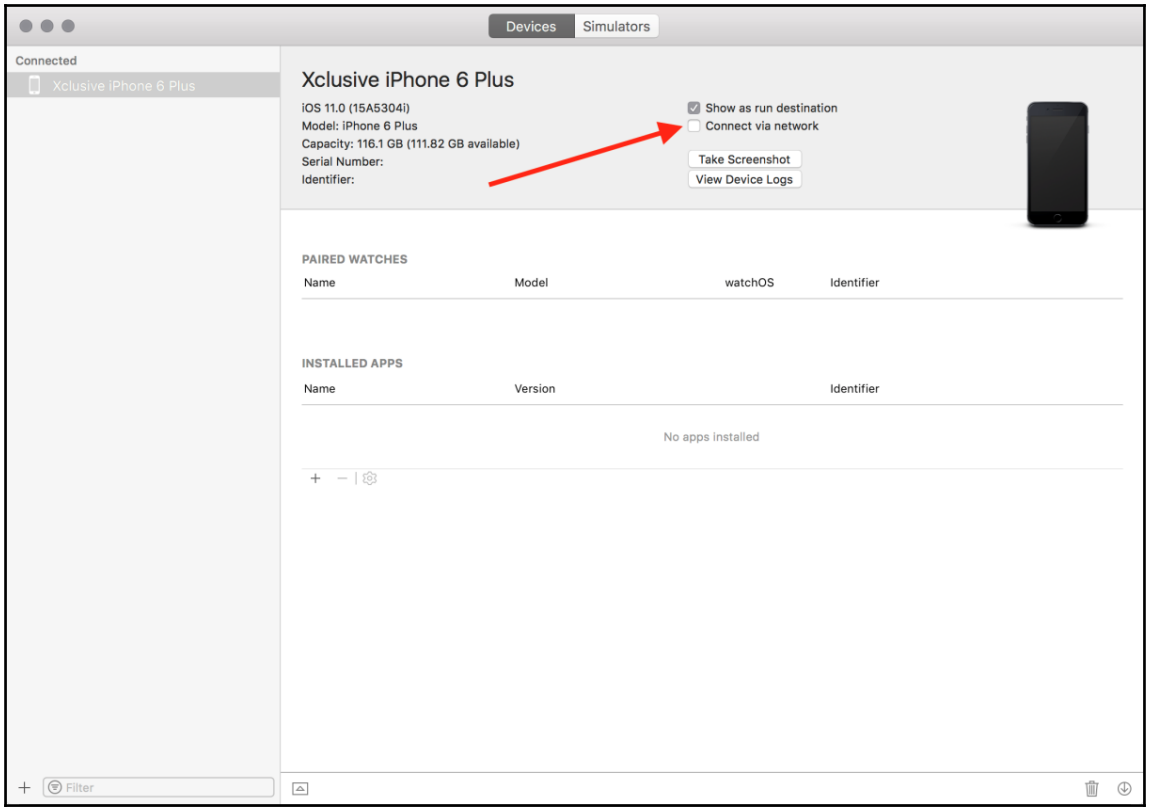

A build only device cannot be used to run this target.
No supported iOS devices are available. Connect a device to run your application or choose a simulated device as the destination.
OK

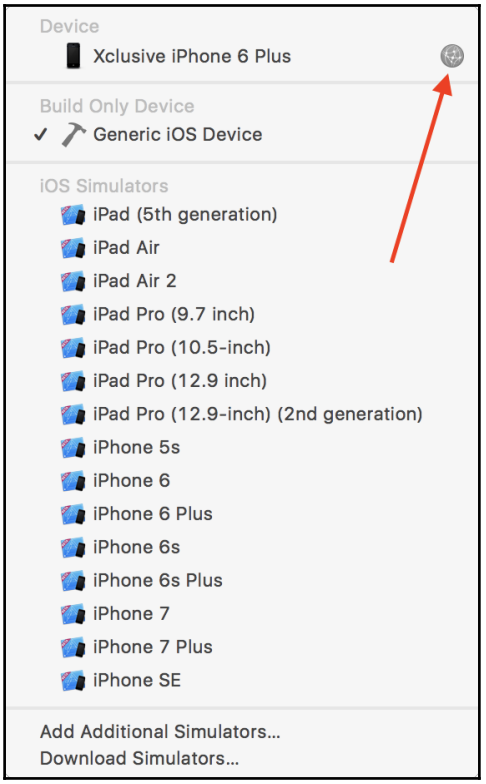

Device
Xclusive iPhone 6 Plus
Build Only Device
Generic iOS Device
iOS Simulators
iPad (5th generation)
iPad Air
iPad Air 2
iPad Pro (9.7 inch)
iPad Pro (10.5-inch)
iPad Pro (12.9 inch)
iPad Pro (12.9-inch) (2nd generation)
iPhone 5s
iPhone 6
iPhone 6 Plus
iPhone 6s
iPhone 6s Plus
iPhone 7
iPhone 7 Plus
iPhone SE
Add Additional Simulators...
Download Simulators...

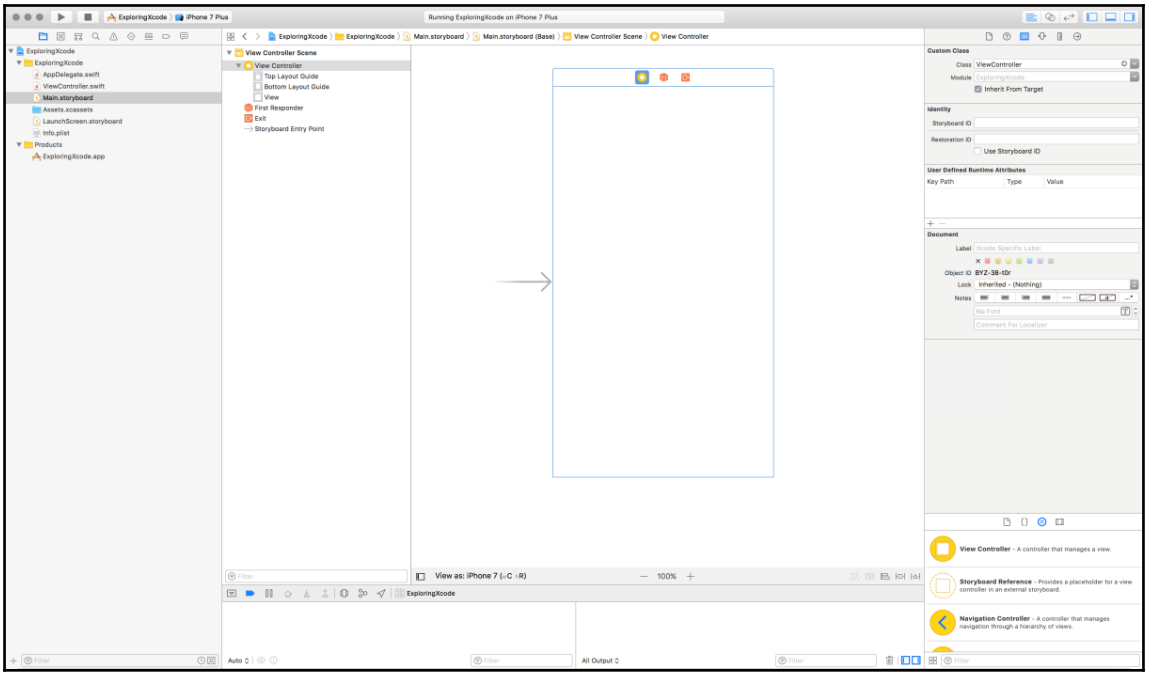

ExploringXcode
iPhone 7 Plus
ExploringXcode | Build ExploringXcode: Succeeded | 6/22/17 at 1:01 AM
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
ExploringXcode
TARGETS
ExploringXcode
Identity
Display Name
Bundle Identifier
academy.cocoa.ExploringXcode
Version
1.0
Build
1
No accounts found
Add a developer account to sign your app.
Add Account...

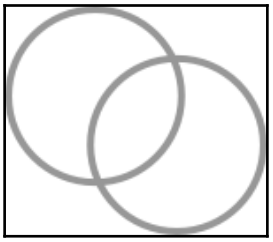

Accounts
General
Accounts
Behaviors
Navigation
Fonts & Colors
Text Editing
Key Bindings
Source Control
Components
Locations
Server & Bots
Apple IDs
Select the type of account you would like to add:
Apple ID
GitHub
GitHub Enterprise
Xcode Server
Cancel
Continue
Role
Download All Profiles
Manage Certificates...

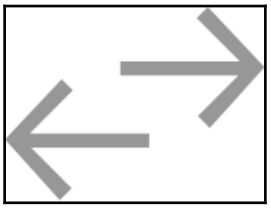

Accounts
General
Accounts
Behaviors
Navigation
Fonts & Colors
Text Editing
Key Bindings
Source Control
Components
Locations
Apple IDs
en @yahoo.com
en @yahoo.com
Apple ID
Apple ID: en @yahoo.com
Description: en @yahoo.com
Team
Role
Craig Clayton (Personal Team)
User
View Details...

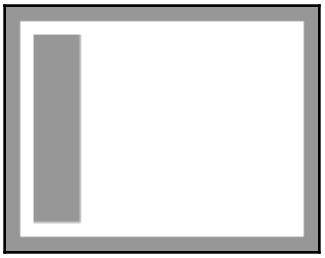

Devices
Simulators
Connected
Xclusive iPhone 6 Plus
Xclusive iPhone 6 Plus
iOS 11.0 (15A5304i)
Model: iPhone 6 Plus
Capacity: 116.1 GB (111.82 GB available)
Serial Number:
Identifier:
Show as run destination
Connect via network
Take Screenshot
View Device Logs
PAIRED WATCHES
Name
Model
watchOS
Identifier
INSTALLED APPS
Name
Version
Identifier
No apps installed
Filter


Device
Xclusive iPhone 6 Plus
Build Only Device
Generic iOS Device
iOS Simulators
iPad (5th generation)
iPad Air
iPad Air 2
iPad Pro (9.7 inch)
iPad Pro (10.5-inch)
iPad Pro (12.9 inch)
iPad Pro (12.9-inch) (2nd generation)
iPhone 5s
iPhone 6
iPhone 6 Plus
iPhone 6s
iPhone 6s Plus
iPhone 7
iPhone 7 Plus
iPhone SE
Add Additional Simulators...
Download Simulators...

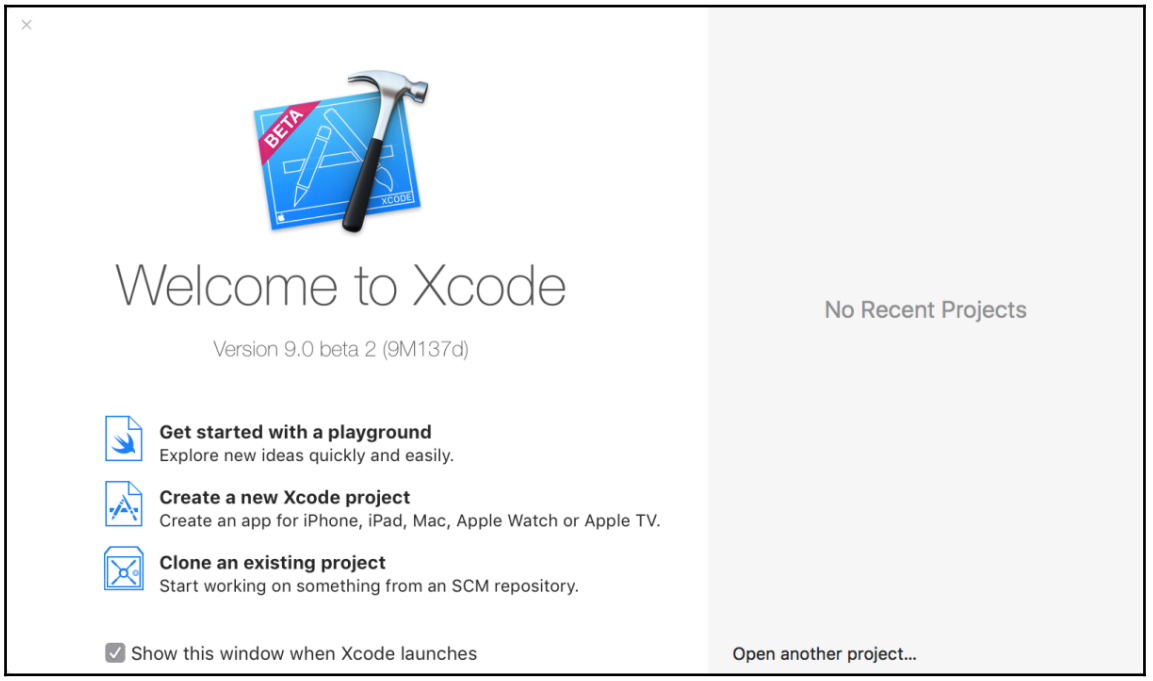


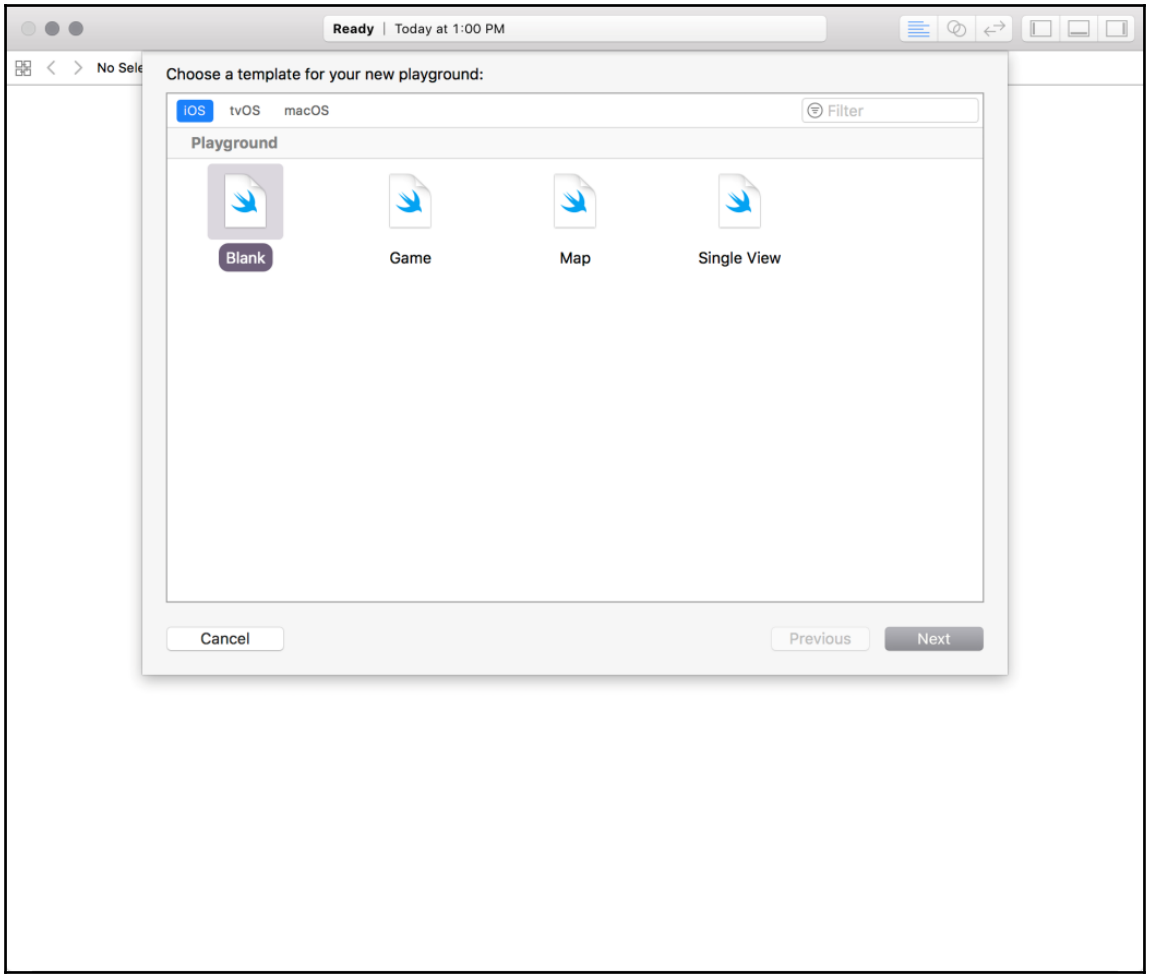


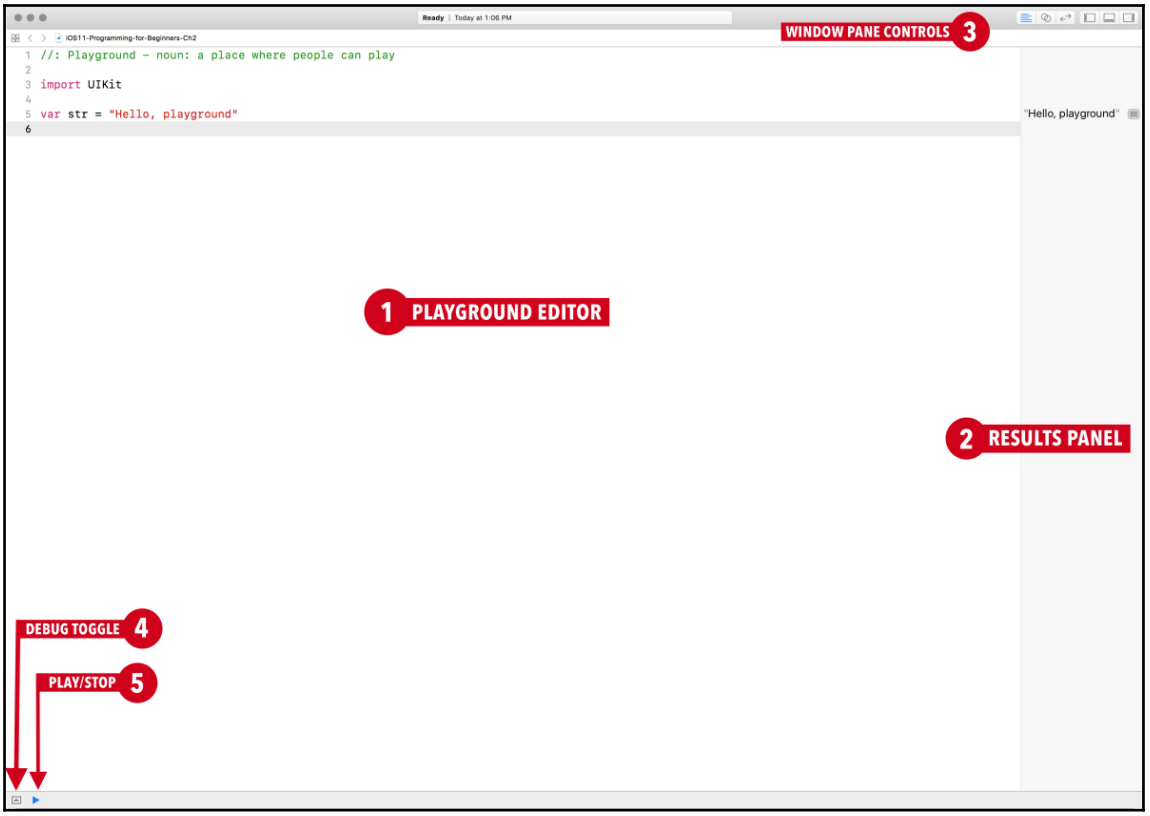




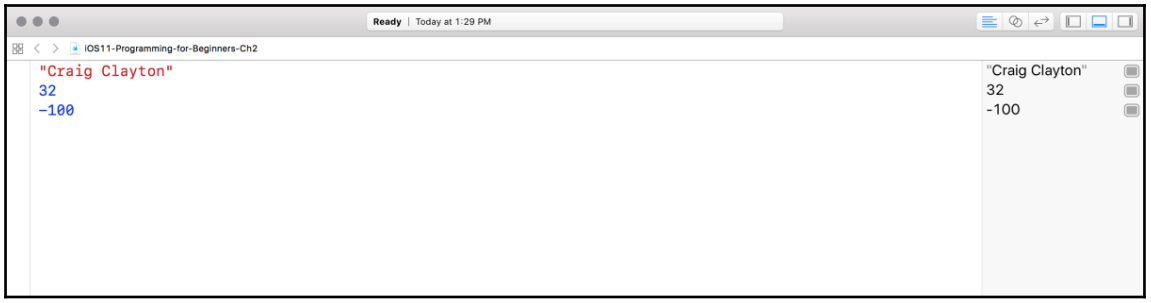
Chapter 2: Building a Foundation with Swift

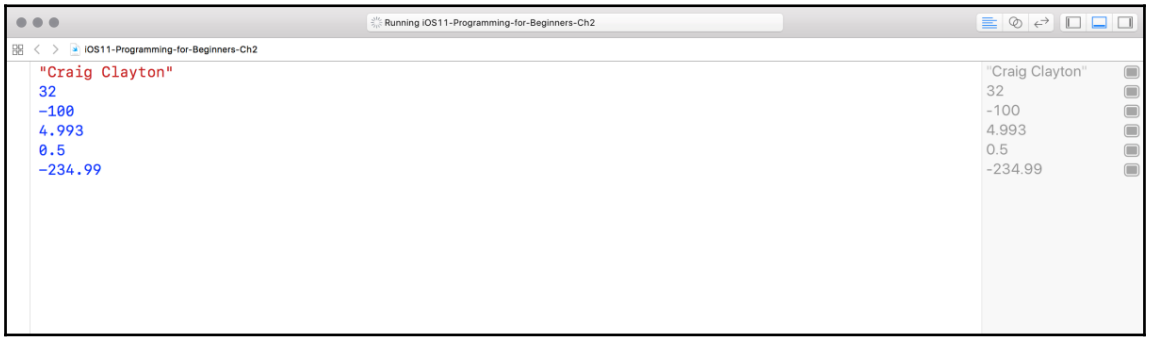



Ready | Today at 1:00 PM
Choose a template for your new playground:
iOS
tvOS
macOS
Filter
Playground
Blank
Game
Map
Single View
Cancel
Previous
Next


Ready | Today at 1:06 PM
iOS11-Programming-for-Beginners-Ch2
WINDOW PANE CONTROLS 3
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
"Hello, playground"
1 PLAYGROUND EDITOR
2 RESULTS PANEL
DEBUG TOGGLE 4
PLAY/STOP 5


Ready | Today at 1:29 PM
iOS11-Programming-for-Beginners-Ch2
"Craig Clayton"
"Craig Clayton"

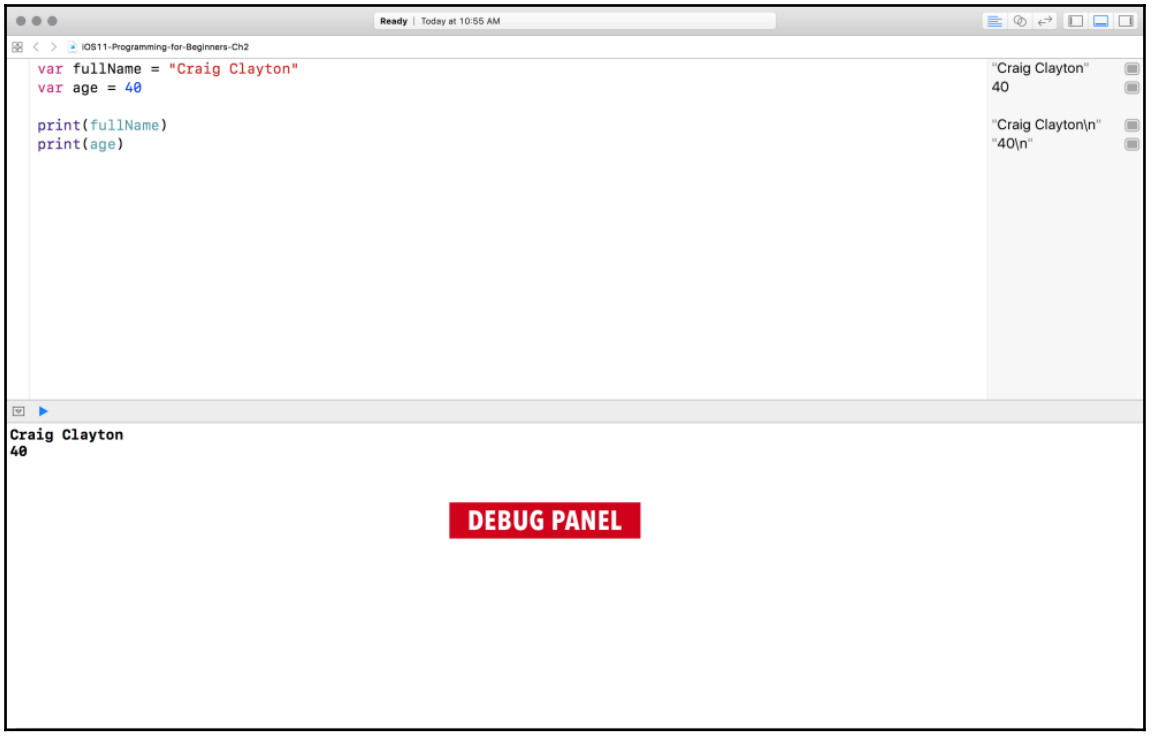

Ready | Today at 1:29 PM
iOS11-Programming-for-Beginners-Ch2
"Craig Clayton"
32
-100
"Craig Clayton"
32
-100

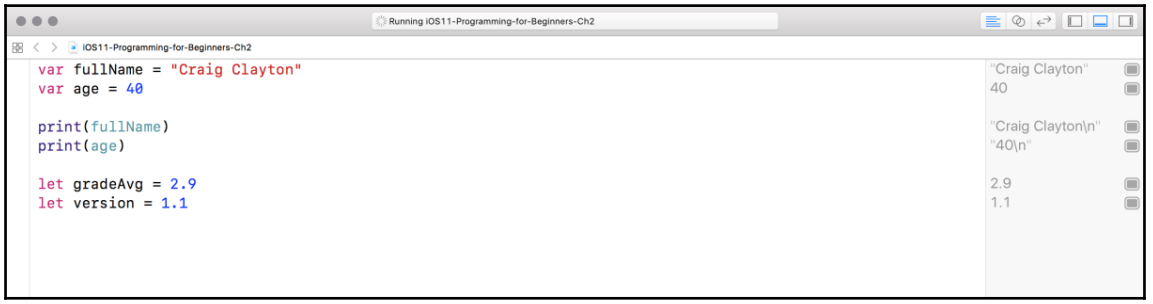

Running iOS11-Programming-for-Beginners-Ch2
iOS11-Programming-for-Beginners-Ch2
"Craig Clayton"
32
-100
4.993
0.5
-234.99
"Craig Clayton"
32
-100
4.993
0.5
-234.99


Ready | Today at 1:30 PM
iOS11-Programming-for-Beginners-Ch2
"Craig Clayton"
32
-100
4.993
0.5
-234.99
true
false
"Craig Clayton"
32
-100
4.993
0.5
-234.99
true
false


Ready | Today at 1:31 PM
iOS11-Programming-for-Beginners-Ch2
var fullName = "Craig Clayton"
"Craig Clayton"

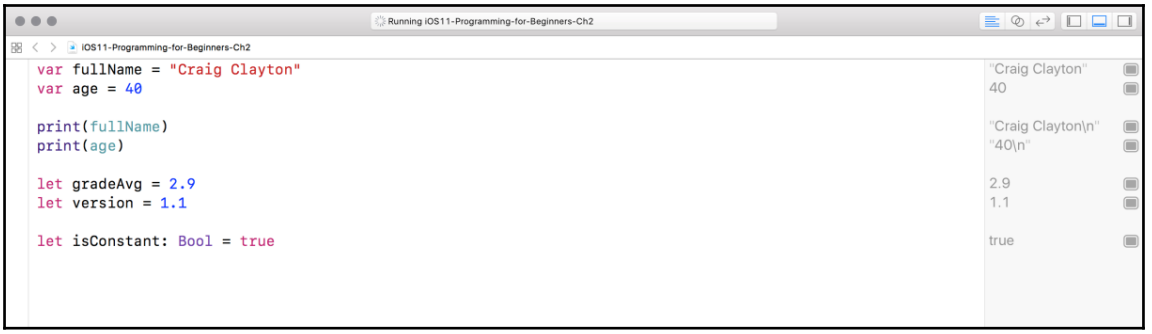

Ready | Today at 1:32 PM
iOS11-Programming-for-Beginners-Ch2
var fullName = "Craig Clayton"
var age = 40
"Craig Clayton"
40

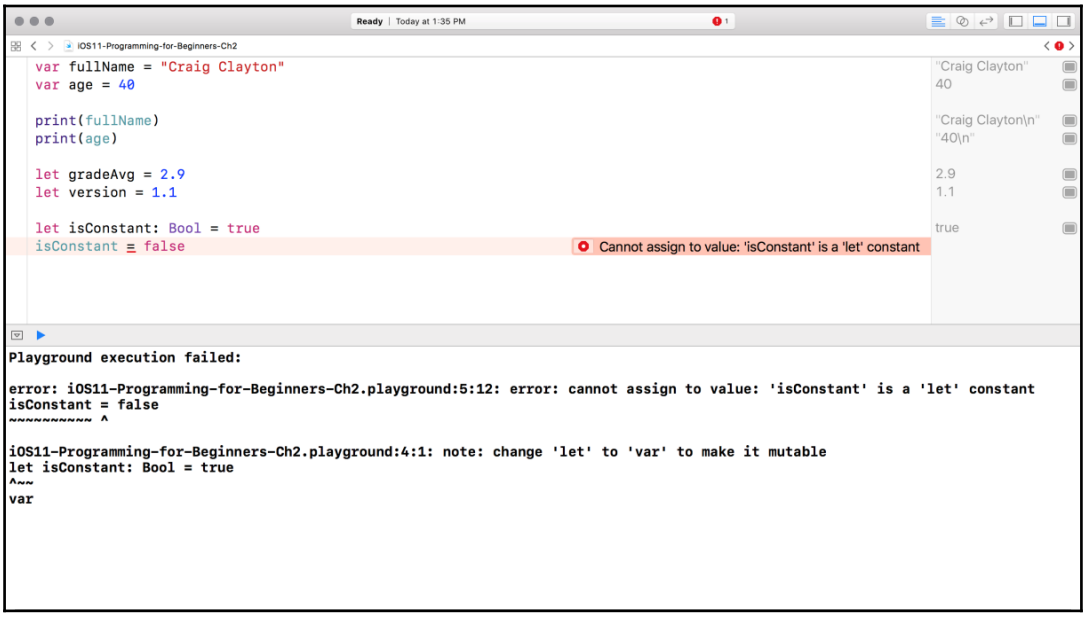


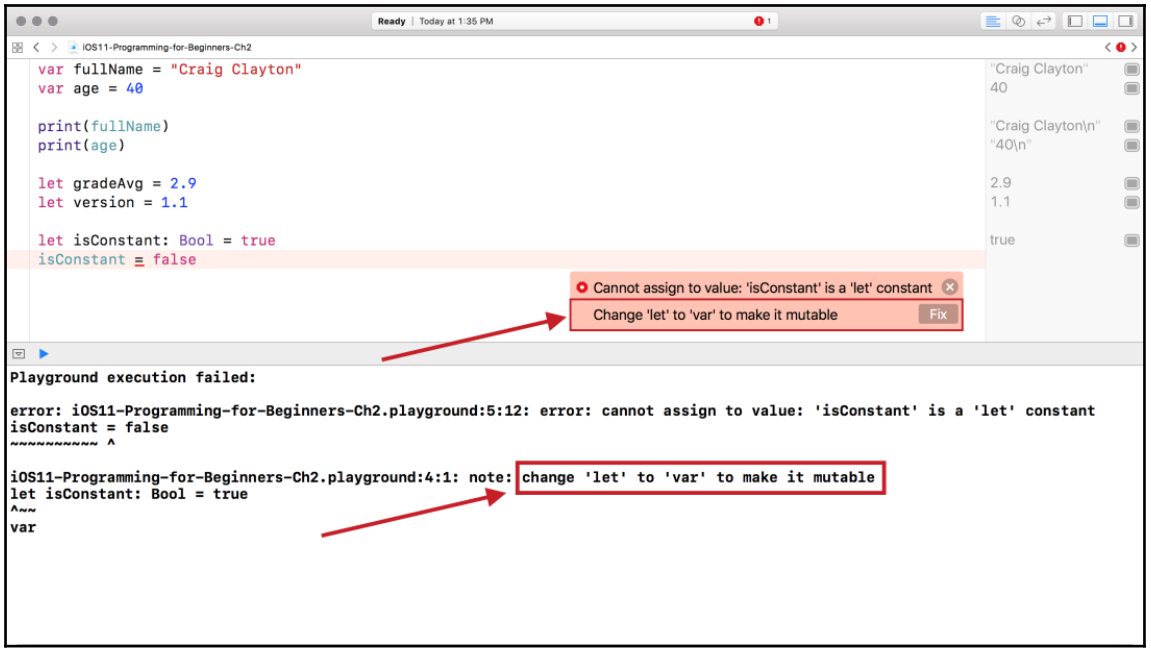


Double vs Float

let lessPrecisePI = Float("3.14")
let morePrecisePI = Float("3.1415926536")


Running iOS11-Programming-for-Beginners-Ch2
iOS11-Programming-for-Beginners-Ch2
var fullName = "Craig Clayton"
var age = 40
print(fullName)
print(age)
let gradeAvg = 2.9
let version = 1.1
let isConstant: Bool = true
"Craig Clayton"
40
"Craig Clayton\n"
"40\n"
2.9
1.1
true


Ready | Today at 1:35 PM
iOS11-Programming-for-Beginners-Ch2
var fullName = "Craig Clayton"
var age = 40
print(fullName)
print(age)
let gradeAvg = 2.9
let version = 1.1
let isConstant: Bool = true
isConstant = false
Cannot assign to value: 'isConstant' is a 'let' constant
"Craig Clayton"
40
"Craig Clayton\n"
"40\n"
2.9
1.1
true
Playground execution failed:
error: iOS11-Programming-for-Beginners-Ch2.playground:5:12: error: cannot assign to value: 'isConstant' is a 'let' constant
isConstant = false
~~~~~~~~~~ ^
iOS11-Programming-for-Beginners-Ch2.playground:4:1: note: change 'let' to 'var' to make it mutable
let isConstant: Bool = true
^~~
var
~~~~~~~~~~


Ready | Today at 1:35 PM
iOS11-Programming-for-Beginners-Ch2
var fullName = "Craig Clayton"
var age = 40
print(fullName)
print(age)
let gradeAvg = 2.9
let version = 1.1
let isConstant: Bool = true
isConstant = false
"Craig Clayton"
40
"Craig Clayton\n"
"40\n"
2.9
1.1
true
Cannot assign to value: 'isConstant' is a 'let' constant
Change 'let' to 'var' to make it mutable
Fix
Playground execution failed:
error: iOS11-Programming-for-Beginners-Ch2.playground:5:12: error: cannot assign to value: 'isConstant' is a 'let' constant
isConstant = false
~~~~~~~~~~ ^
iOS11-Programming-for-Beginners-Ch2.playground:4:1: note: change 'let' to 'var' to make it mutable
let isConstant: Bool = true
^~~
var


Ready | Today at 1:36 PM
iOS11-Programming-for-Beginners-Ch2
print(fullName)
print(age)
let gradeAvg = 2.9
let version = 1.1
let isConstant: Bool = true
// Single line comment
/*
This comment is meant for
multiple lines
*/
"Craig Clayton\n"
"40\n"
2.9
1.1
true
Craig Clayton
40
~~~~~~~~~~


Ready | Today at 4:49 PM
iOS11-Programming-for-Beginners-Ch2
let firstName = "Craig"
let lastName = "Clayton"
"Craig"
"Clayton"


Ready | Today at 4:52 PM
iOS11-Programming-for-Beginners-Ch2
let firstName = "Craig"
let lastName = "Clayton"
let full = "\(firstName) \(lastName)"
"Craig"
"Clayton"
"Craig Clayton"

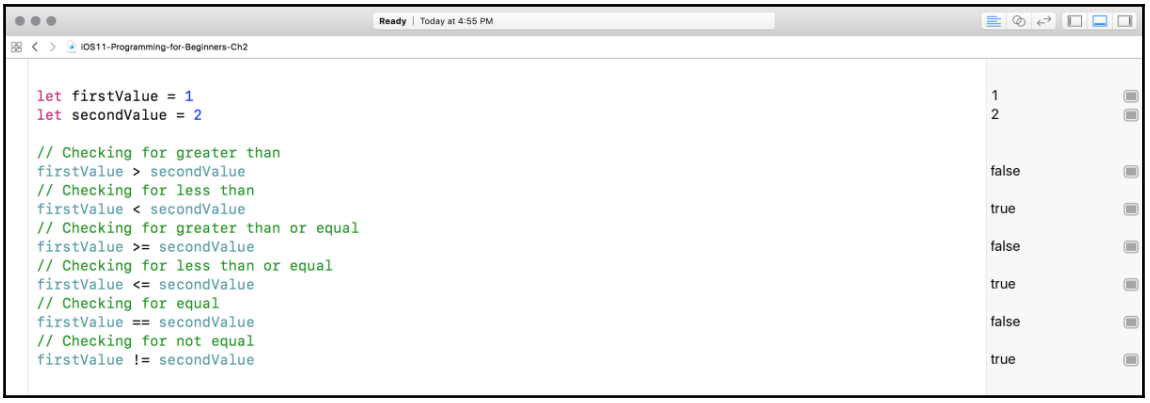

Ready | Today at 4:52 PM
iOS11-Programming-for-Beginners-Ch2
let firstName = "Craig"
let lastName = "Clayton"
let full = "\(firstName) \(lastName)"
print("\(firstName) \(lastName)")
"Craig"
"Clayton"
"Craig Clayton"
"Craig Clayton\n"
Craig Clayton


Ready | Today at 4:53 PM
iOS11-Programming-for-Beginners-Ch2
// (+) operator
let sum = 23 + 20
// (-) operator
let result = 32 - sum
// (*) operator
let total = result * 5
// (/) operator
let divide = total / 10
43
-11
-55
-5

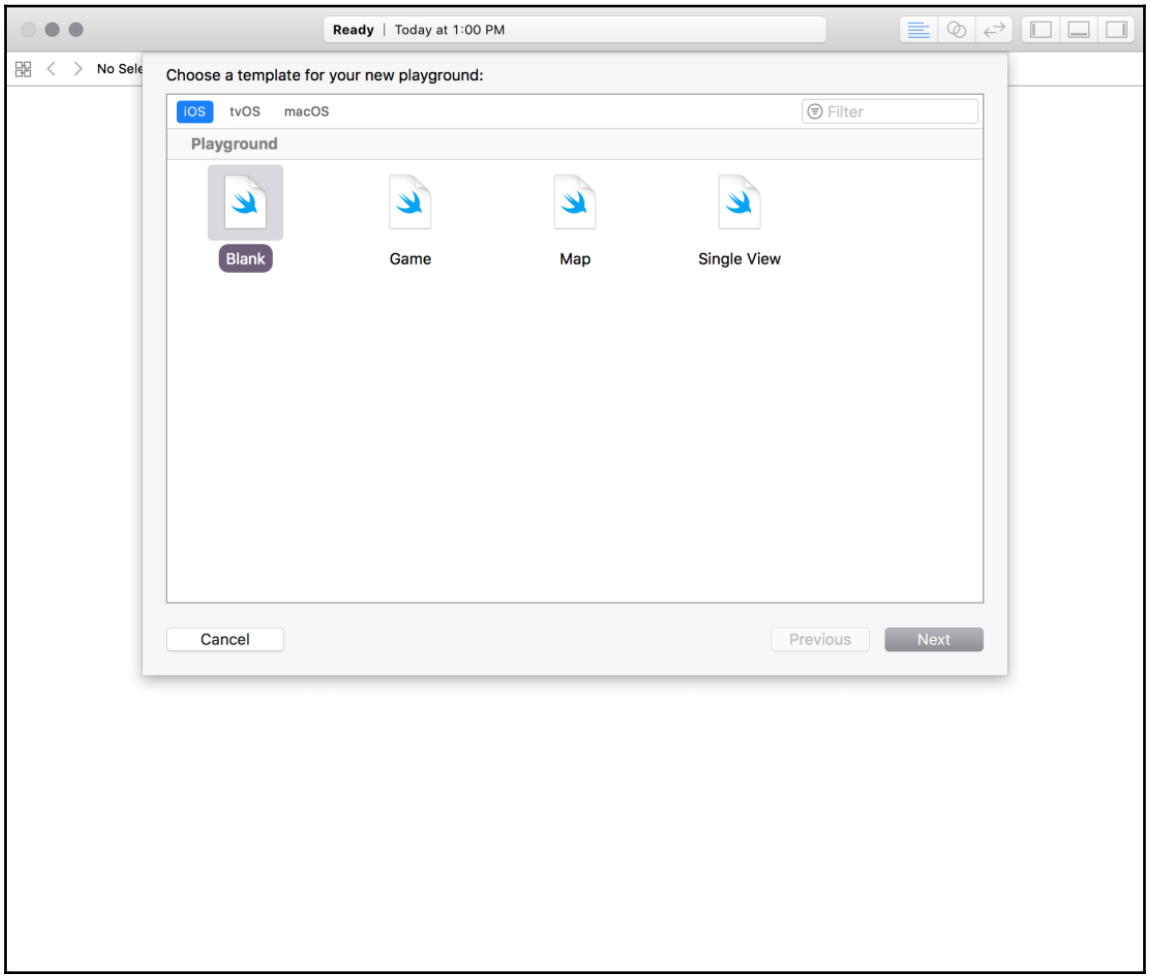

Ready | Today at 4:53 PM
iOS11-Programming-for-Beginners-Ch2
// (+) operator
let sum = 23 + 20
// (-) operator
let result = 32 - sum
// (*) operator
let total = result * 5
// (/) operator
let divide = total / 10
let divide2 = Double(total) / 10
43
-11
-55
-5
-5.5

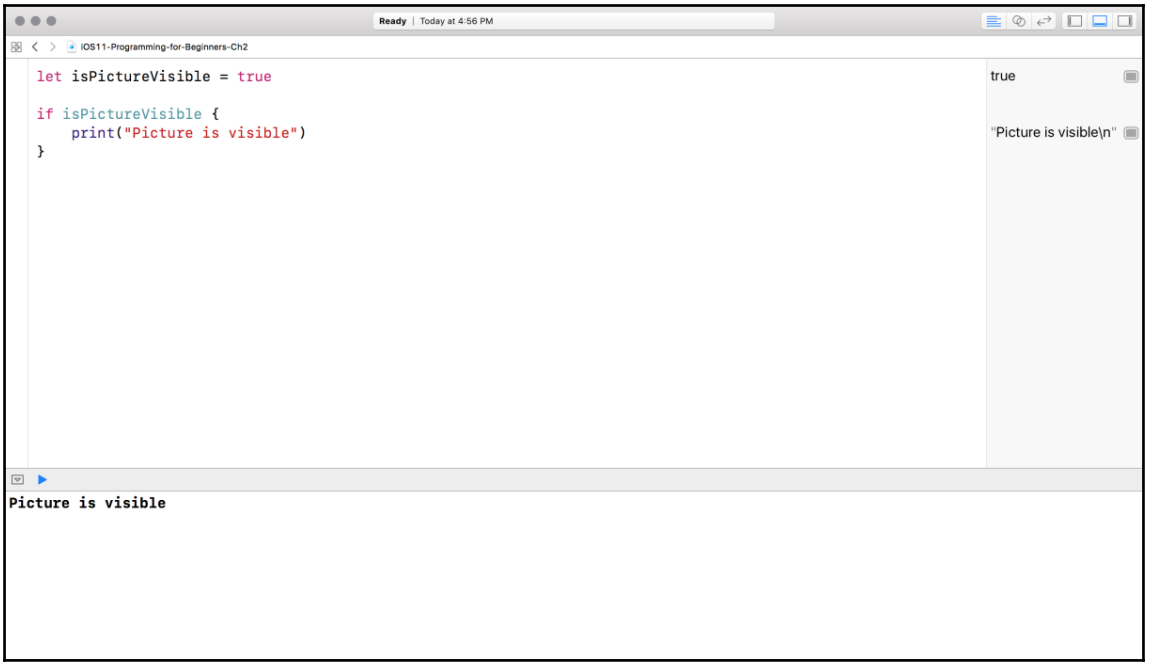

Ready | Today at 4:54 PM
iOS11-Programming-for-Beginners-Ch2
// (+) operator
let sum = 23 + 20
// (-) operator
let result = 32 - sum
// (*) operator
let total = result * 5
// (/) operator
let divide = total / 10
let divide2 = Double(total) / 10
let mod = 7 % 3
43
-11
-55
-5
-5.5
1

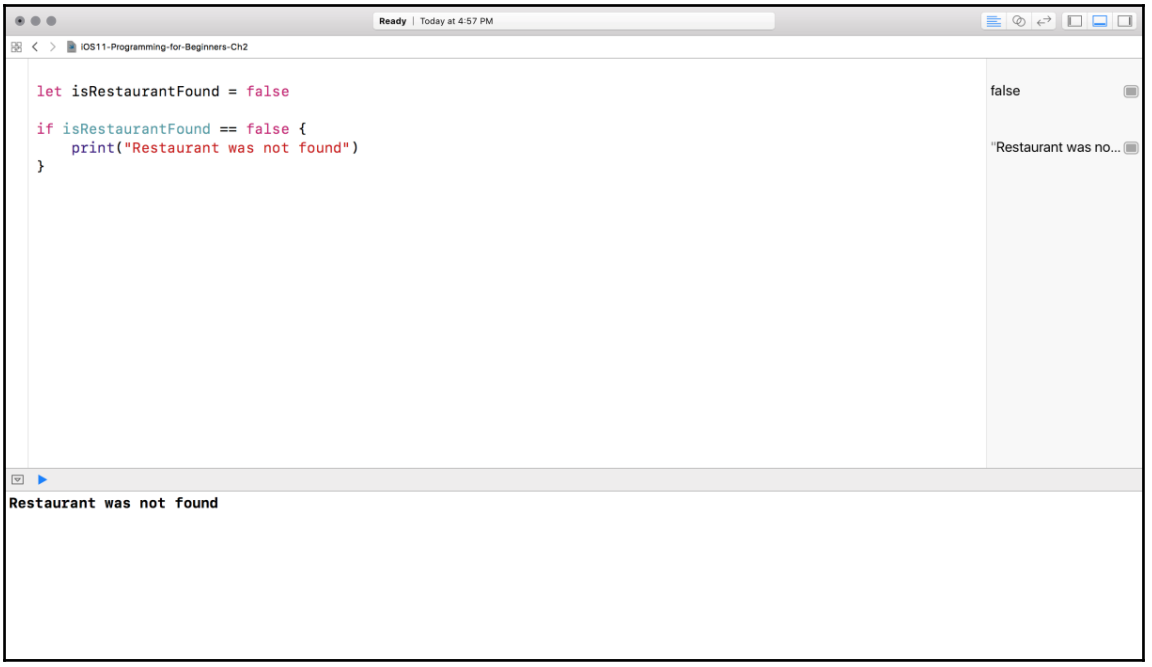

Ready | Today at 4:54 PM
iOS11-Programming-for-Beginners-Ch2
var count = 0
// Option #1
count = count + 1
count = count - 1
// Option #2
count += 1
count -= 1
0
1
0
1
0

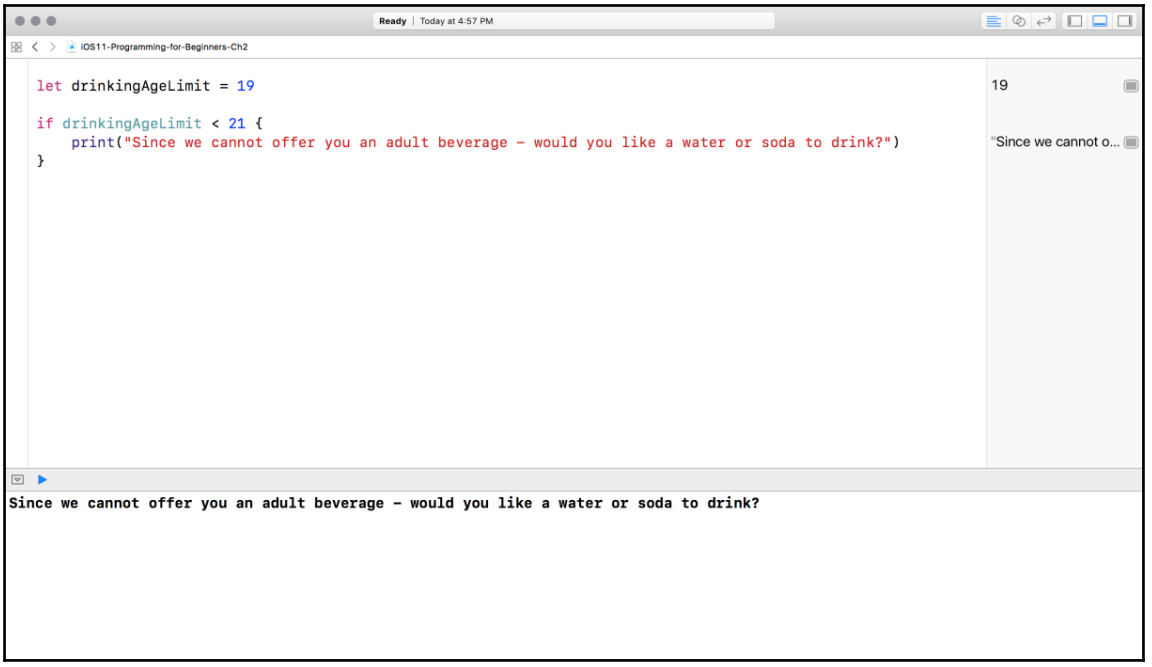

Ready | Today at 4:55 PM
iOS11-Programming-for-Beginners-Ch2
let firstValue = 1
let secondValue = 2
// Checking for greater than
firstValue > secondValue
// Checking for less than
firstValue < secondValue
// Checking for greater than or equal
firstValue >= secondValue
// Checking for less than or equal
firstValue <= secondValue
// Checking for equal
firstValue == secondValue
// Checking for not equal
firstValue != secondValue
1
2
false
true
false
true
false
true

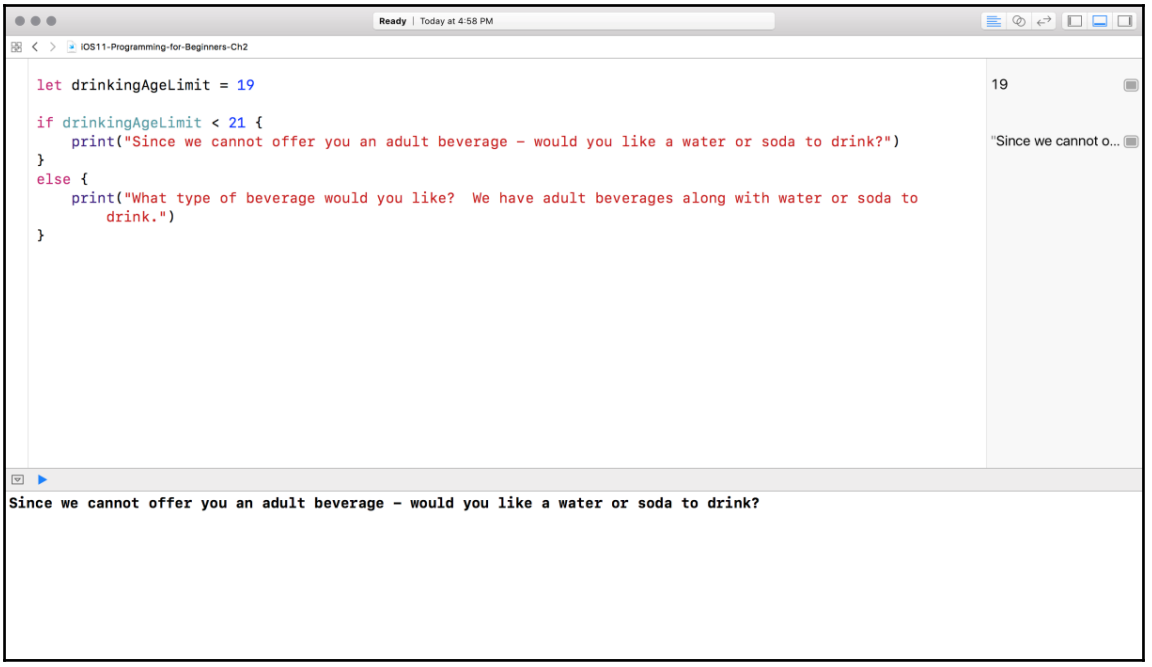
Chapter 3: Building on the Swift Foundation

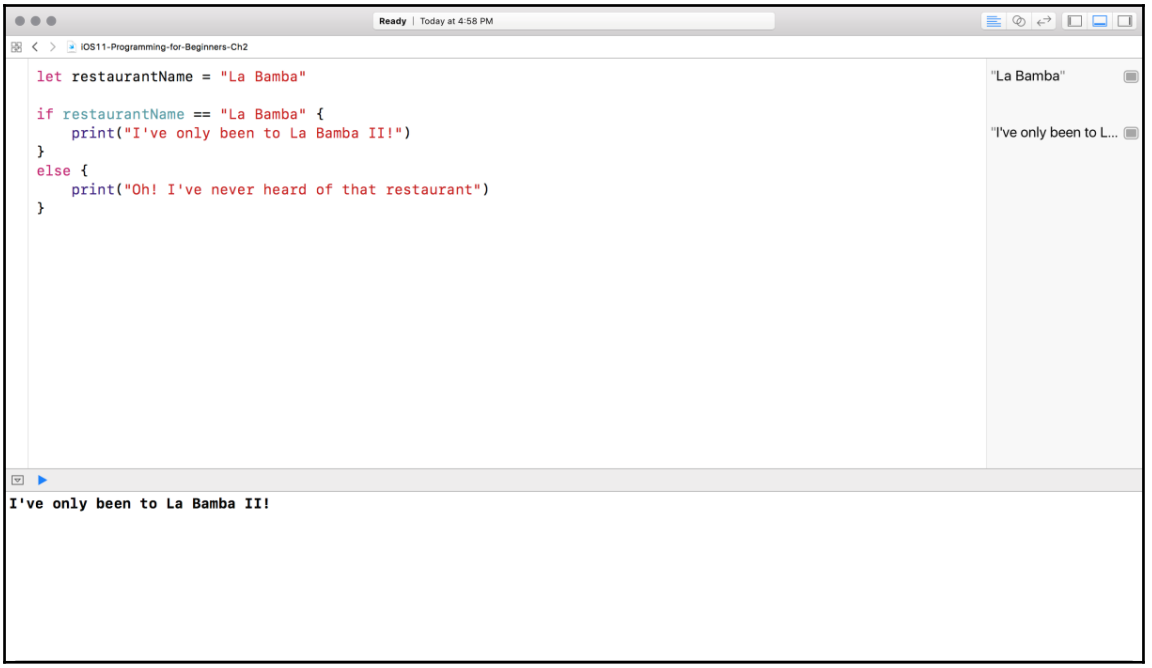


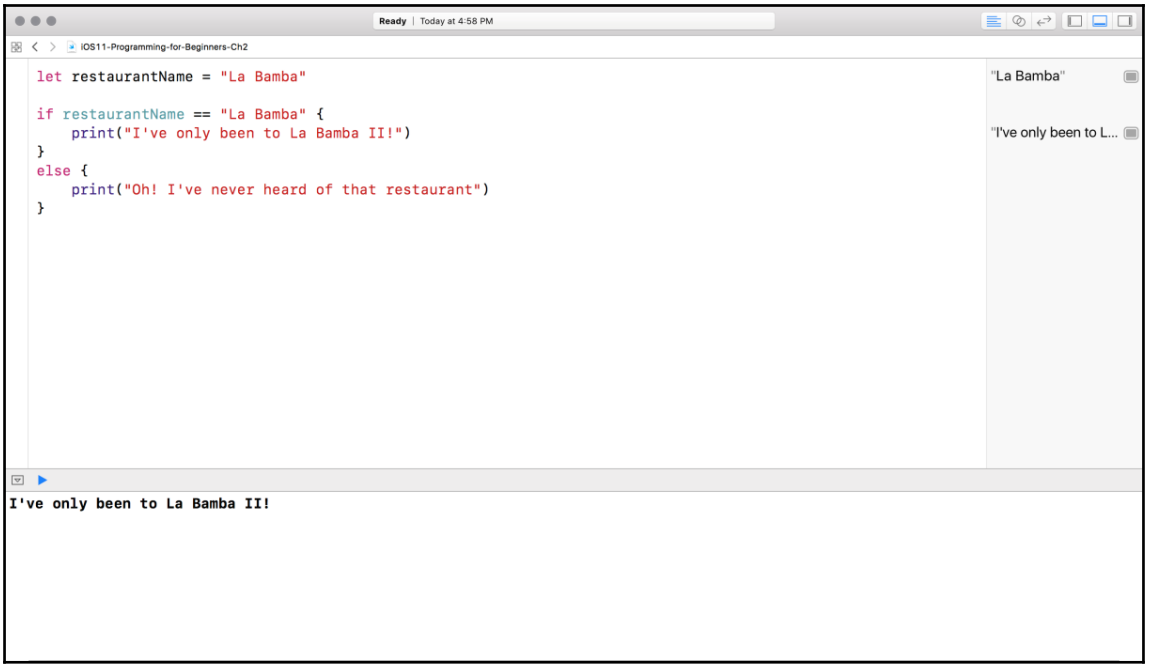

Ready | Today at 4:56 PM
iOS11-Programming-for-Beginners-Ch2
let isPictureVisible = true
if isPictureVisible {
print("Picture is visible")
}
true
"Picture is visible\n"
Picture is visible

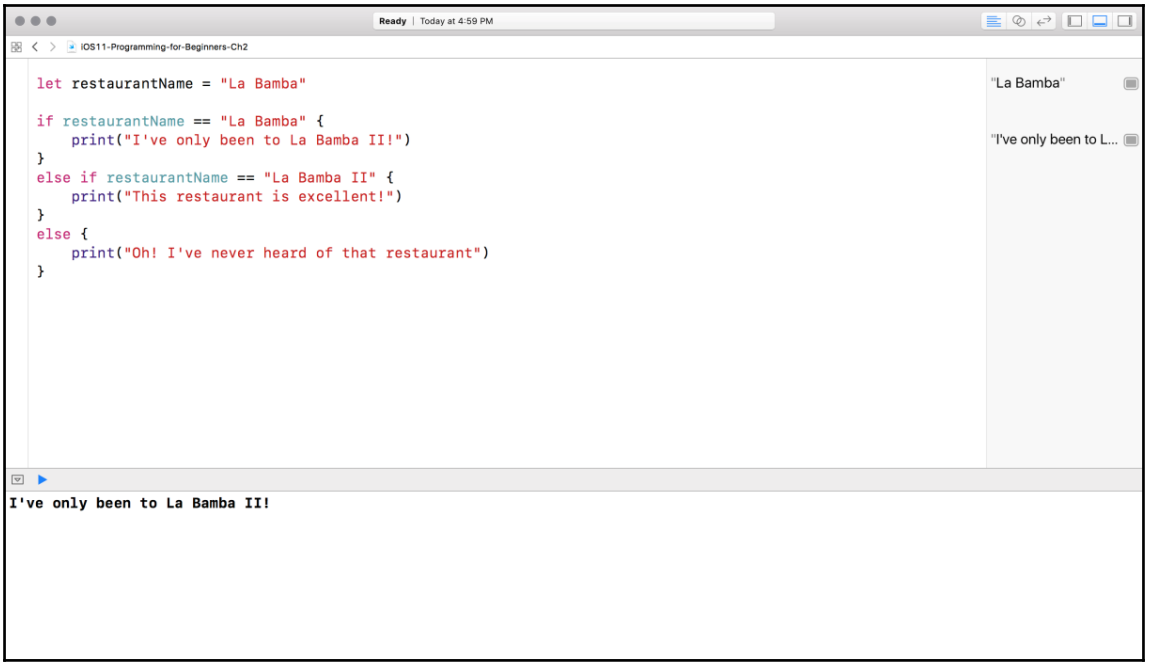

Ready | Today at 4:57 PM
iOS11-Programming-for-Beginners-Ch2
let isRestaurantFound = false
if isRestaurantFound == false {
print("Restaurant was not found")
}
false
"Restaurant was no...
Restaurant was not found

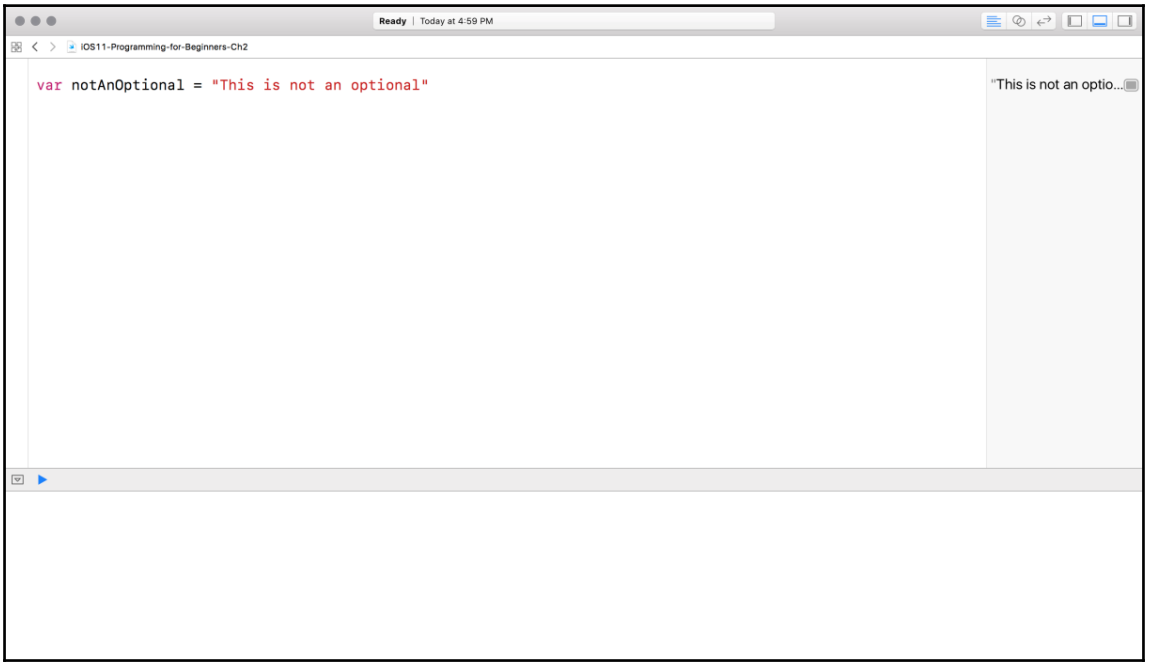

Ready | Today at 4:57 PM
iOS11-Programming-for-Beginners-Ch2
let drinkingAgeLimit = 19
if drinkingAgeLimit < 21 {
print("Since we cannot offer you an adult beverage - would you like a water or soda to drink?")
}
19
"Since we cannot o...
Since we cannot offer you an adult beverage - would you like a water or soda to drink?

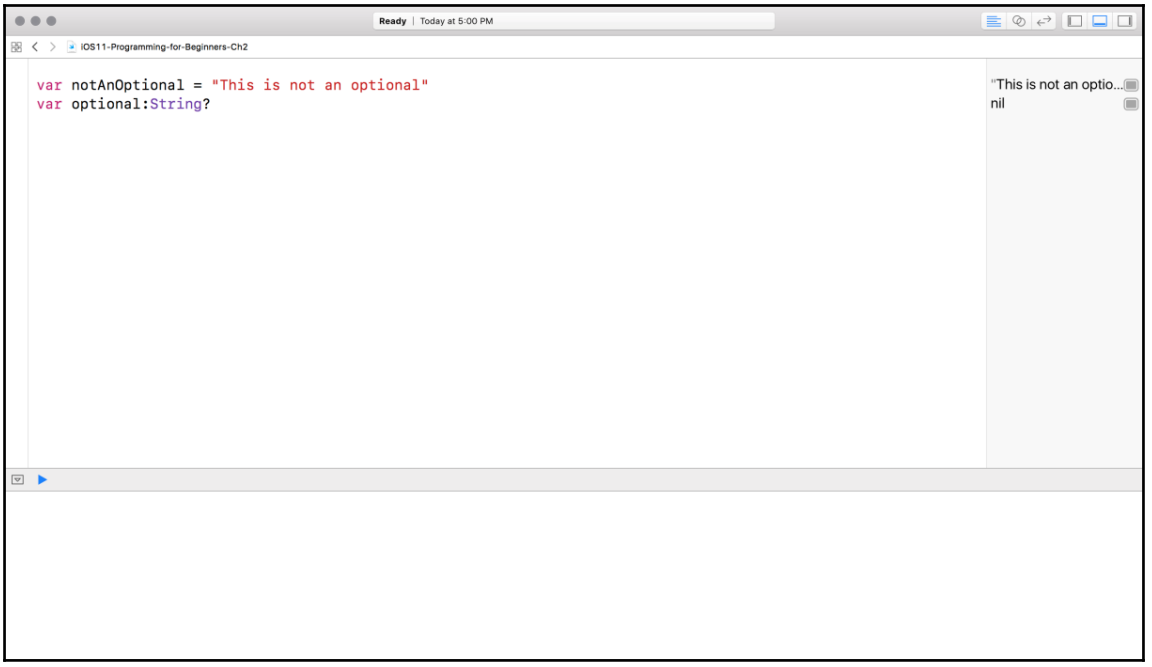

Ready | Today at 4:58 PM
iOS11-Programming-for-Beginners-Ch2
let drinkingAgeLimit = 19
if drinkingAgeLimit < 21 {
print("Since we cannot offer you an adult beverage - would you like a water or soda to drink?")
}
else {
print("What type of beverage would you like? We have adult beverages along with water or soda to
drink.")
}
19
"Since we cannot o...
Since we cannot offer you an adult beverage - would you like a water or soda to drink?

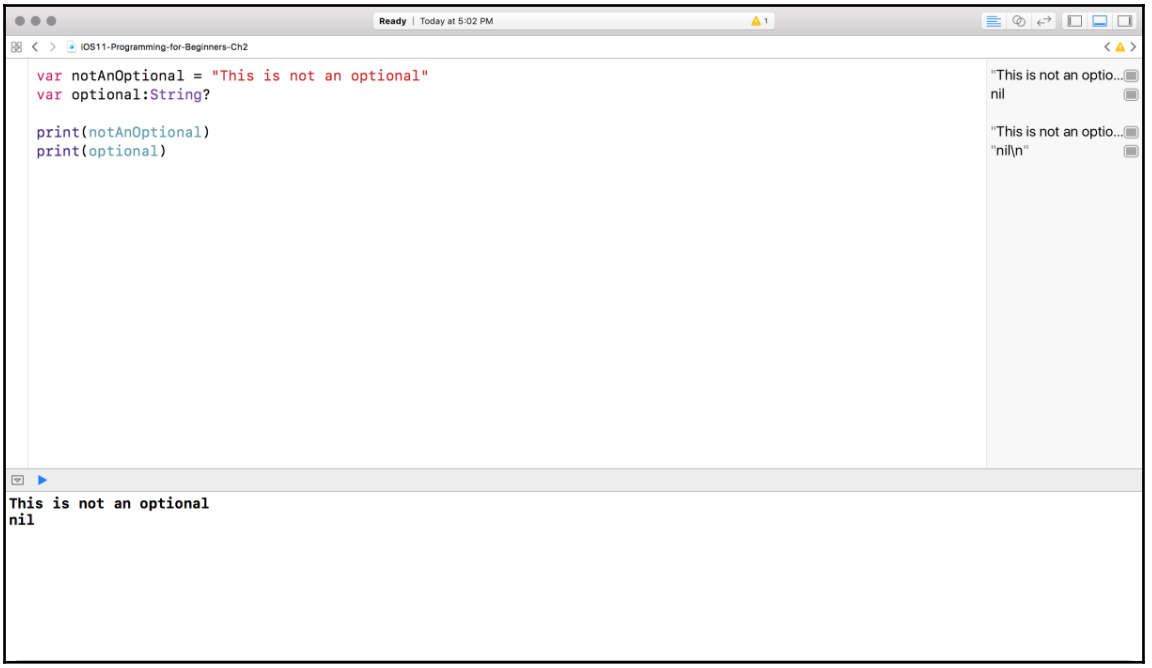

Ready | Today at 4:58 PM
iOS11-Programming-for-Beginners-Ch2
let restaurantName = "La Bamba"
if restaurantName == "La Bamba" {
print("I've only been to La Bamba II!")
}
else {
print("Oh! I've never heard of that restaurant")
}
"La Bamba"
"I've only been to L...
I've only been to La Bamba II!

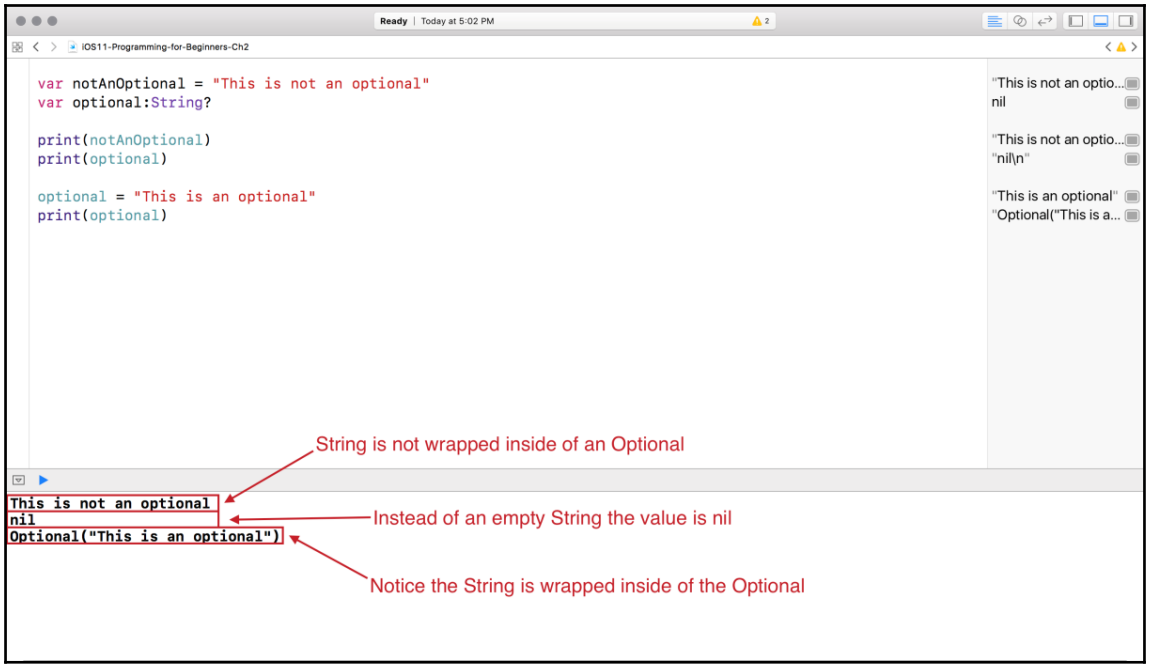

Ready | Today at 4:58 PM
iOS11-Programming-for-Beginners-Ch2
let restaurantName = "La Bamba"
if restaurantName == "La Bamba" {
print("I've only been to La Bamba II!")
}
else {
print("Oh! I've never heard of that restaurant")
}
"La Bamba"
"I've only been to L...
I've only been to La Bamba II!

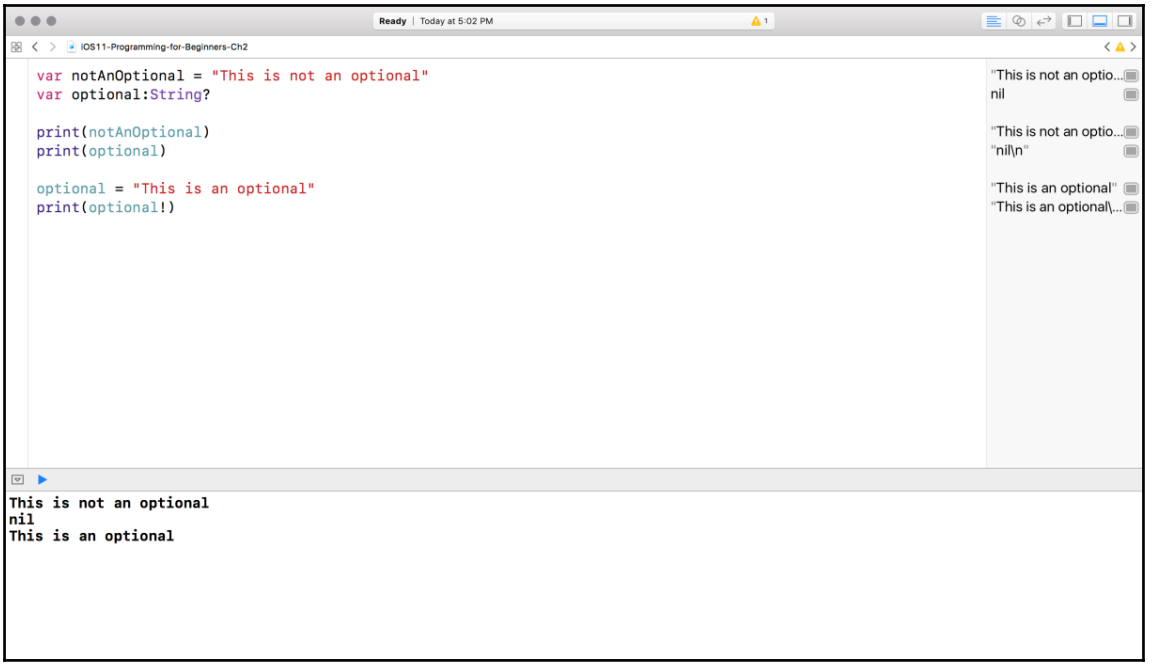

Ready | Today at 4:59 PM
iOS11-Programming-for-Beginners-Ch2
let restaurantName = "La Bamba"
if restaurantName == "La Bamba" {
print("I've only been to La Bamba II!")
}
else if restaurantName == "La Bamba II" {
print("This restaurant is excellent!")
}
else {
print("Oh! I've never heard of that restaurant")
}
"La Bamba"
"I've only been to L...
I've only been to La Bamba II!

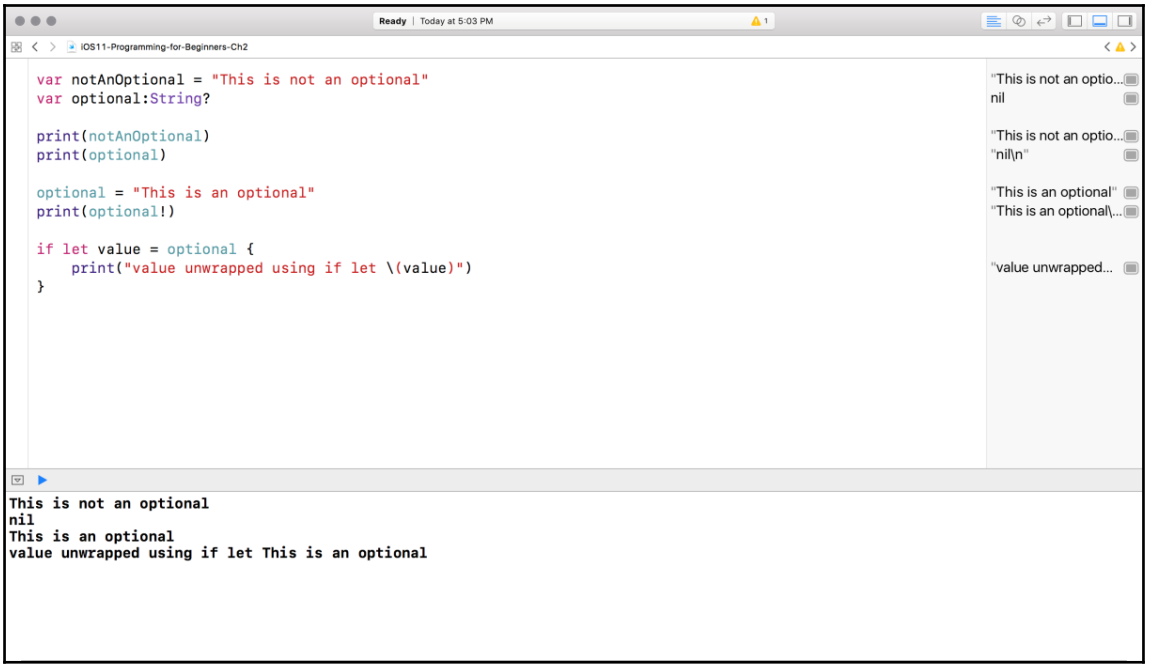

Ready | Today at 4:59 PM
iOS11-Programming-for-Beginners-Ch2
var notAnOptional = "This is not an optional"
"This is not an optio...

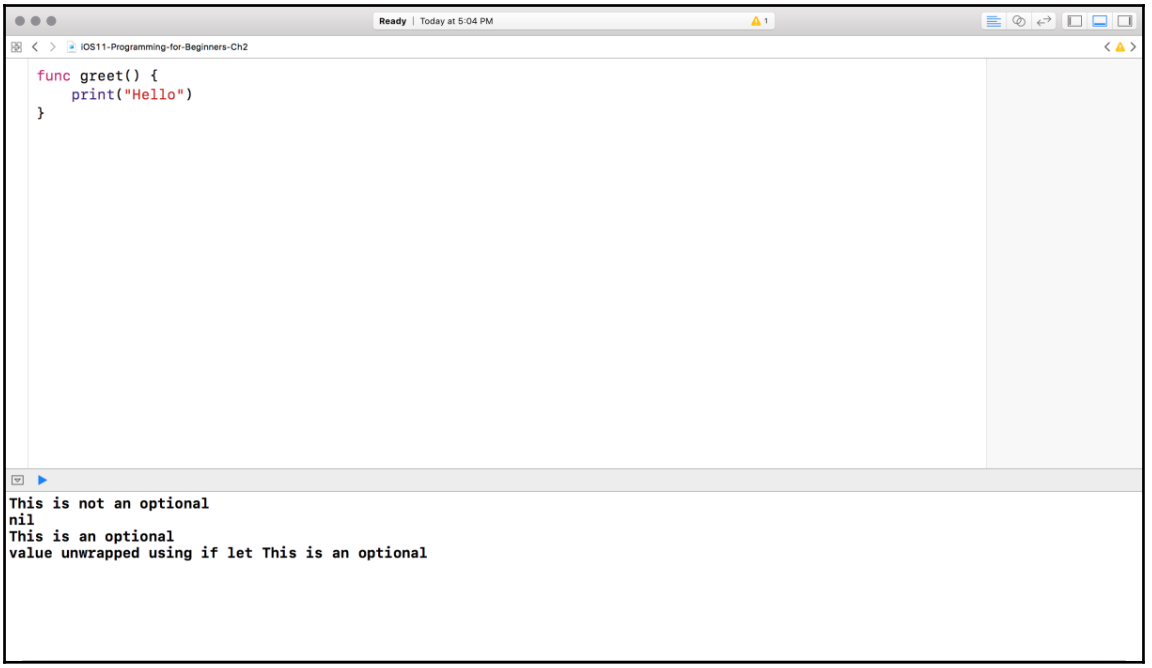

Ready | Today at 5:00 PM
iOS11-Programming-for-Beginners-Ch2
var notAnOptional = "This is not an optional"
var optional:String?
"This is not an optio...
nil

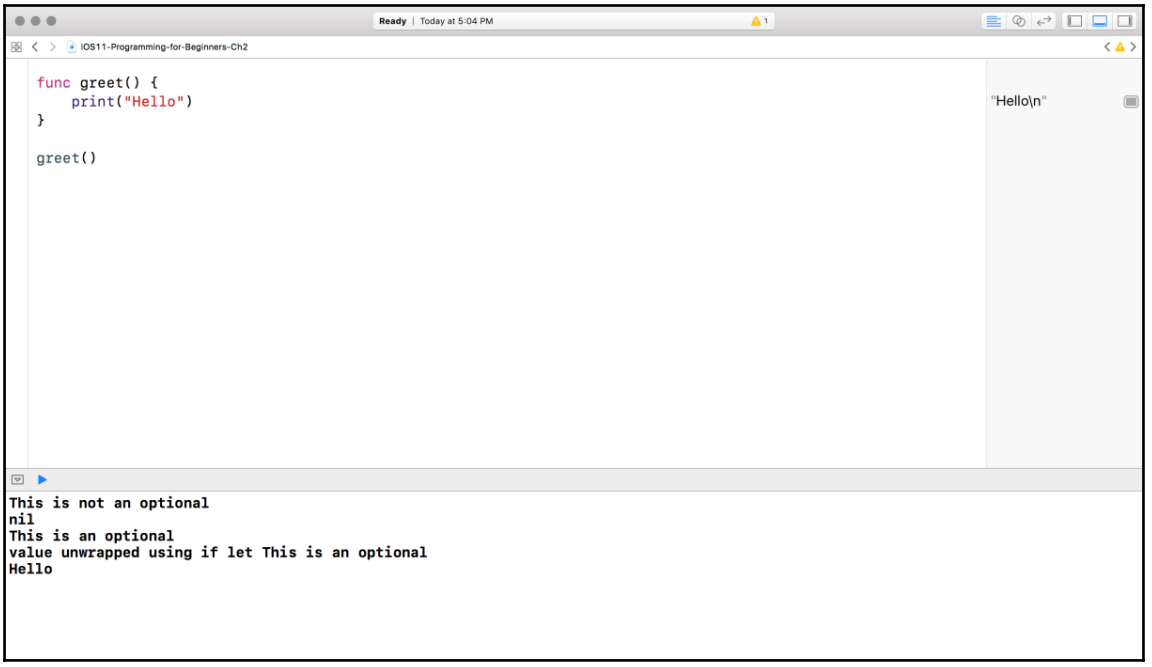

Ready | Today at 5:02 PM
iOS11-Programming-for-Beginners-Ch2
var notAnOptional = "This is not an optional"
var optional:String?
print(notAnOptional)
print(optional)
"This is not an optio...
nil
"This is not an optio...
"nil\n"
This is not an optional
nil

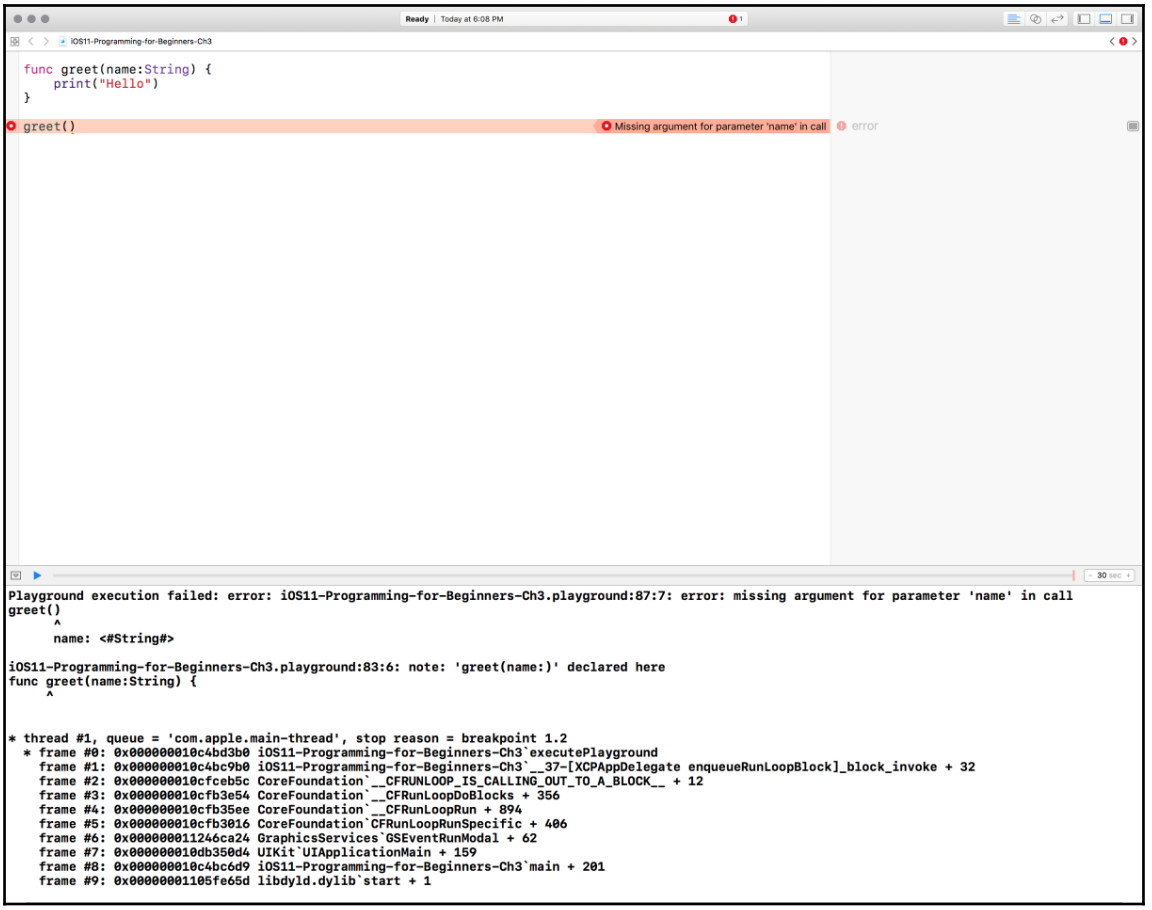

Ready | Today at 5:02 PM
iOS11-Programming-for-Beginners-Ch2
var notAnOptional = "This is not an optional"
var optional:String?
print(notAnOptional)
print(optional)
optional = "This is an optional"
print(optional)
"This is not an optio...
nil
"This is not an optio...
"nil\n"
"This is an optional"
"Optional("This is a...
String is not wrapped inside of an Optional
This is not an optional
nil
Optional("This is an optional")
Instead of an empty String the value is nil
Notice the String is wrapped inside of the Optional

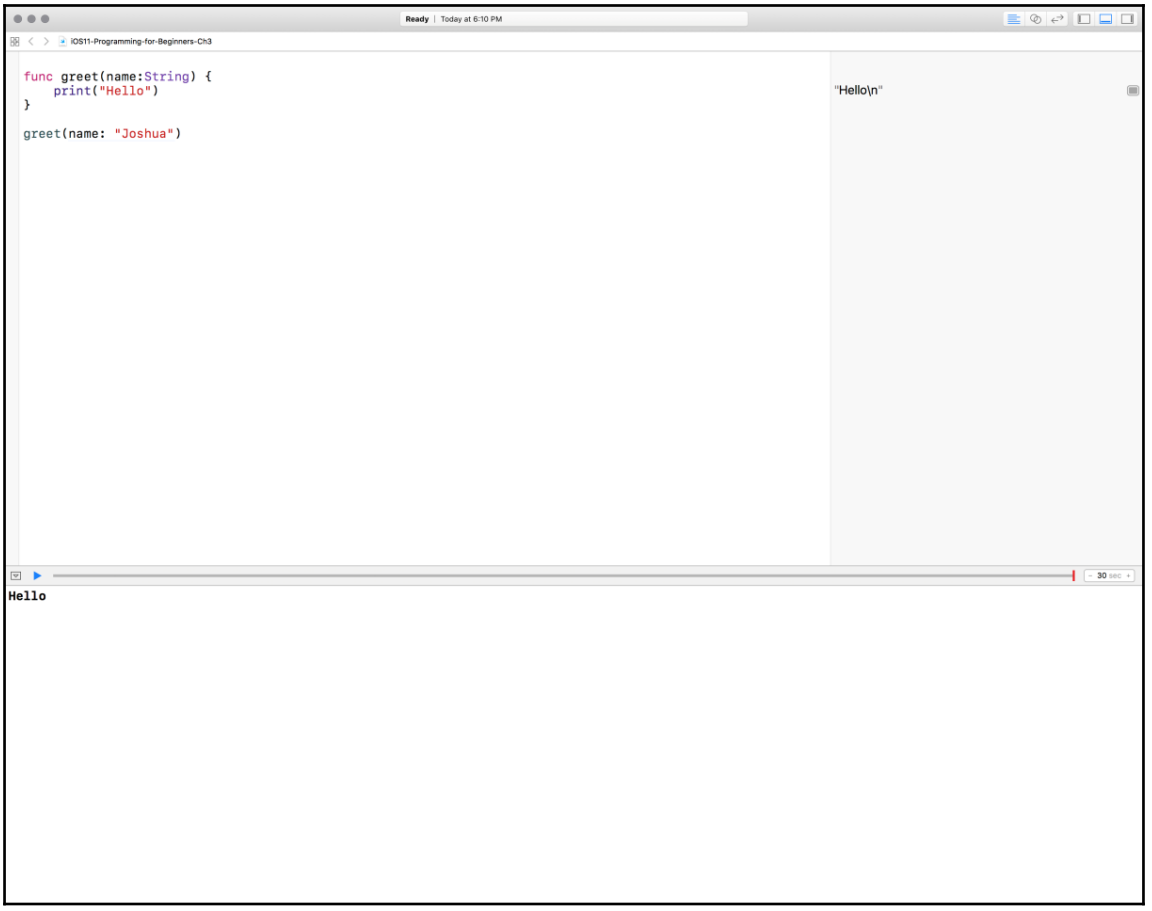

Ready | Today at 5:02 PM
iOS11-Programming-for-Beginners-Ch2
var notAnOptional = "This is not an optional"
var optional:String?
print(notAnOptional)
print(optional)
optional = "This is an optional"
print(optional!)
"This is not an optio...
nil
"This is not an optio...
"nil\n"
"This is an optional"
"This is an optional\...
This is not an optional
nil
This is an optional

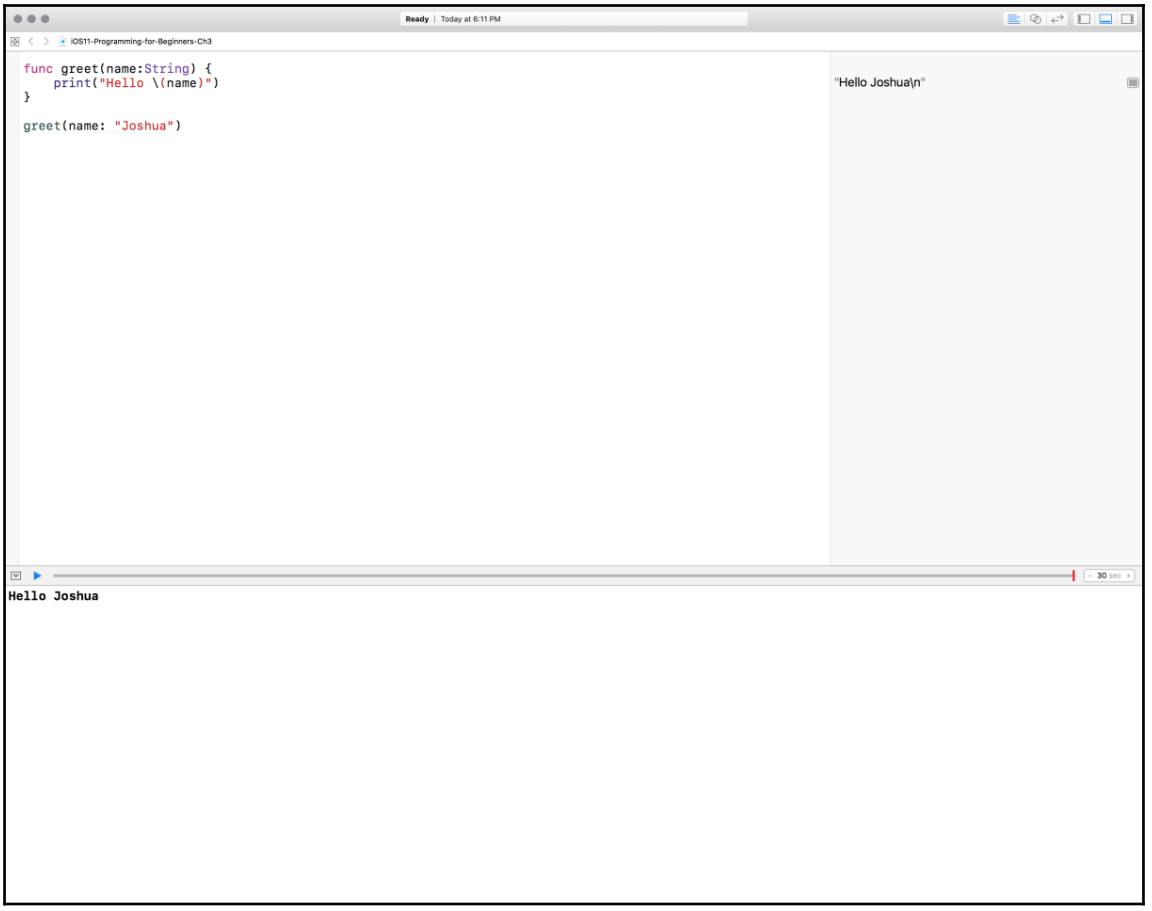

Ready | Today at 5:03 PM
iOS11-Programming-for-Beginners-Ch2
var notAnOptional = "This is not an optional"
var optional:String?
print(notAnOptional)
print(optional)
optional = "This is an optional"
print(optional!)
if let value = optional {
print("value unwrapped using if let \(value)")
}
"This is not an optio...
nil
"This is not an optio...
"nil\n"
"This is an optional"
"This is an optional\...
"value unwrapped...
This is not an optional
nil
This is an optional
value unwrapped using if let This is an optional


Ready | Today at 5:04 PM
iOS11-Programming-for-Beginners-Ch2
func greet() {
print("Hello")
}
This is not an optional
nil
This is an optional
value unwrapped using if let This is an optional


Ready | Today at 5:04 PM
iOS11-Programming-for-Beginners-Ch2
func greet() {
print("Hello")
}
greet()
"Hello\n"
This is not an optional
nil
This is an optional
value unwrapped using if let This is an optional
Hello


Ready | Today at 6:08 PM
iOS11-Programming-for-Beginners-Ch3
func greet(name:String) {
print("Hello")
}
greet()
Missing argument for parameter 'name' in call
error
Playground execution failed: error: iOS11-Programming-for-Beginners-Ch3.playground:87:7: error: missing argument for parameter 'name' in call
greet()
^
name: <#String#>
iOS11-Programming-for-Beginners-Ch3.playground:83:6: note: 'greet(name:)' declared here
func greet(name:String) {
^
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.2
* frame #0: 0x000000010c4bd3b0 iOS11-Programming-for-Beginners-Ch3`executePlayground
frame #1: 0x000000010c4bc9b0 iOS11-Programming-for-Beginners-Ch3`__37-[XCPAppDelegate enqueueRunLoopBlock]_block_invoke + 32
frame #2: 0x000000010cfceb5c CoreFoundation`__CFRUNLOOP_IS_CALLING_OUT_TO_A_BLOCK__ + 12
frame #3: 0x000000010cfb3e54 CoreFoundation`__CFRunLoopDoBlocks + 356
frame #4: 0x000000010cfb35ee CoreFoundation`__CFRunLoopRun + 894
frame #5: 0x000000010cfb3016 CoreFoundation`CFRunLoopRunSpecific + 406
frame #6: 0x0000000011246ca24 GraphicsServices`GSEventRunModal + 62
frame #7: 0x000000010db350d4 UIKit`UIApplicationMain + 159
frame #8: 0x000000010c4bc6d9 iOS11-Programming-for-Beginners-Ch3`main + 201
frame #9: 0x00000001105fe65d libdyld.dylib`start + 1

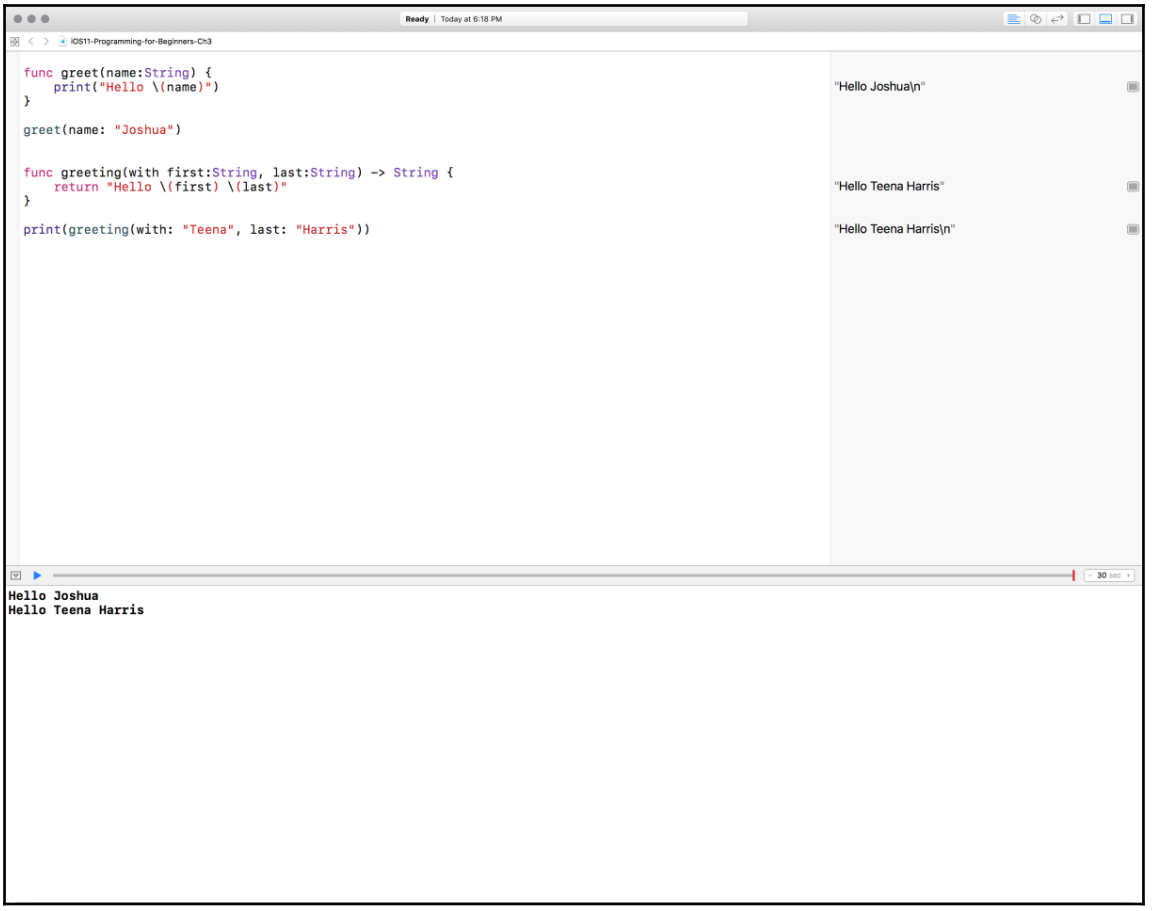

Ready | Today at 6:10 PM
iOS11-Programming-for-Beginners-Ch3
func greet(name:String) {
print("Hello")
}
greet(name: "Joshua")
"Hello\n"
30 sec
Hello


Ready | Today at 6:11 PM
iOS11-Programming-for-Beginners-Ch3
func greet(name:String) {
print("Hello \(name)")
}
greet(name: "Joshua")
"Hello Joshua\n"
30 sec
Hello Joshua

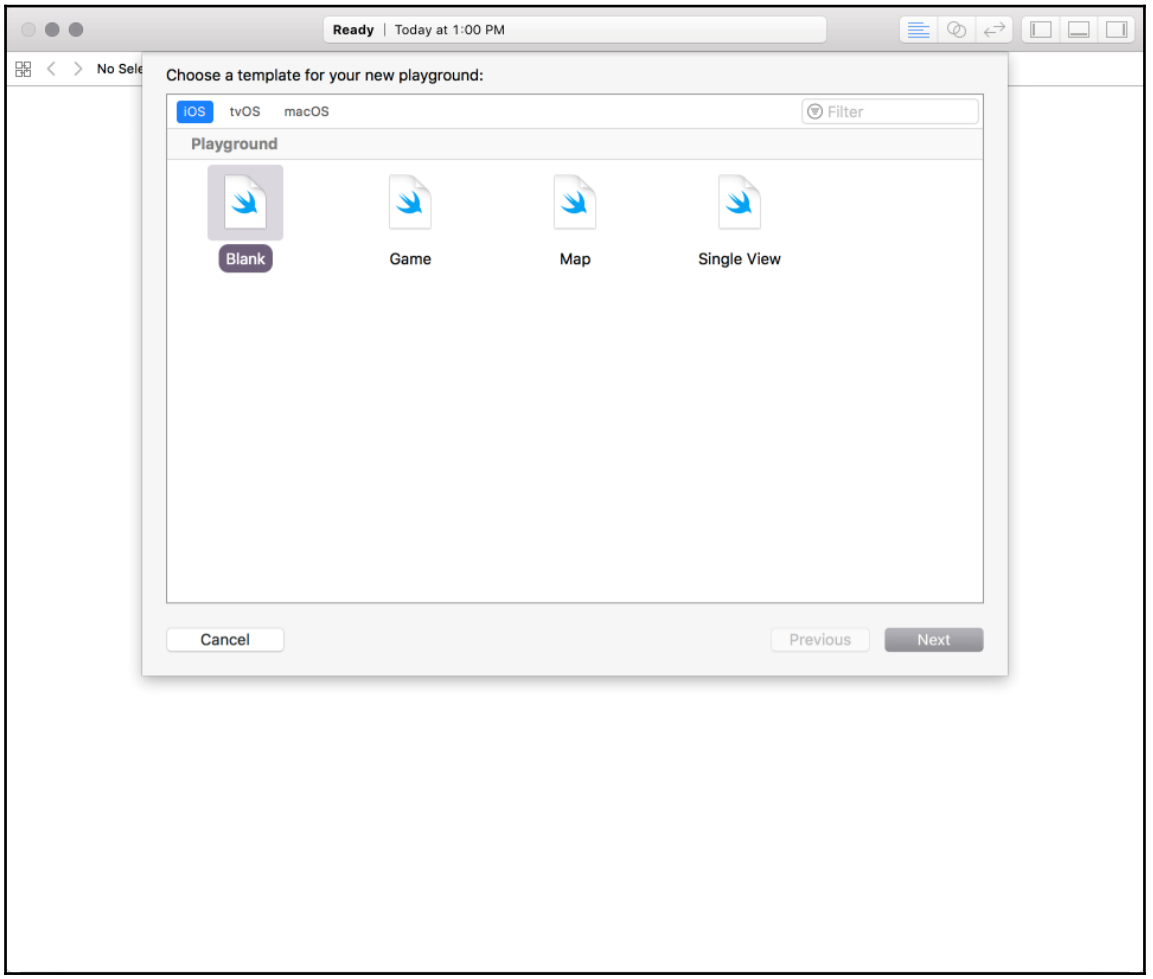

Ready | Today at 5:08 PM
iOS11-Programming-for-Beginners-Ch2
func greet(name:String) {
print("Hello \(name)")
}
greet(name: "Joshua")
func greet(first:String, last:String) {
print("Hello \(first) \(last)")
}
"Hello Joshua\n"
"Hello Jason Clayto...
This is not an optional
This is an optional
value unwrapped using if let This is an optional
Hello Joshua
Hello Jason Clayton


Ready | Today at 5:08 PM
iOS11-Programming-for-Beginners-Ch2
func greet(name:String) {
print("Hello \(name)")
}
greet(name: "Joshua")
func greet(first:String, last:String) {
print("Hello \(first) \(last)")
}
greet(first: "Jason", last: "Clayton")
"Hello Joshua\n"
"Hello Jason Clayto...
This is not an optional
This is an optional
value unwrapped using if let This is an optional
Hello Joshua
Hello Jason Clayton


Ready | Today at 6:15 PM
iOS11-Programming-for-Beginners-Ch3
func greet(name:String) {
print("Hello \(name)")
}
greet(name: "Joshua")
func greeting(with first:String, last:String) -> String {
return "Hello \(first) \(last)"
}
"Hello Joshua\n"
Hello Joshua


Ready | Today at 6:18 PM
iOS11-Programming-for-Beginners-Ch3
func greet(name:String) {
print("Hello \(name)")
}
greet(name: "Joshua")
func greeting(with first:String, last:String) -> String {
return "Hello \(first) \(last)"
}
print(greeting(with: "Teena", last: "Harris"))
"Hello Joshua\n"
"Hello Teena Harris"
"Hello Teena Harris\n"
Hello Joshua
Hello Teena Harris

Chapter 4: Digging Deeper

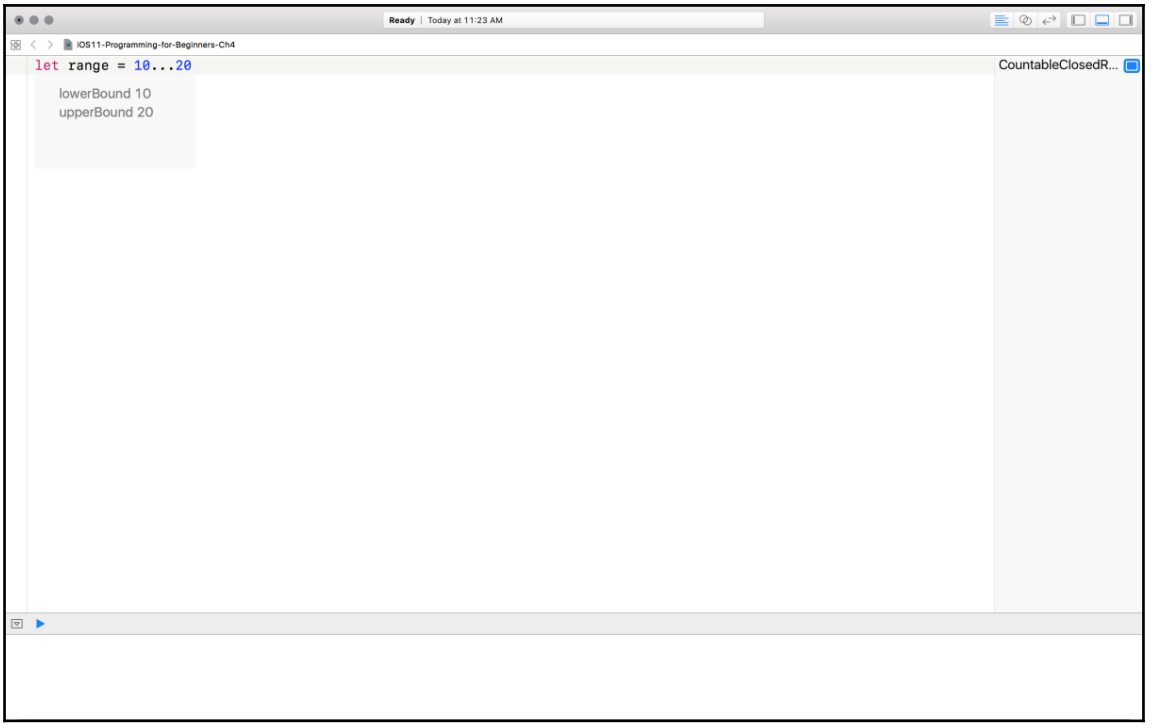


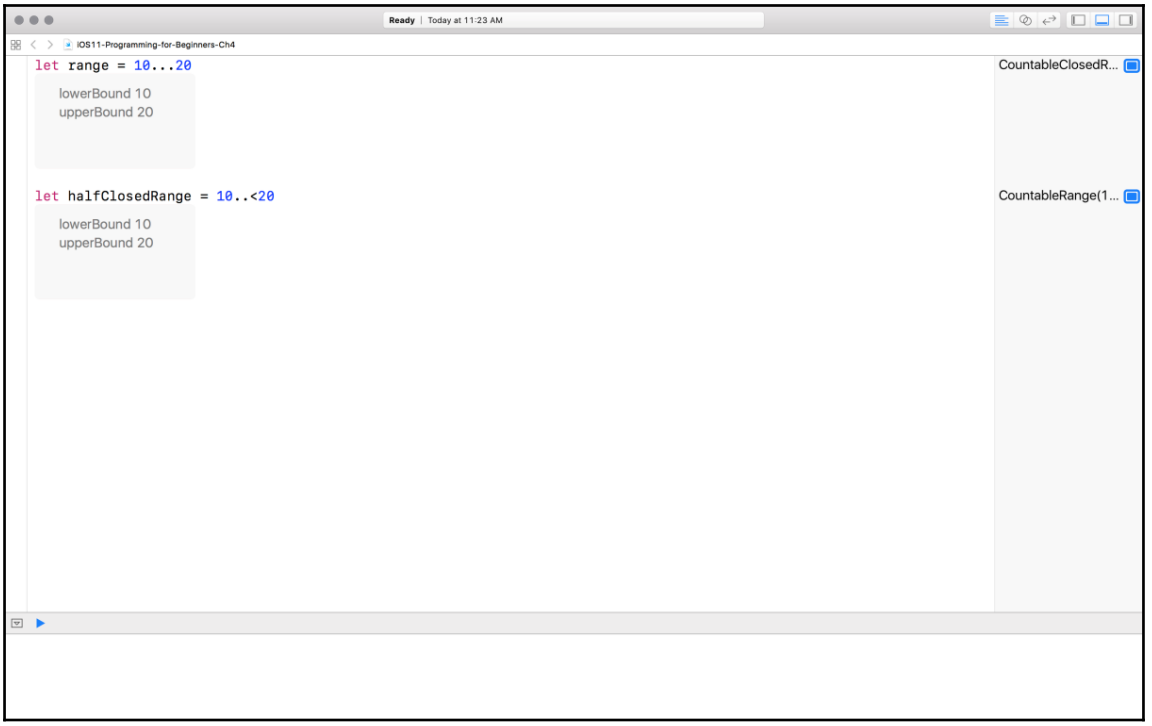
10 11 12 13 14 15 16 17 18 19 20

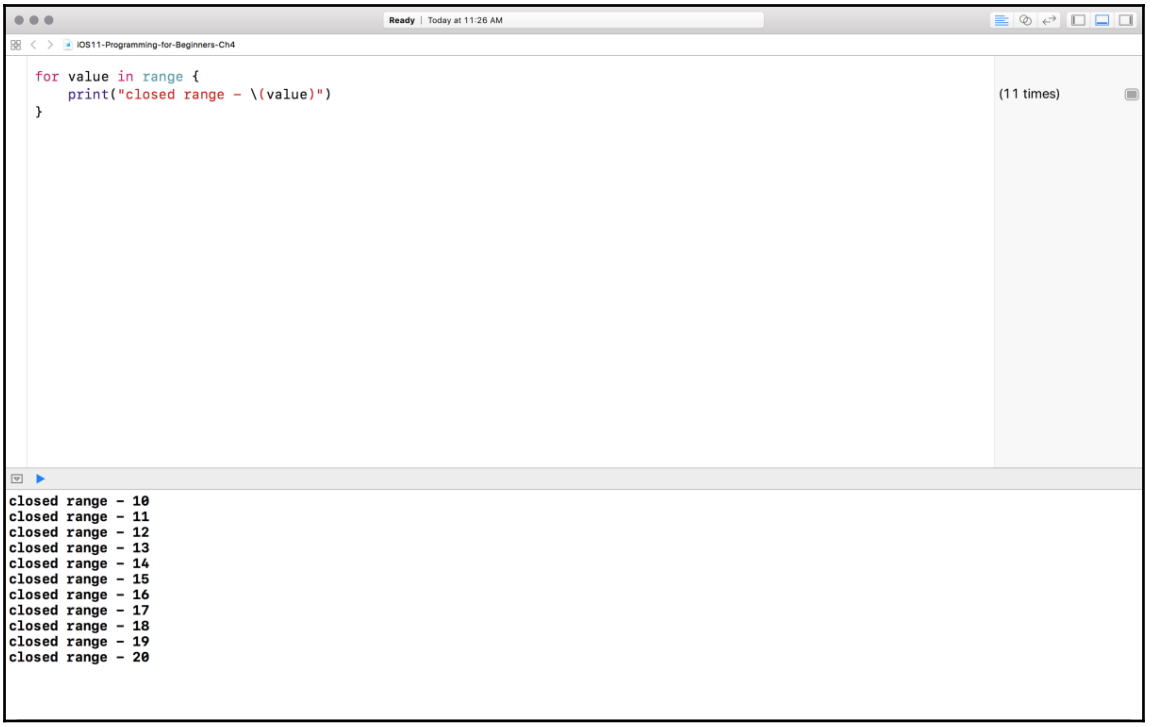
10 11 12 13 14 15 16 17 18 19 20

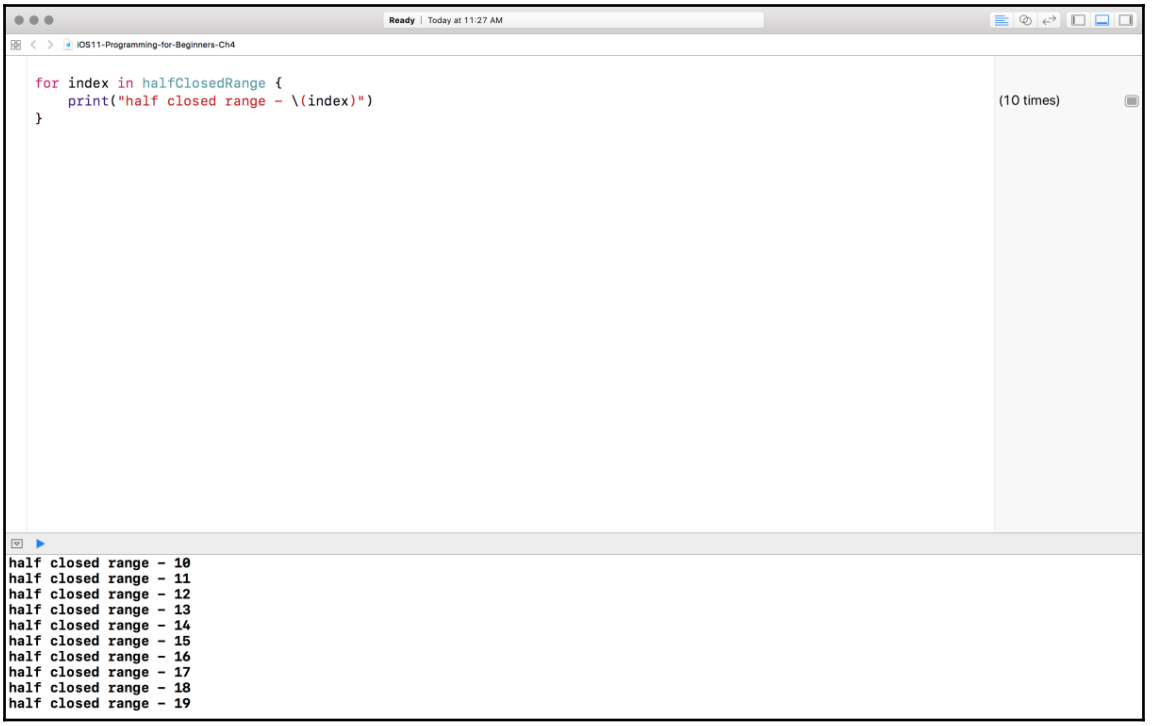

Ready | Today at 11:20 AM
iOS11-Programming-for-Beginners-Ch4
let range = 10...20
CountableClosedR...

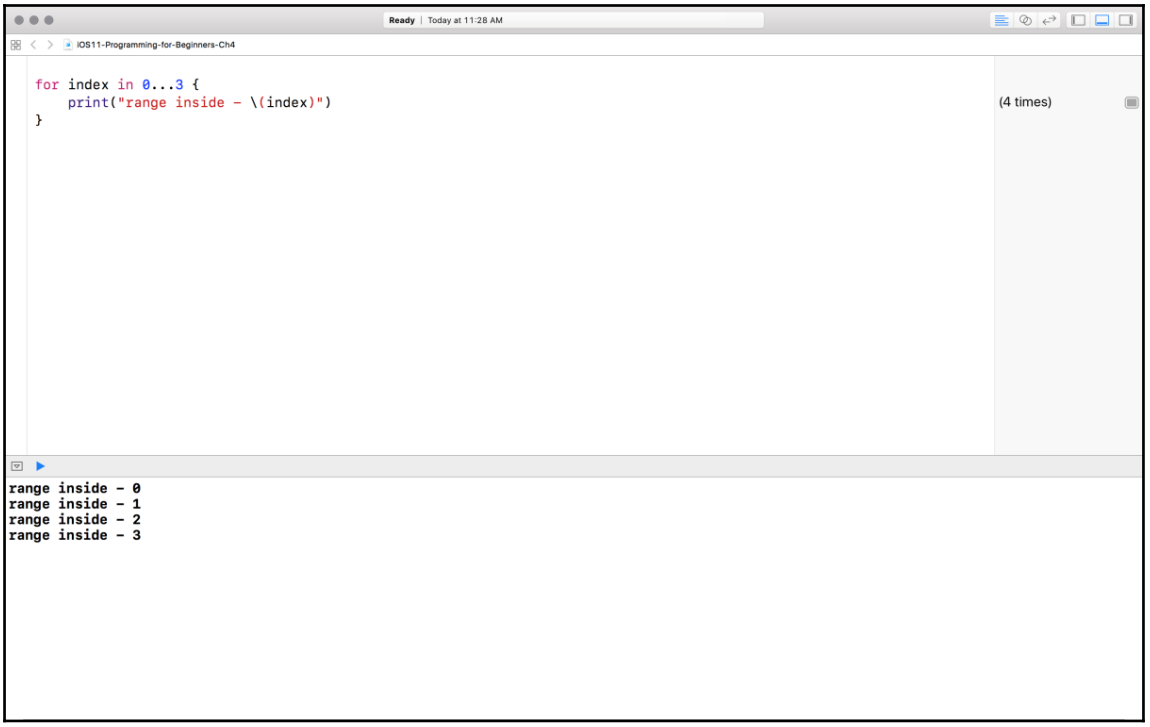

Ready | Today at 11:20 AM
iOS11-Programming-for-Beginners-Ch4
let range = 10...20
CountableClosedR...

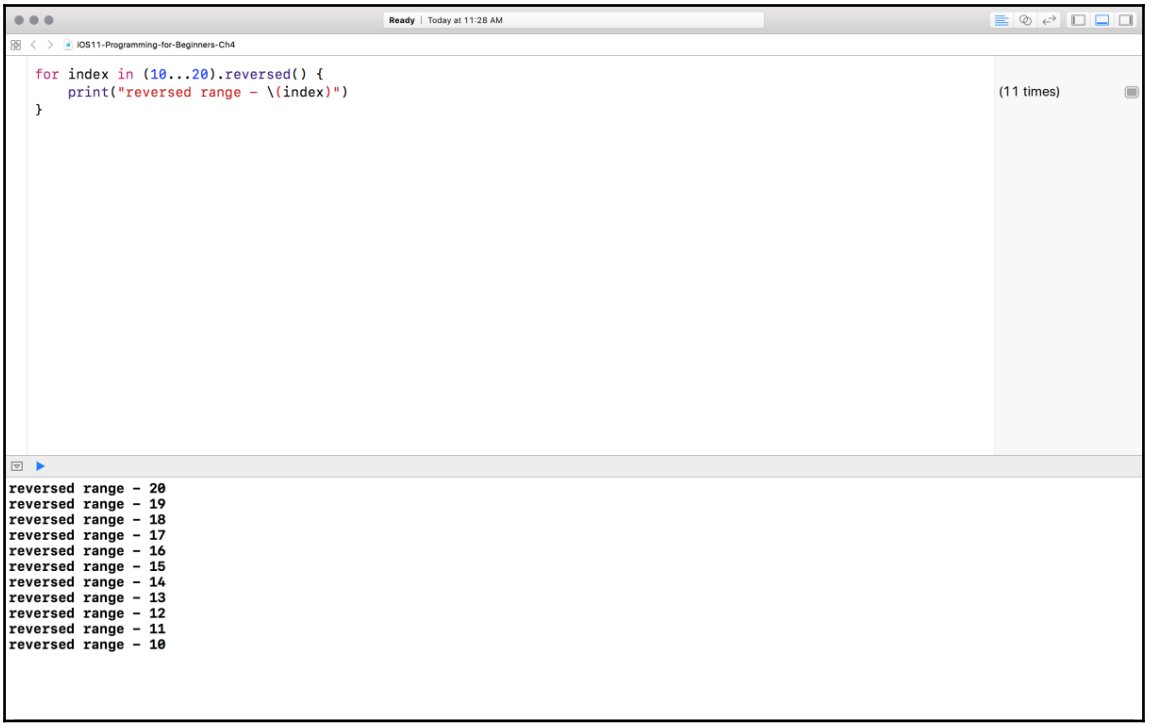

Ready | Today at 11:23 AM
iOS11-Programming-for-Beginners-Ch4
let range = 10...20
lowerBound 10
upperBound 20
CountableClosedR...

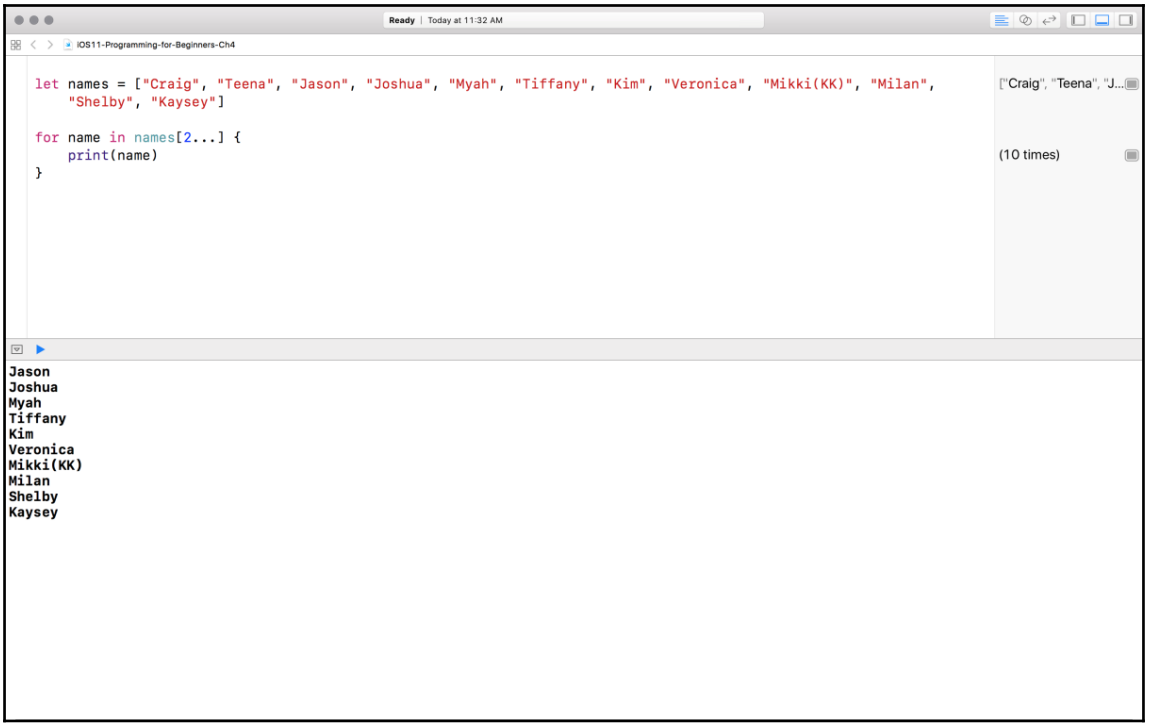

Ready | Today at 11:23 AM
iOS11-Programming-for-Beginners-Ch4
let range = 10...20
lowerBound 10
upperBound 20
CountableClosedR...
let halfClosedRange = 10..<20
lowerBound 10
upperBound 20
CountableRange(1...

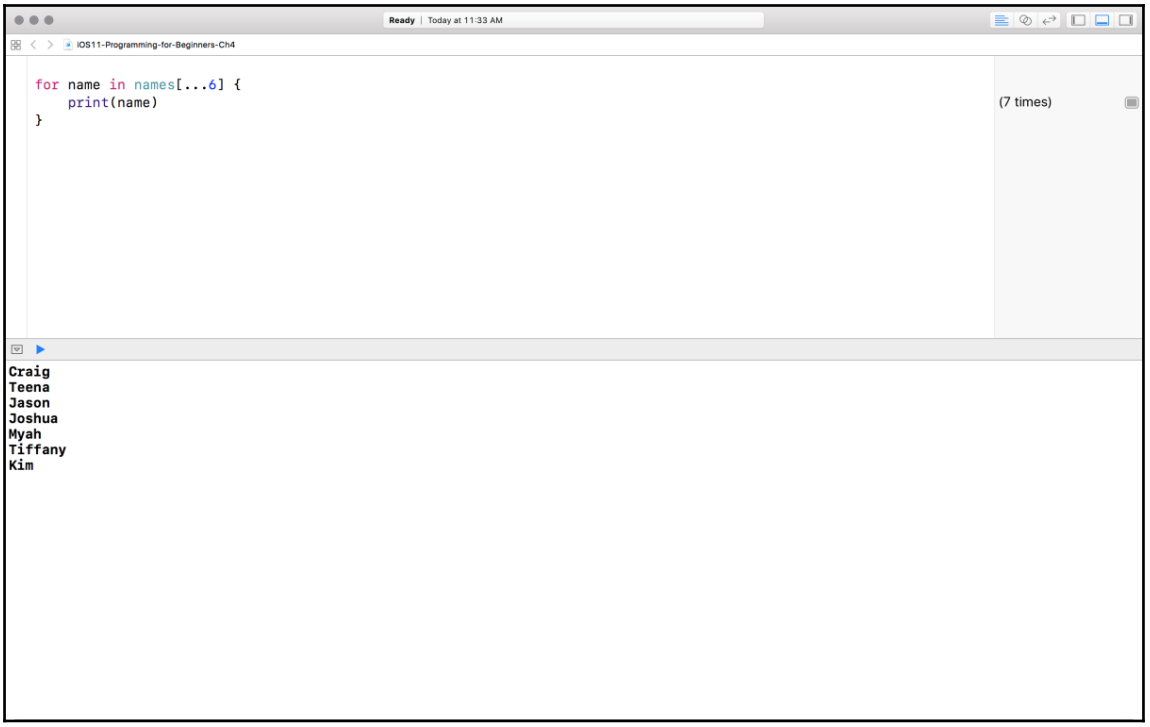

Ready | Today at 11:26 AM
iOS11-Programming-for-Beginners-Ch4
for value in range {
print("closed range - \(value)")
}
(11 times)
closed range - 10
closed range - 11
closed range - 12
closed range - 13
closed range - 14
closed range - 15
closed range - 16
closed range - 17
closed range - 18
closed range - 19
closed range - 20

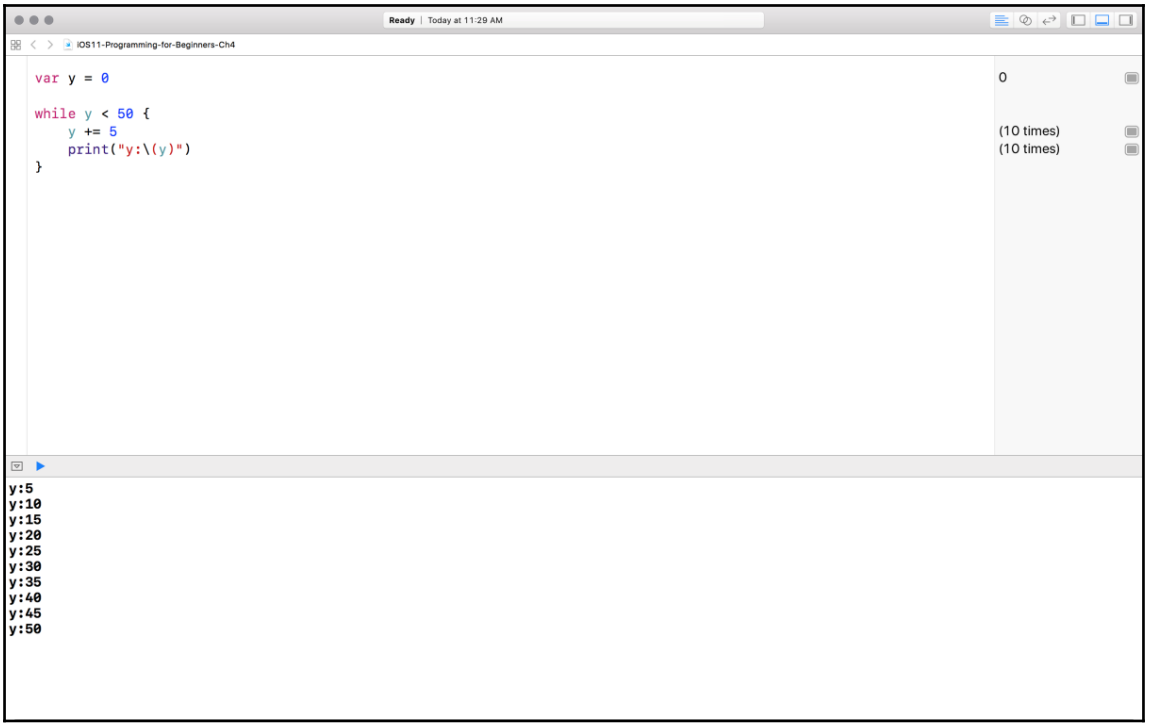

Ready | Today at 11:27 AM
iOS11-Programming-for-Beginners-Ch4
for index in halfClosedRange {
print("half closed range - \(index)")
}
(10 times)
half closed range - 10
half closed range - 11
half closed range - 12
half closed range - 13
half closed range - 14
half closed range - 15
half closed range - 16
half closed range - 17
half closed range - 18
half closed range - 19

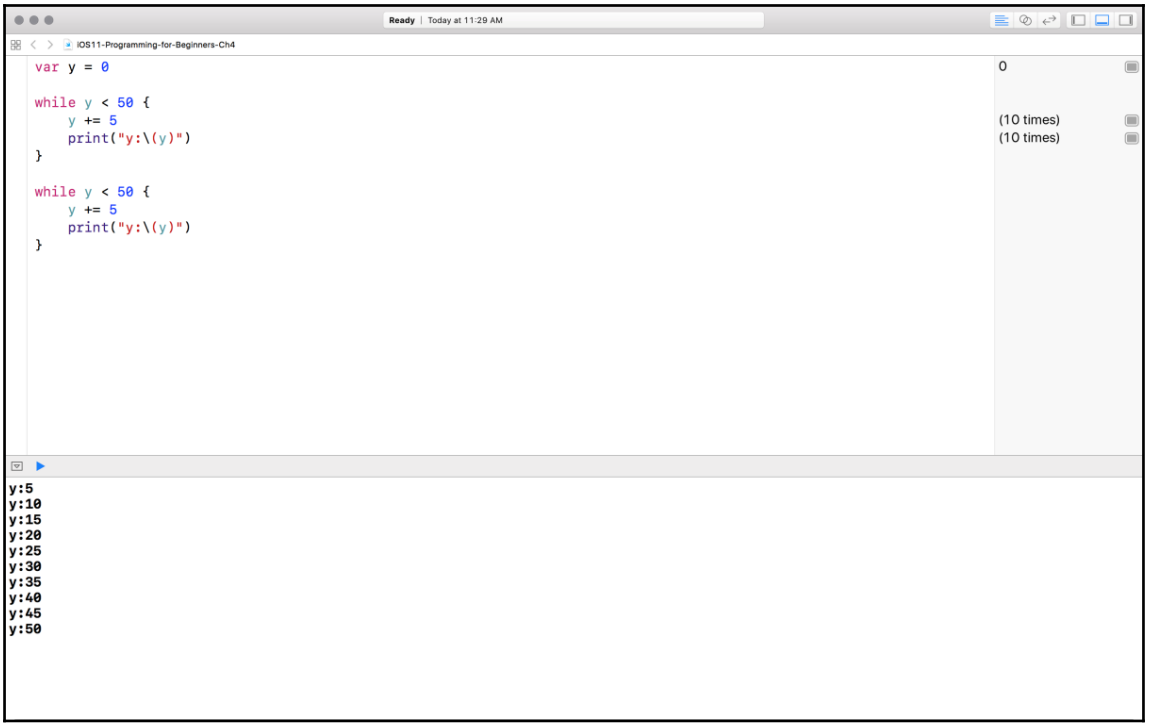

Ready | Today at 11:28 AM
iOS11-Programming-for-Beginners-Ch4
for index in 0...3 {
print("range inside - \(index)")
}
(4 times)
range inside - 0
range inside - 1
range inside - 2
range inside - 3

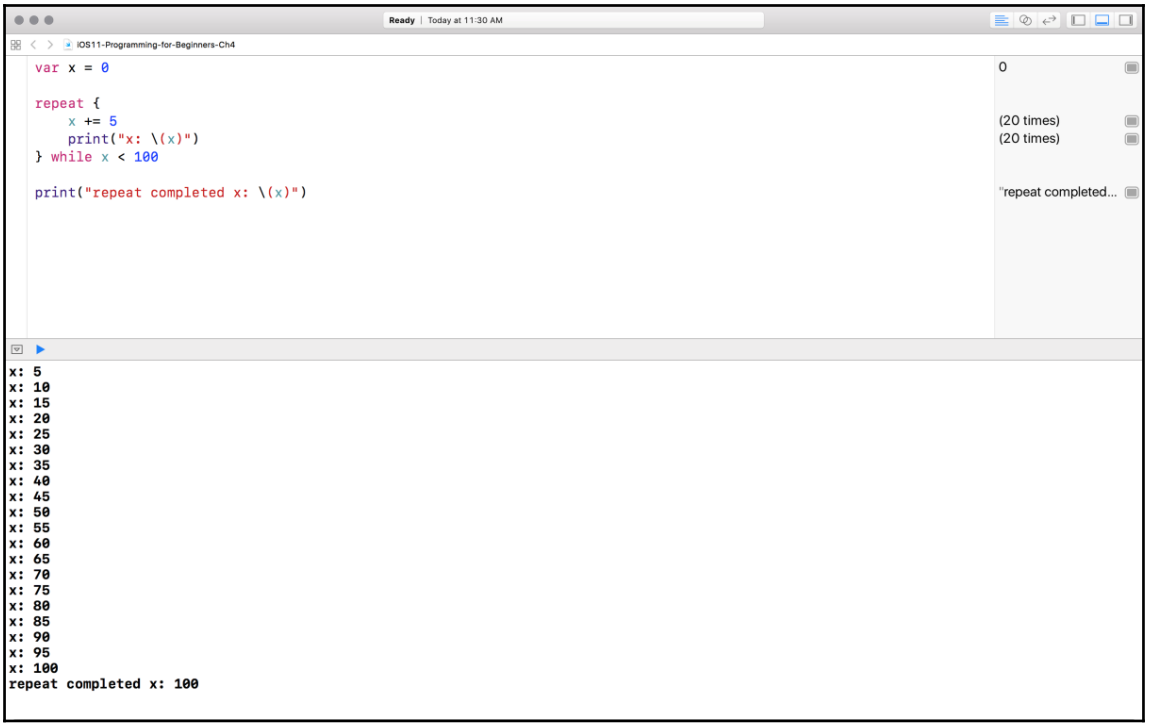

Ready | Today at 11:28 AM
iOS11-Programming-for-Beginners-Ch4
for index in (10...20).reversed() {
print("reversed range - \(index)")
}
(11 times)
reversed range - 20
reversed range - 19
reversed range - 18
reversed range - 17
reversed range - 16
reversed range - 15
reversed range - 14
reversed range - 13
reversed range - 12
reversed range - 11
reversed range - 10

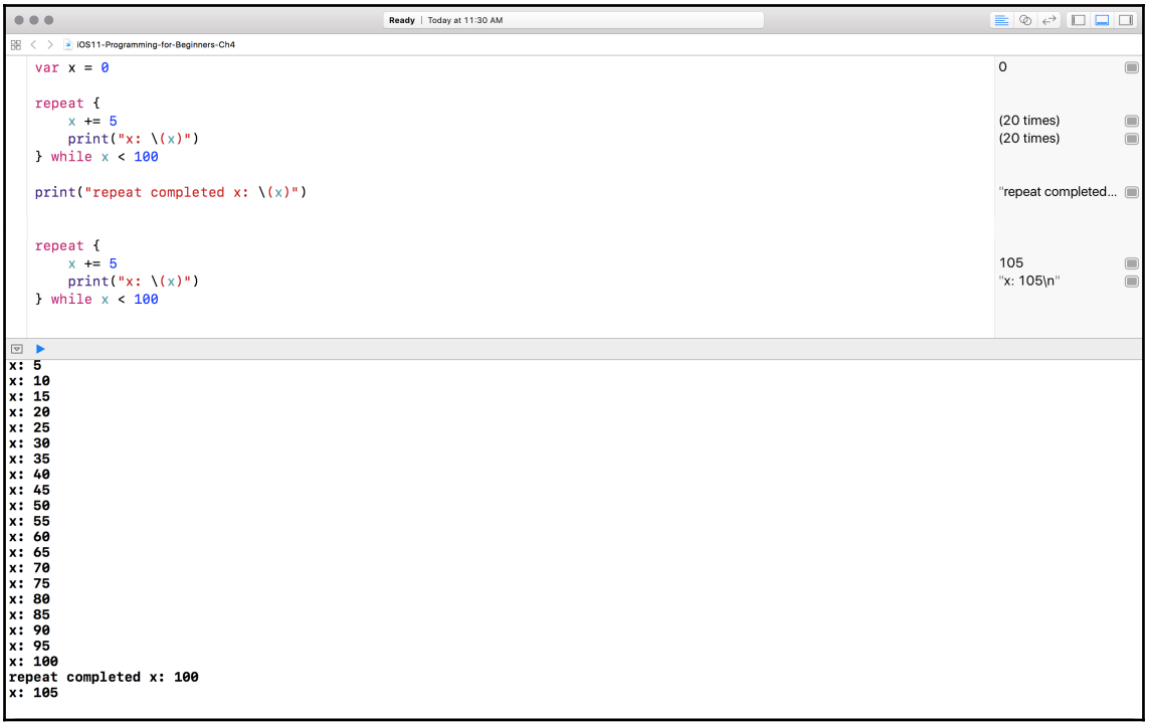

Ready | Today at 11:32 AM
iOS11-Programming-for-Beginners-Ch4
let names = ["Craig", "Teena", "Jason", "Joshua", "Myah", "Tiffany", "Kim", "Veronica", "Mikki(KK)", "Milan",
"Shelby", "Kaysey"]
for name in names[2...] {
print(name)
}
["Craig", "Teena", "J...
(10 times)
Jason
Joshua
Myah
Tiffany
Kim
Veronica
Mikki(KK)
Milan
Shelby
Kaysey


Ready | Today at 11:33 AM
iOS11-Programming-for-Beginners-Ch4
for name in names[...6] {
print(name)
}
(7 times)
Craig
Teena
Jason
Joshua
Myah
Tiffany
Kim


Ready | Today at 11:29 AM
iOS11-Programming-for-Beginners-Ch4
var y = 0
while y < 50 {
y += 5
print("y:\(y)")
}
0
(10 times)
(10 times)
y:5
y:10
y:15
y:20
y:25
y:30
y:35
y:40
y:45
y:50


Ready | Today at 11:29 AM
iOS11-Programming-for-Beginners-Ch4
var y = 0
while y < 50 {
y += 5
print("y:\(y)")
}
while y < 50 {
y += 5
print("y:\(y)")
}
0
(10 times)
(10 times)
y:5
y:10
y:15
y:20
y:25
y:30
y:35
y:40
y:45
y:50


Ready | Today at 11:30 AM
iOS11-Programming-for-Beginners-Ch4
var x = 0
repeat {
x += 5
print("x: \(x)")
} while x < 100
print("repeat completed x: \(x)")
0
(20 times)
(20 times)
"repeat completed...
x: 5
x: 10
x: 15
x: 20
x: 25
x: 30
x: 35
x: 40
x: 45
x: 50
x: 55
x: 60
x: 65
x: 70
x: 75
x: 80
x: 85
x: 90
x: 95
x: 100
repeat completed x: 100


Ready | Today at 11:30 AM
iOS11-Programming-for-Beginners-Ch4
var x = 0
repeat {
x += 5
print("x: \(x)")
} while x < 100
print("repeat completed x: \(x)")
repeat {
x += 5
print("x: \(x)")
} while x < 100
0
(20 times)
(20 times)
"repeat completed...
105
"x: 105\n"
x: 5
x: 10
x: 15
x: 20
x: 25
x: 30
x: 35
x: 40
x: 45
x: 50
x: 55
x: 60
x: 65
x: 70
x: 75
x: 80
x: 85
x: 90
x: 95
x: 100
repeat completed x: 100
x: 105

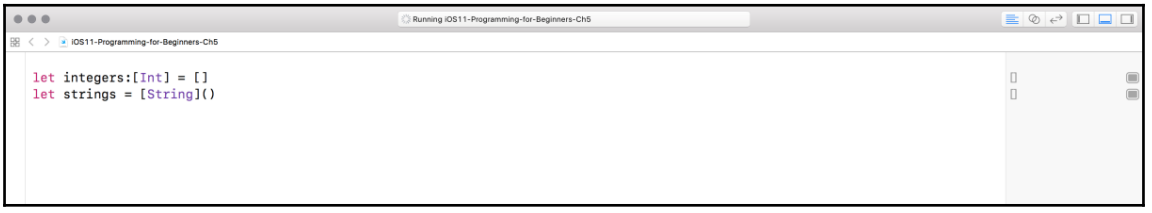
Chapter 5: Digging into Collections

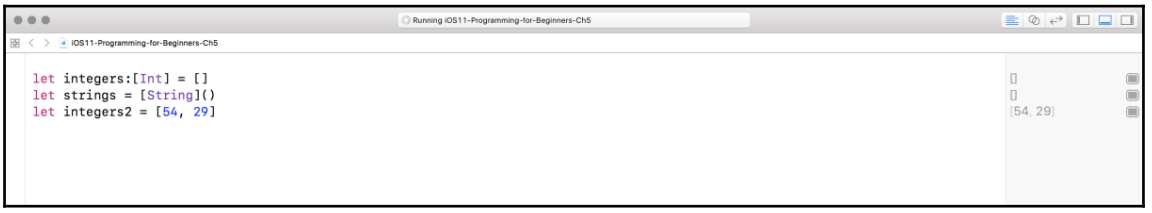
0	Florida
1	Ohio
2	California
3	North Carolina
4	Colorado
5	Nevada
6	New York

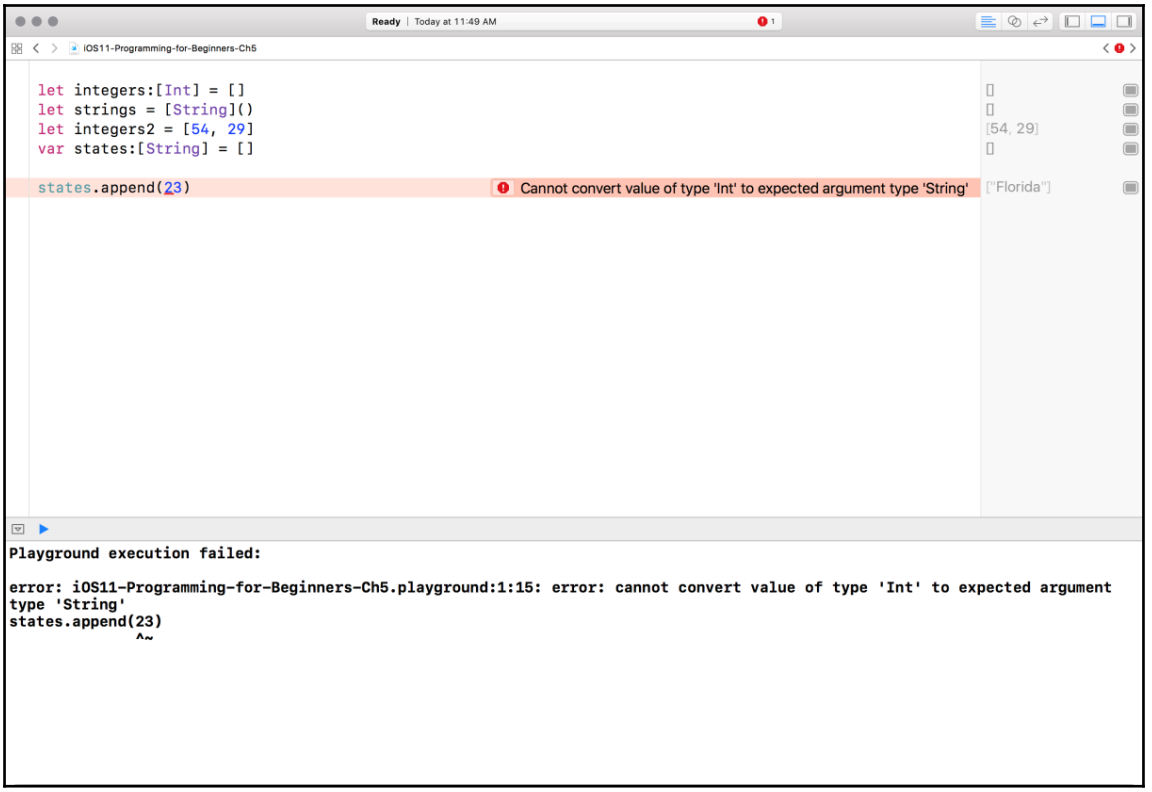
0	45
1	66
2	23
3	10
4	88

0	Florida
1	California
2	32
3	New York
4	99
5	true
6	9.0

0	Florida
1	California
2	32
3	New York
4	99
5	true
6	9.0

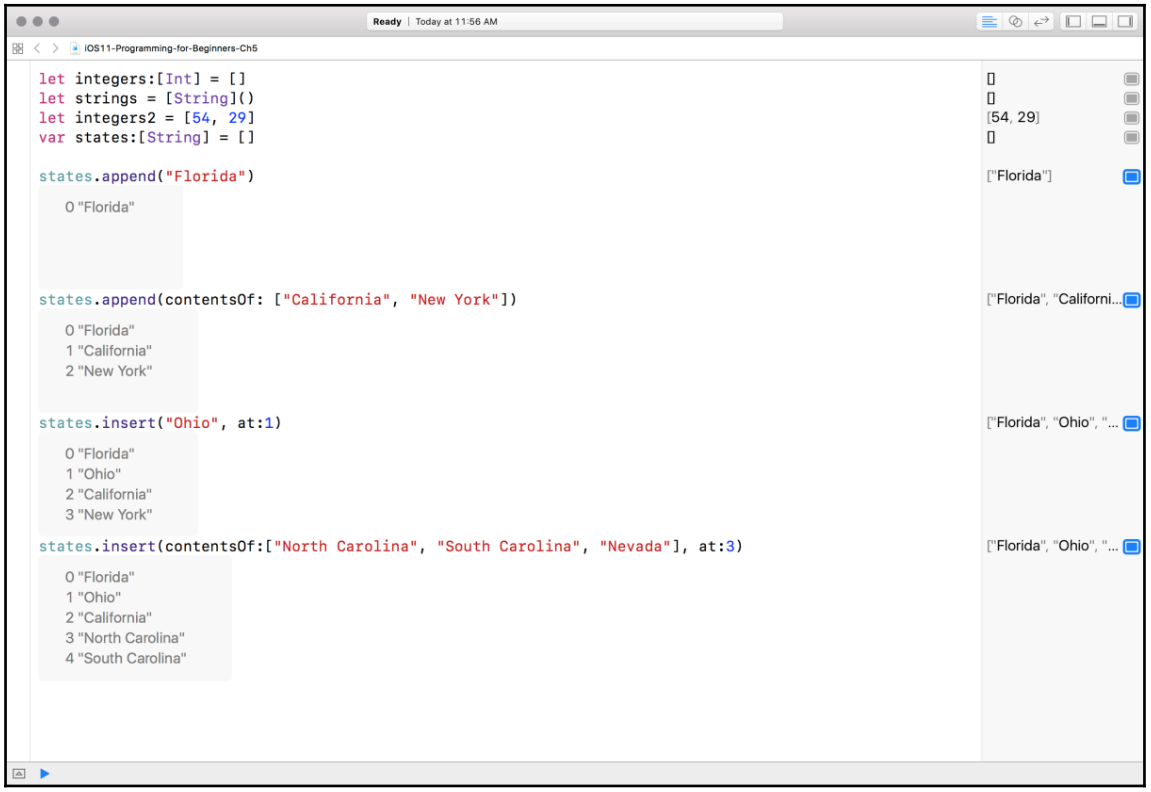


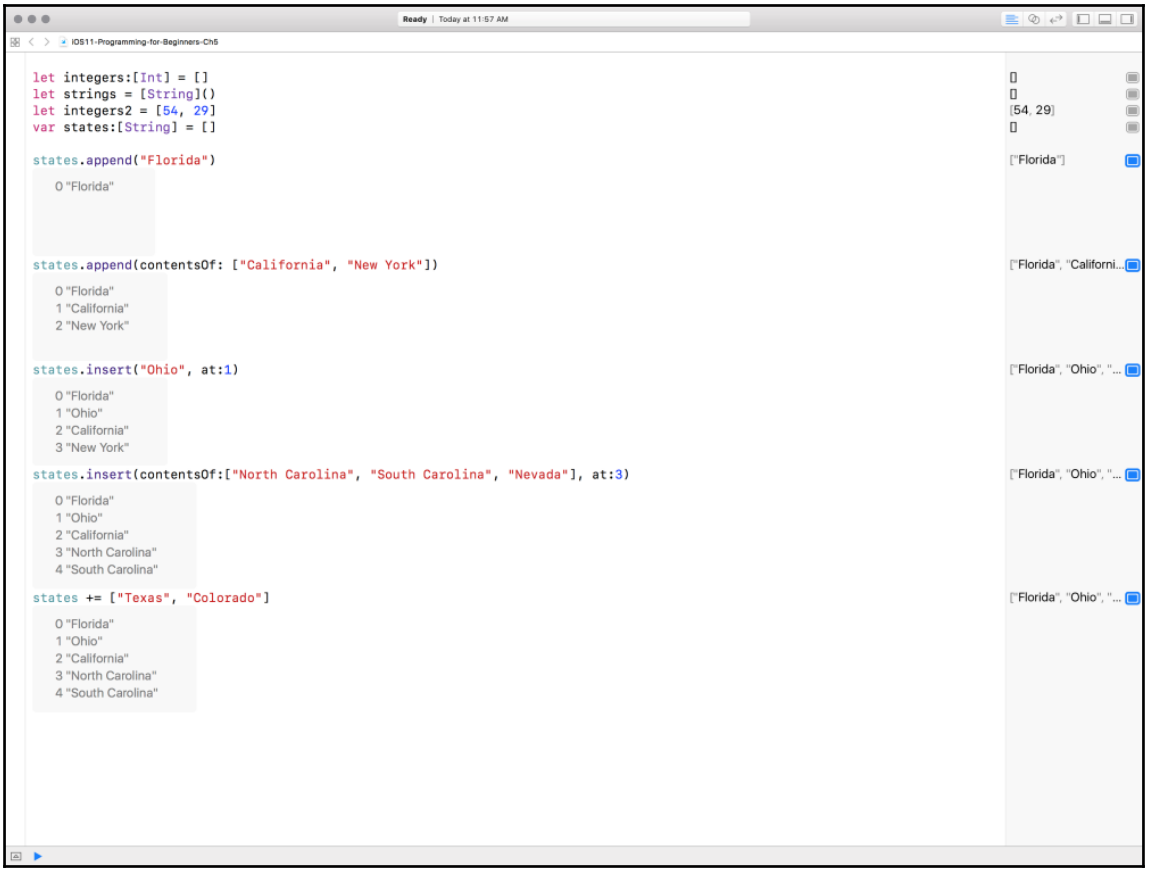


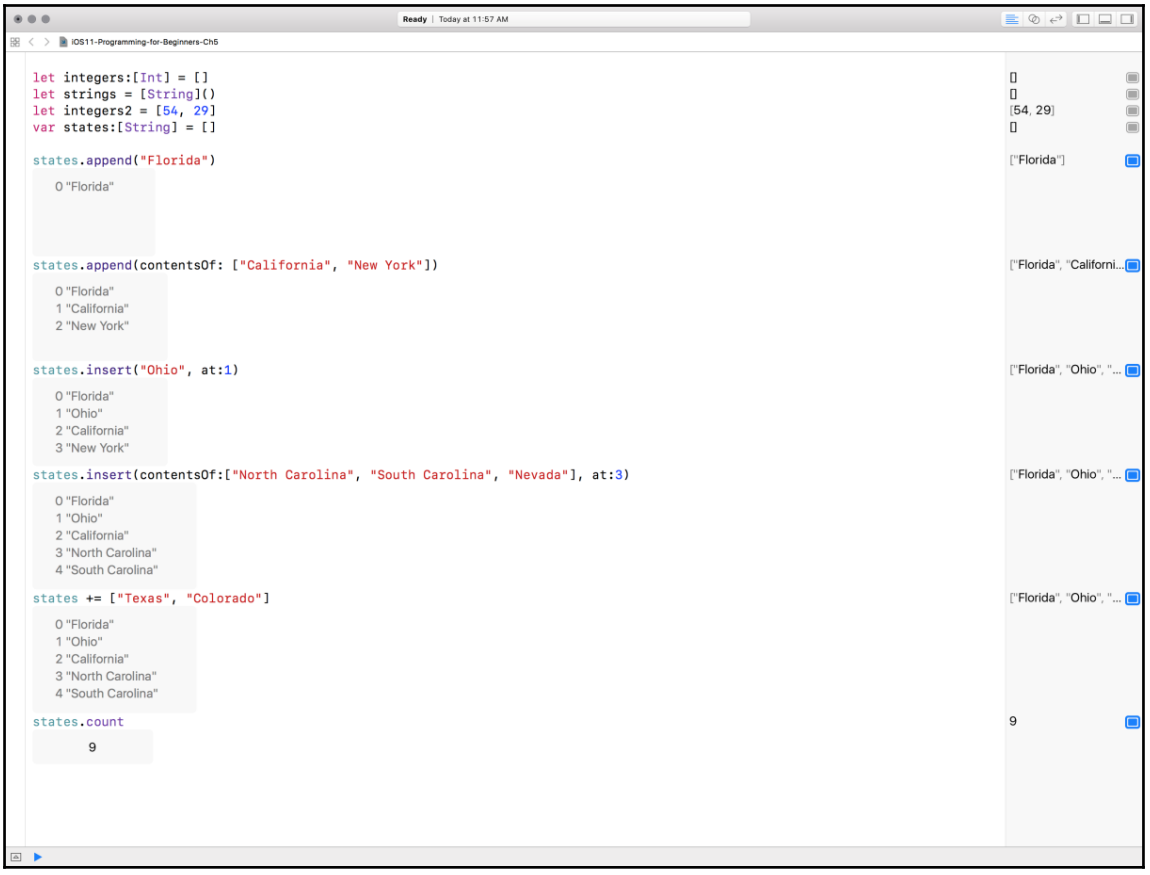

Ready | Today at 11:49 AM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append(23)
Cannot convert value of type 'Int' to expected argument type 'String'
[]
[]
[54, 29]
[]
["Florida"]
Playground execution failed:
error: iOS11-Programming-for-Beginners-Ch5.playground:1:15: error: cannot convert value of type 'Int' to expected argument type 'String'
states.append(23)
^~

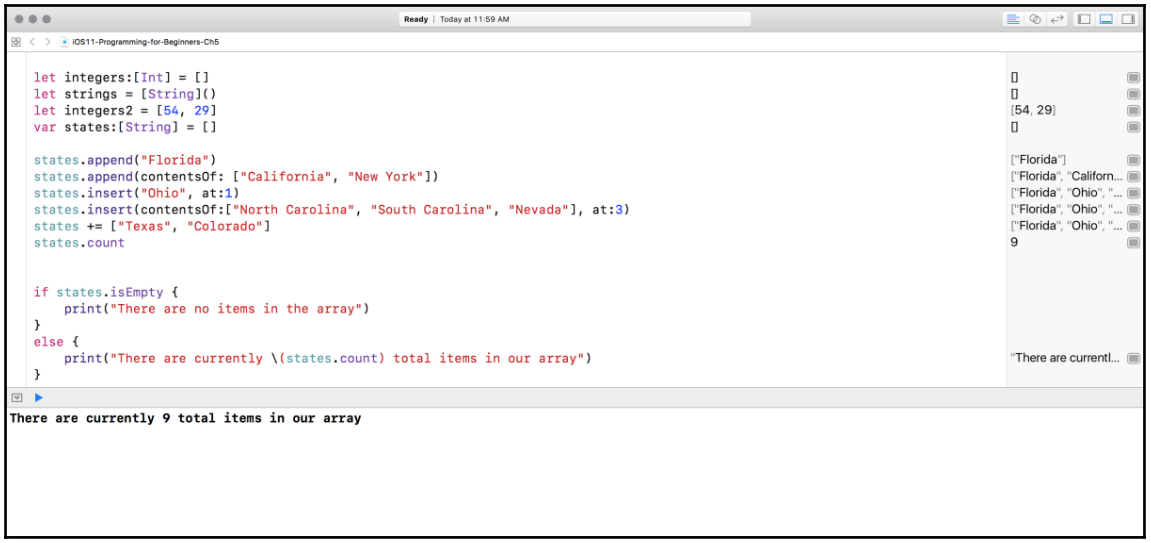

Ready | Today at 11:55 AM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
0 "Florida"
[]
[]
[54, 29]
[]
["Florida"]

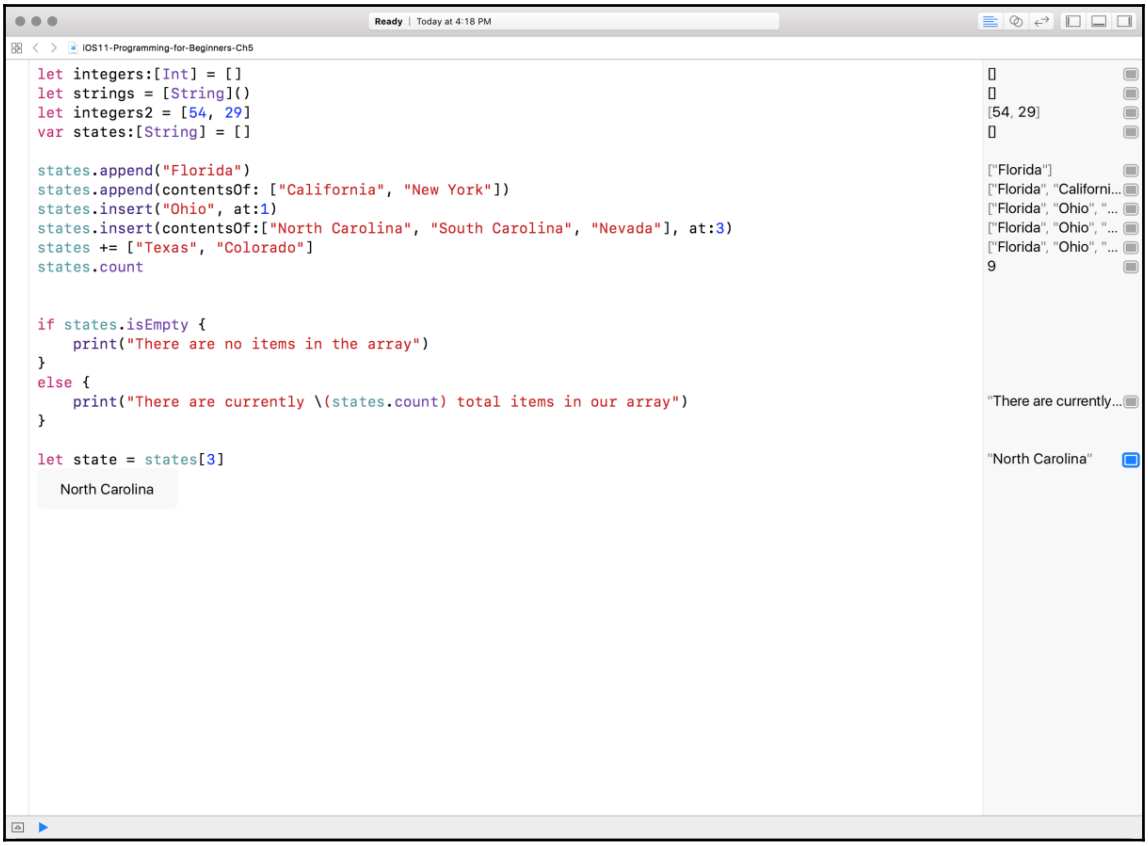

Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
0 "Florida"
states.append(contentsOf: ["California", "New York"])
0 "Florida"
1 "California"
2 "New York"
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californ...

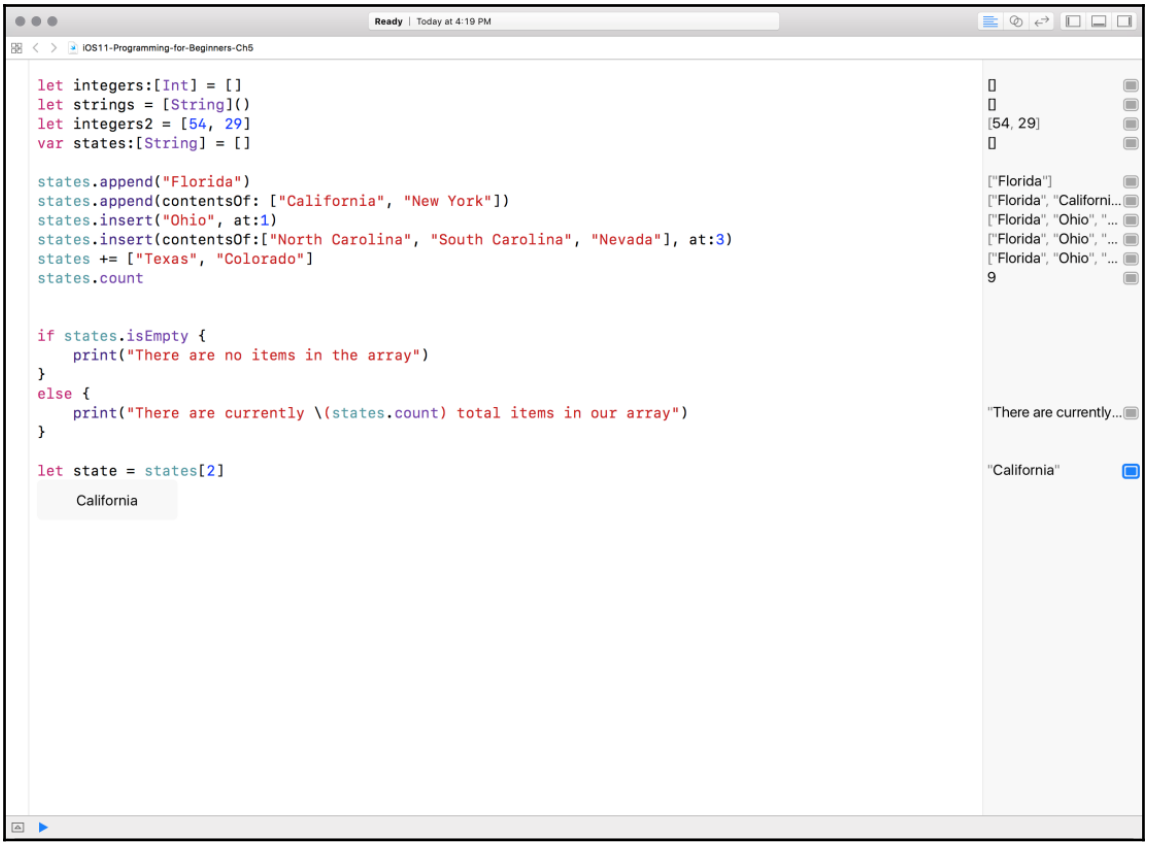

Ready | Today at 11:56 AM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
0 "Florida"
states.append(contentsOf: ["California", "New York"])
0 "Florida"
1 "California"
2 "New York"
states.insert("Ohio", at:1)
0 "Florida"
1 "Ohio"
2 "California"
3 "New York"
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
0 "Florida"
1 "Ohio"
2 "California"
3 "North Carolina"
4 "South Carolina"
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californi...
["Florida", "Ohio", "...
["Florida", "Ohio", "...

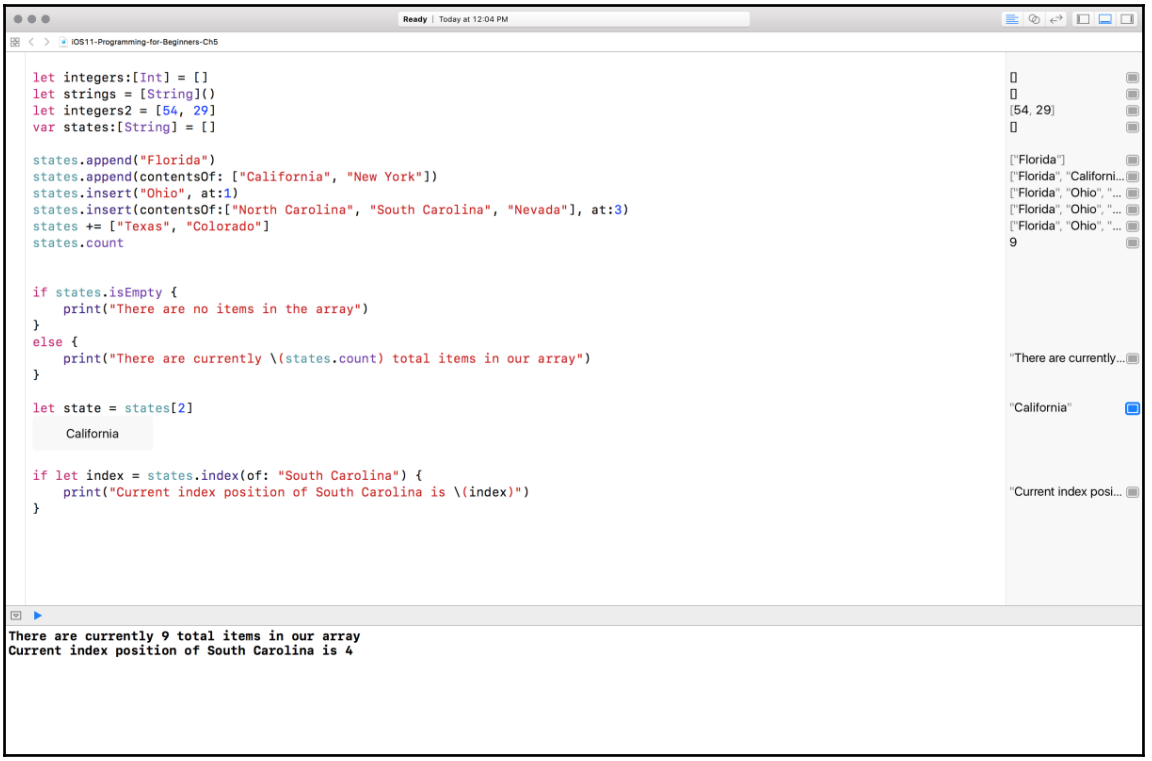

Ready | Today at 11:57 AM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
[]
[]
[54, 29]
[]
states.append("Florida")
0 "Florida"
["Florida"]
states.append(contentsOf: ["California", "New York"])
0 "Florida"
1 "California"
2 "New York"
["Florida", "Californi...
states.insert("Ohio", at:1)
0 "Florida"
1 "Ohio"
2 "California"
3 "New York"
["Florida", "Ohio", "...
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
0 "Florida"
1 "Ohio"
2 "California"
3 "North Carolina"
4 "South Carolina"
["Florida", "Ohio", "...
states += ["Texas", "Colorado"]
0 "Florida"
1 "Ohio"
2 "California"
3 "North Carolina"
4 "South Carolina"
["Florida", "Ohio", "...

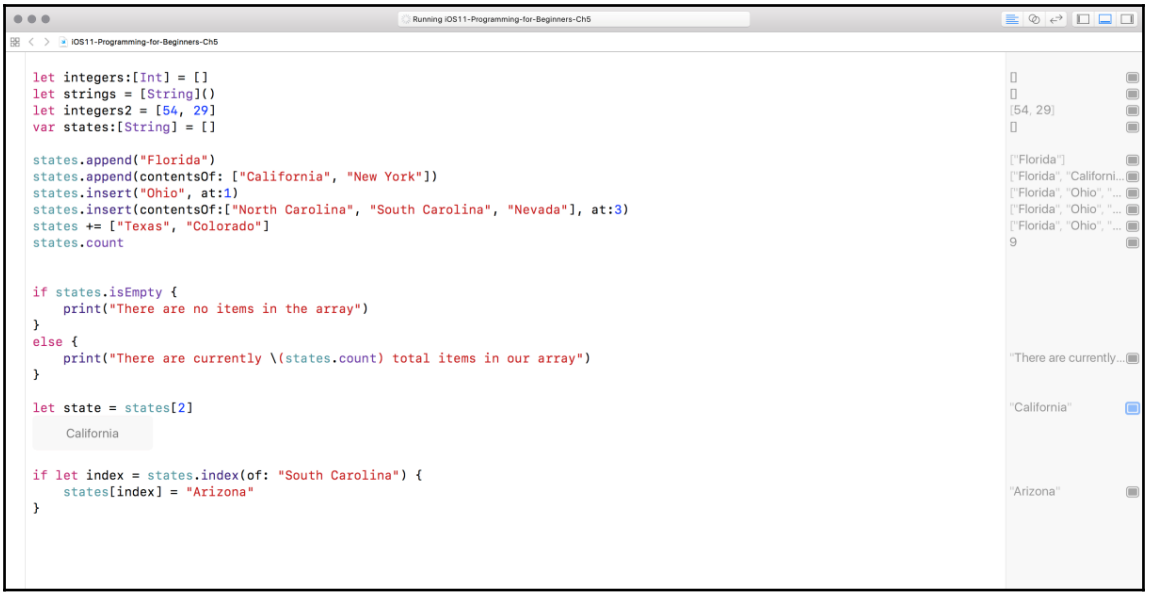

Ready | Today at 11:57 AM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
[]
[]
[54, 29]
[]
states.append("Florida")
0 "Florida"
["Florida"]
states.append(contentsOf: ["California", "New York"])
0 "Florida"
1 "California"
2 "New York"
["Florida", "Californi...
states.insert("Ohio", at:1)
0 "Florida"
1 "Ohio"
2 "California"
3 "New York"
["Florida", "Ohio", "...
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
0 "Florida"
1 "Ohio"
2 "California"
3 "North Carolina"
4 "South Carolina"
["Florida", "Ohio", "...
states += ["Texas", "Colorado"]
0 "Florida"
1 "Ohio"
2 "California"
3 "North Carolina"
4 "South Carolina"
["Florida", "Ohio", "...
states.count
9
9

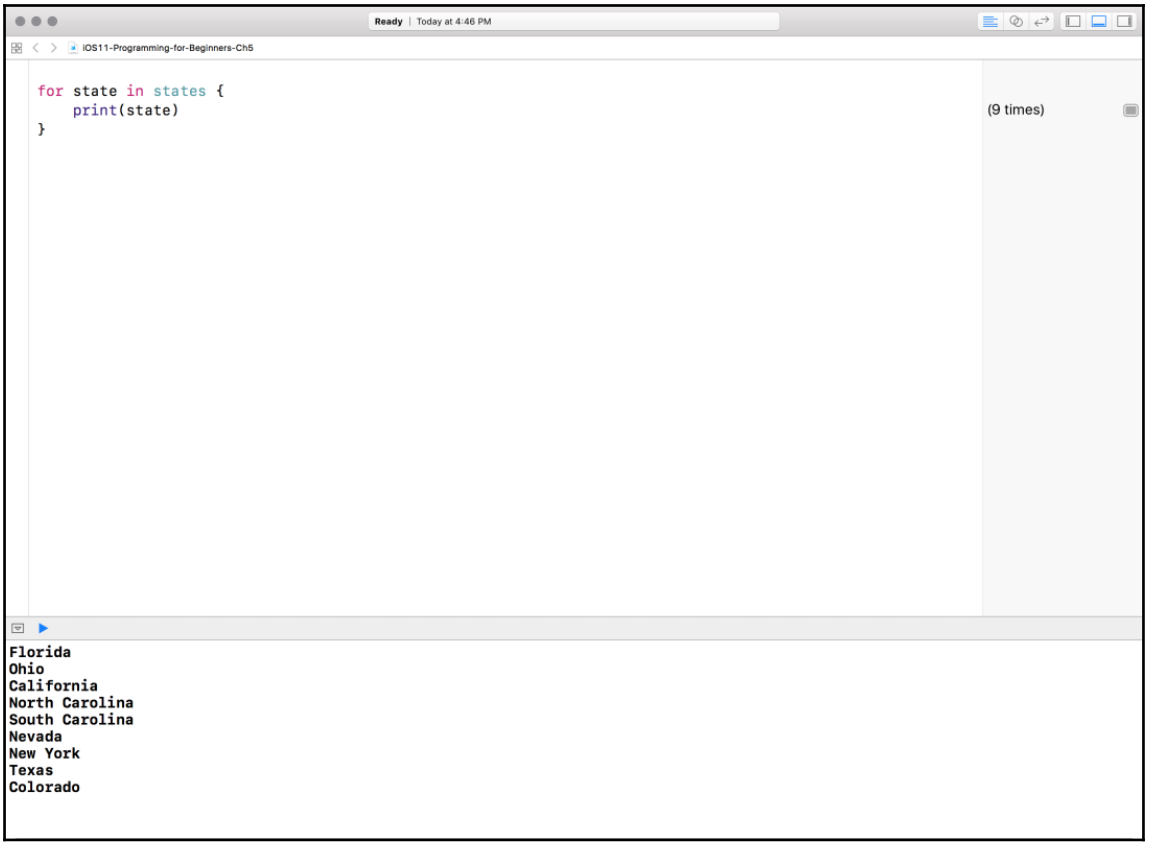

Ready | Today at 11:59 AM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
states.append(contentsOf: ["California", "New York"])
states.insert("Ohio", at:1)
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
states += ["Texas", "Colorado"]
states.count
if states.isEmpty {
print("There are no items in the array")
}
else {
print("There are currently \(states.count) total items in our array")
}
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californ...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
9
"There are currentl...
There are currently 9 total items in our array

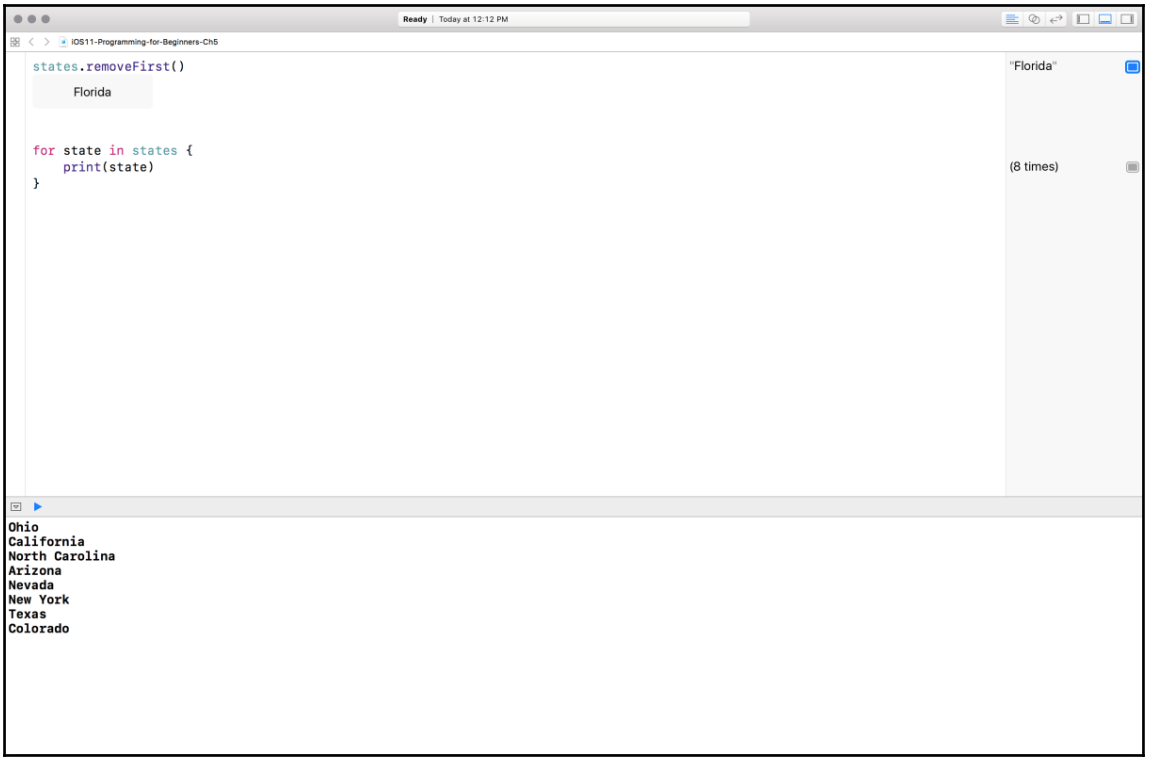

Ready | Today at 4:18 PM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
states.append(contentsOf: ["California", "New York"])
states.insert("Ohio", at:1)
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
states += ["Texas", "Colorado"]
states.count
if states.isEmpty {
print("There are no items in the array")
}
else {
print("There are currently \(states.count) total items in our array")
}
let state = states[3]
North Carolina
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californi...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
9
"There are currently...
"North Carolina"

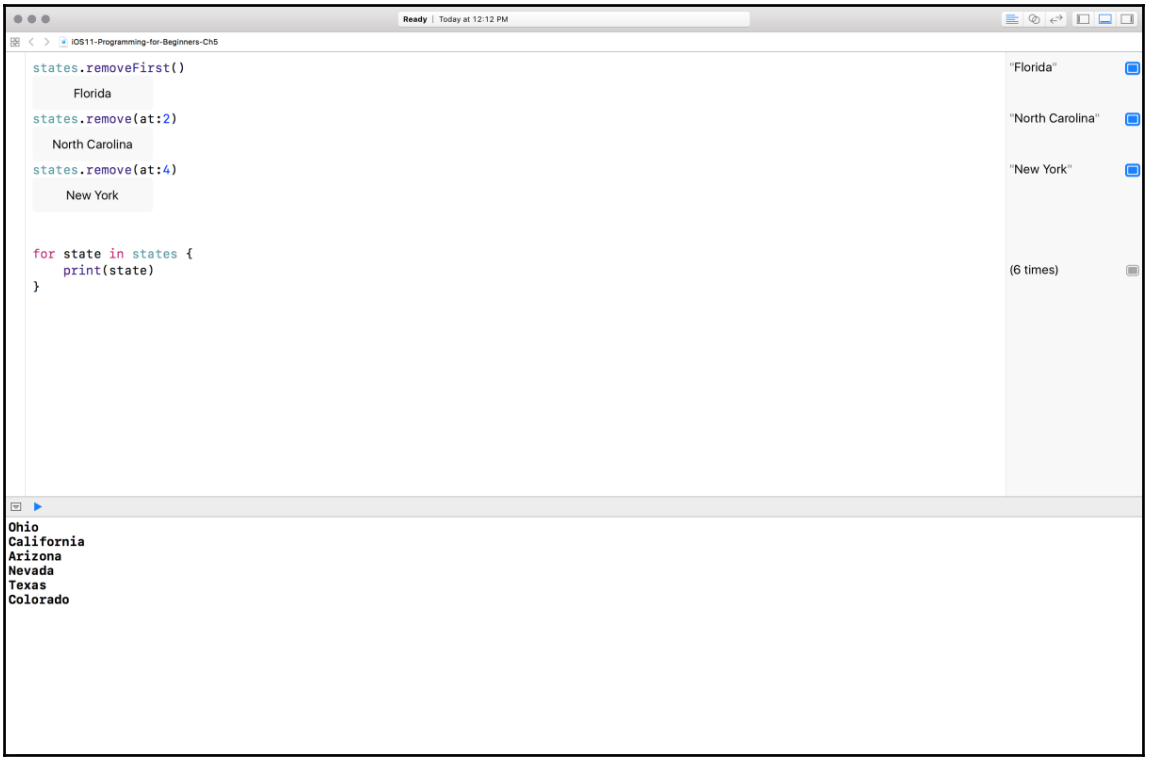

Ready | Today at 4:19 PM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
states.append(contentsOf: ["California", "New York"])
states.insert("Ohio", at:1)
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
states += ["Texas", "Colorado"]
states.count
if states.isEmpty {
print("There are no items in the array")
}
else {
print("There are currently \(states.count) total items in our array")
}
let state = states[2]
California
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californi...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
9
"There are currently...
"California"

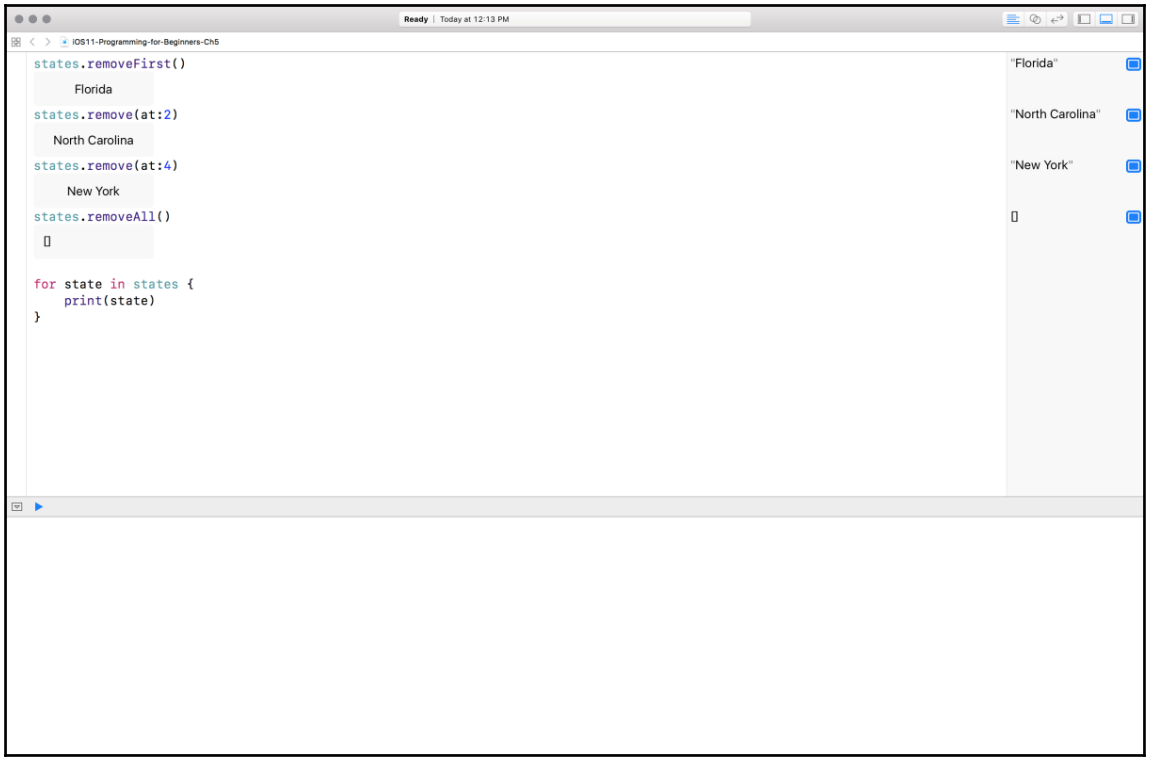

Ready | Today at 12:04 PM
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
states.append(contentsOf: ["California", "New York"])
states.insert("Ohio", at:1)
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
states += ["Texas", "Colorado"]
states.count
if states.isEmpty {
print("There are no items in the array")
}
else {
print("There are currently \(states.count) total items in our array")
}
let state = states[2]
California
if let index = states.index(of: "South Carolina") {
print("Current index position of South Carolina is \(index)")
}
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californi...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
9
"There are currently...
"California"
"Current index posi...
There are currently 9 total items in our array
Current index position of South Carolina is 4

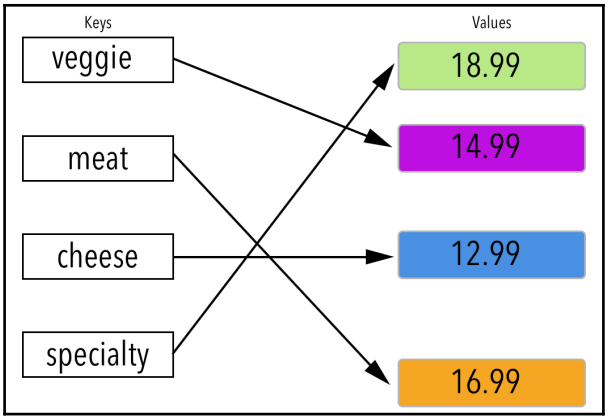

Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []
states.append("Florida")
states.append(contentsOf: ["California", "New York"])
states.insert("Ohio", at:1)
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)
states += ["Texas", "Colorado"]
states.count
if states.isEmpty {
print("There are no items in the array")
}
else {
print("There are currently \(states.count) total items in our array")
}
let state = states[2]
California
if let index = states.index(of: "South Carolina") {
states[index] = "Arizona"
}
[]
[]
[54, 29]
[]
["Florida"]
["Florida", "Californi...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
9
"There are currently...
"California"
"Arizona"


Ready | Today at 4:46 PM
iOS11-Programming-for-Beginners-Ch5
for state in states {
print(state)
}
(9 times)
Florida
Ohio
California
North Carolina
South Carolina
Nevada
New York
Texas
Colorado

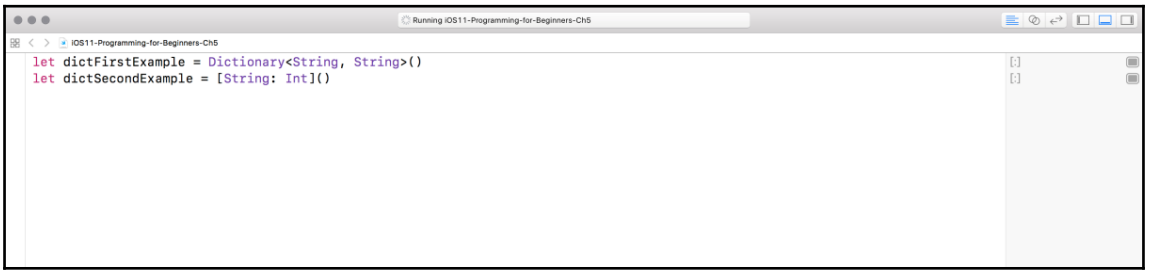

Ready | Today at 12:12 PM
IOS11-Programming-for-Beginners-Ch5
states.removeFirst()
Florida
"Florida"
for state in states {
print(state)
}
(8 times)
Ohio
California
North Carolina
Arizona
Nevada
New York
Texas
Colorado


Ready | Today at 12:12 PM
iOS11-Programming-for-Beginners-Ch5
states.removeFirst()
Florida
"Florida"
states.remove(at:2)
North Carolina
"North Carolina"
states.remove(at:4)
New York
"New York"
for state in states {
print(state)
}
(6 times)
Ohio
California
Arizona
Nevada
Texas
Colorado

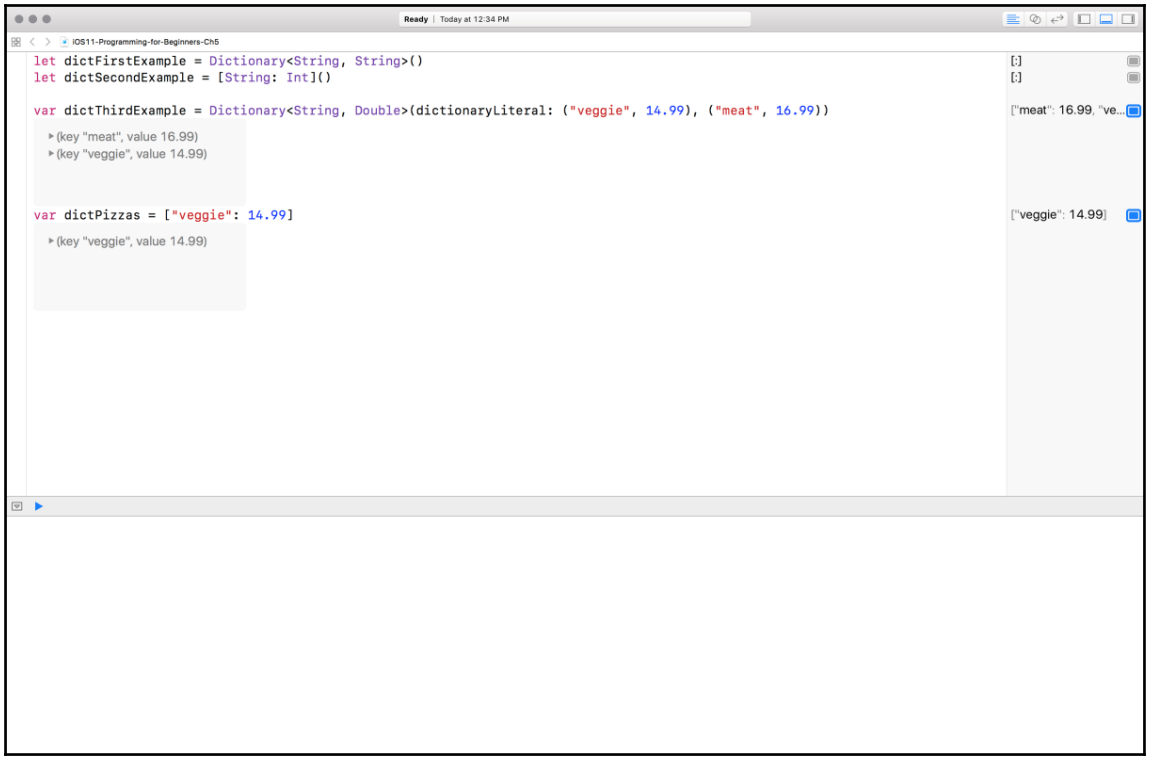

Ready | Today at 12:13 PM
iOS11-Programming-for-Beginners-Ch5
states.removeFirst()
Florida
states.remove(at:2)
North Carolina
states.remove(at:4)
New York
states.removeAll()
[]
for state in states {
print(state)
}
"Florida"
"North Carolina"
"New York"
[]


Keys
Values
veggie
meat
cheese
specialty
18.99
14.99
12.99
16.99

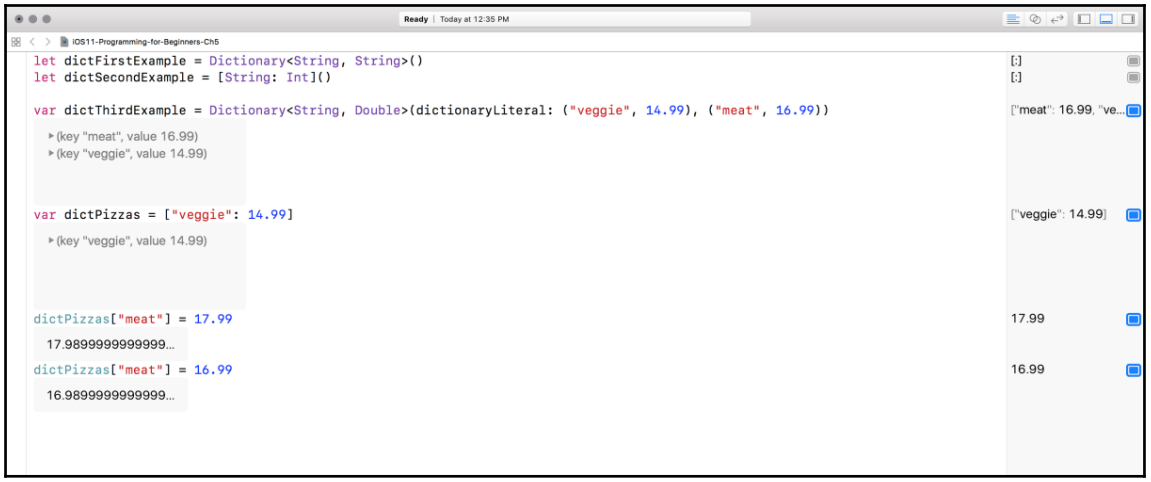

Ready | Today at 12:31 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
[:]

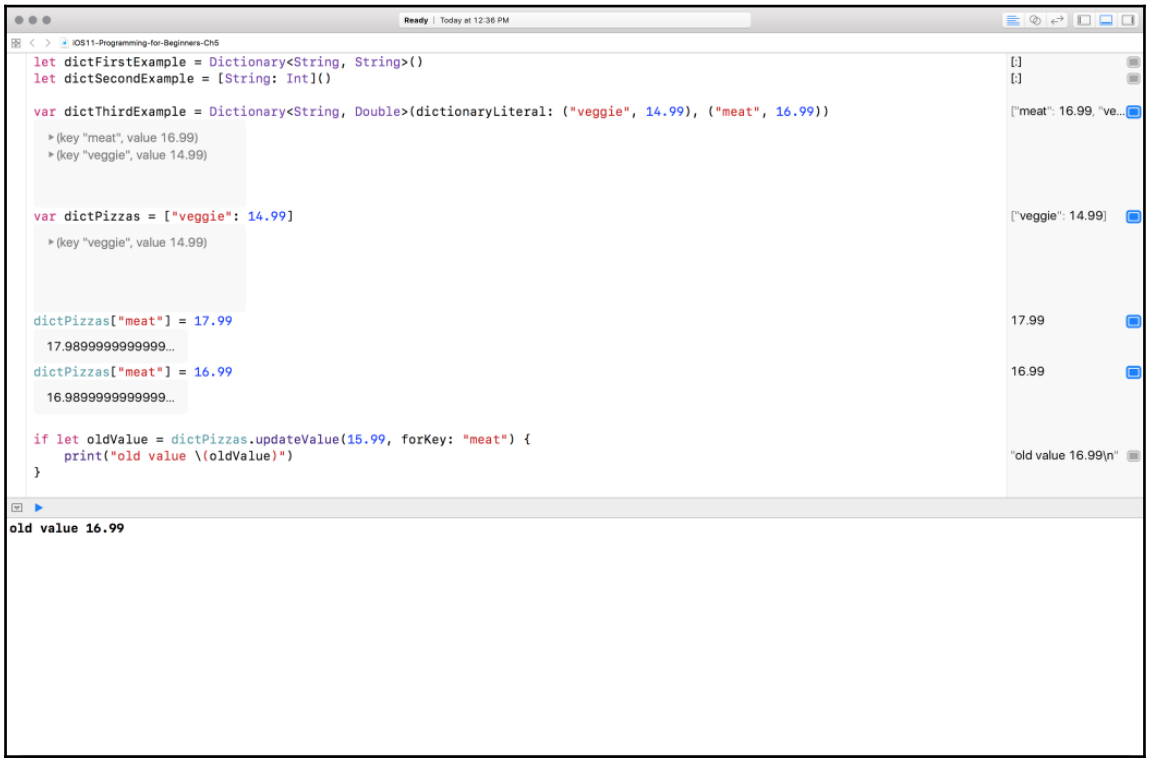

Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
[:]
[:]

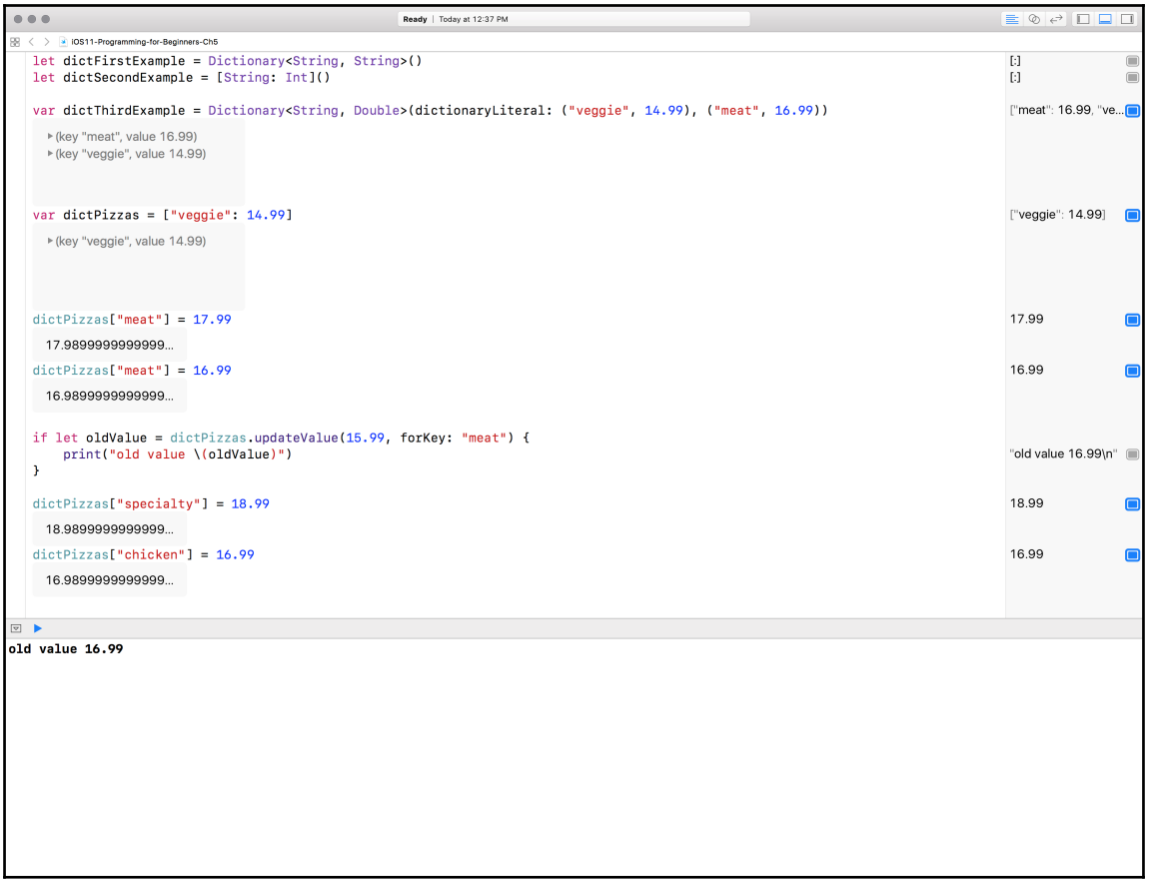

Ready | Today at 12:33 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
(key "meat", value 16.99)
(key "veggie", value 14.99)
[:]
[:]
["meat": 16.99, "ve...


Ready | Today at 12:34 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
(key "meat", value 16.99)
(key "veggie", value 14.99)
var dictPizzas = ["veggie": 14.99]
(key "veggie", value 14.99)
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]


Ready | Today at 12:34 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
(key "meat", value 16.99)
(key "veggie", value 14.99)
var dictPizzas = ["veggie": 14.99]
(key "veggie", value 14.99)
dictPizzas["meat"] = 17.99
17.9899999999999...
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99

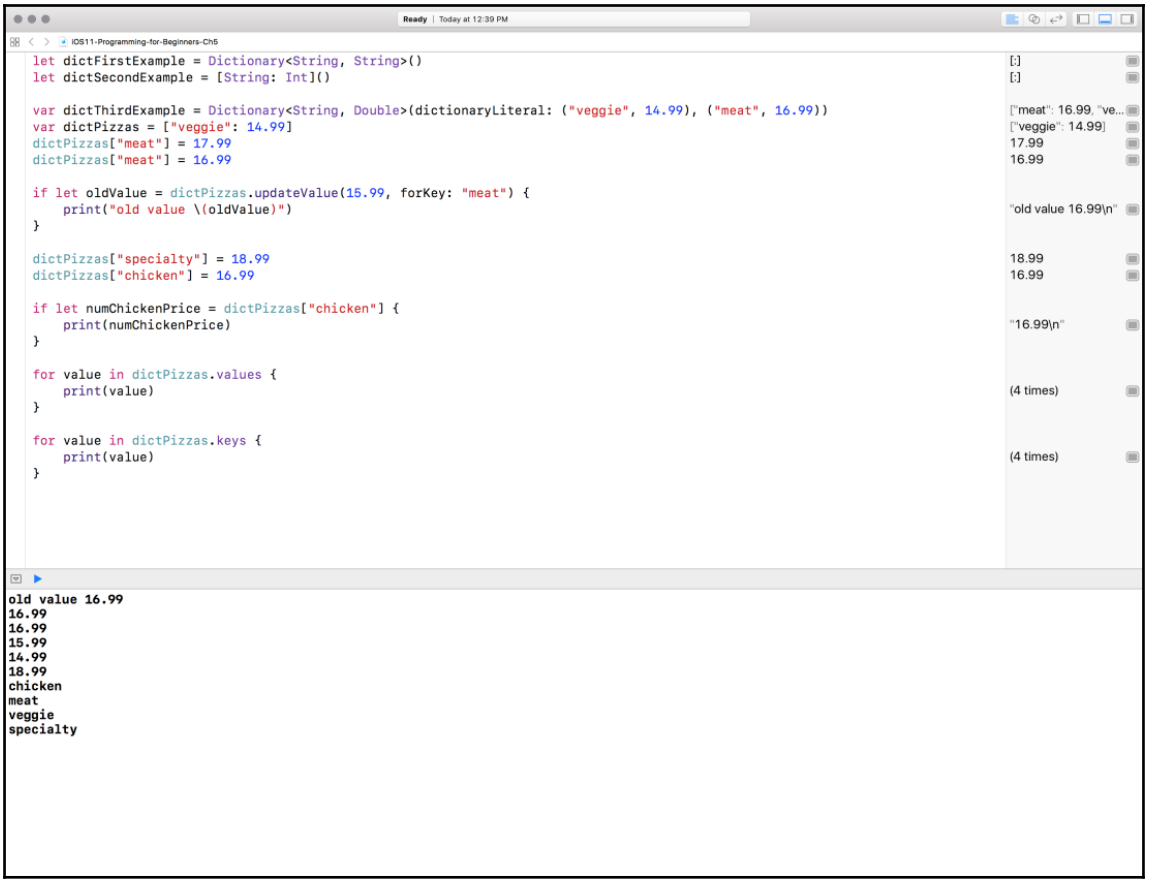

Ready | Today at 12:35 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
(key "meat", value 16.99)
(key "veggie", value 14.99)
var dictPizzas = ["veggie": 14.99]
(key "veggie", value 14.99)
dictPizzas["meat"] = 17.99
17.9899999999999...
dictPizzas["meat"] = 16.99
16.9899999999999...
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99

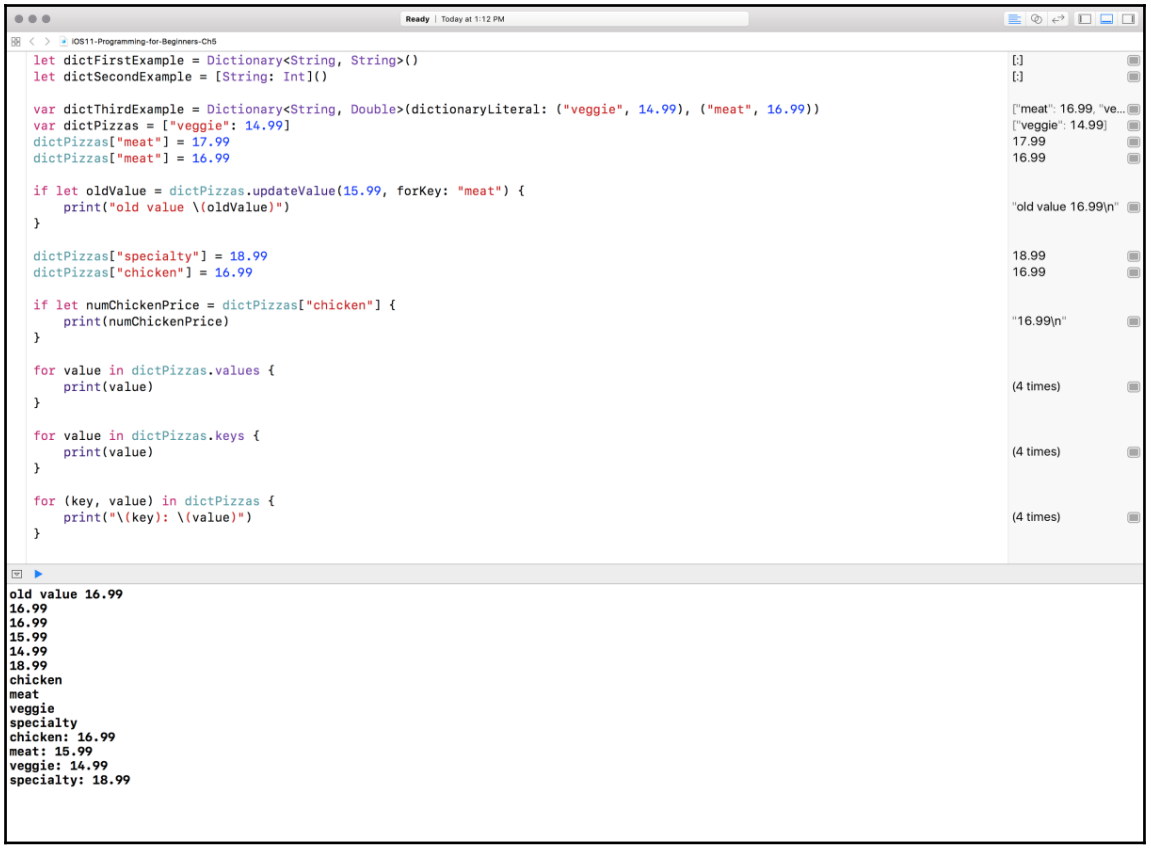

Ready | Today at 12:36 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
(key "meat", value 16.99)
(key "veggie", value 14.99)
var dictPizzas = ["veggie": 14.99]
(key "veggie", value 14.99)
dictPizzas["meat"] = 17.99
17.9899999999999...
dictPizzas["meat"] = 16.99
16.9899999999999...
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
old value 16.99

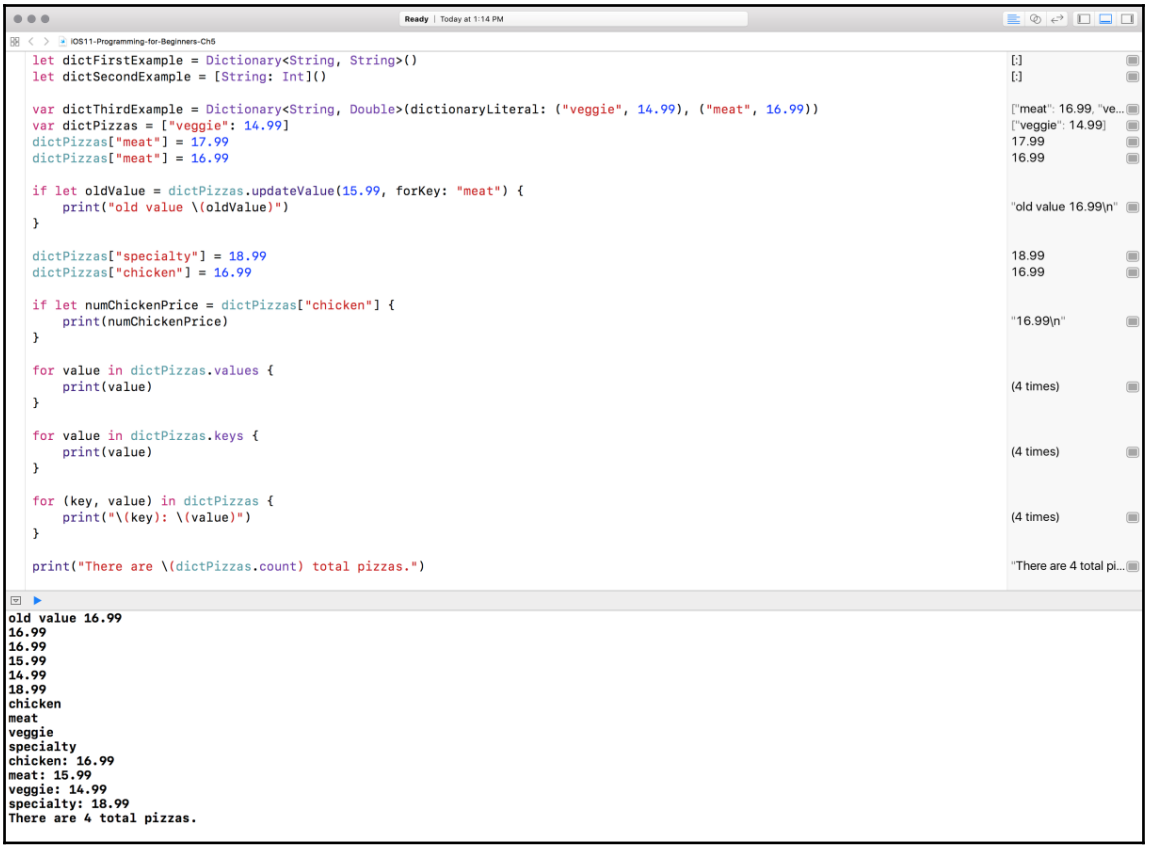

Ready | Today at 12:37 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
(key "meat", value 16.99)
(key "veggie", value 14.99)
var dictPizzas = ["veggie": 14.99]
(key "veggie", value 14.99)
dictPizzas["meat"] = 17.99
17.9899999999999...
dictPizzas["meat"] = 16.99
16.9899999999999...
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
dictPizzas["specialty"] = 18.99
18.9899999999999...
dictPizzas["chicken"] = 16.99
16.9899999999999...
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
old value 16.99

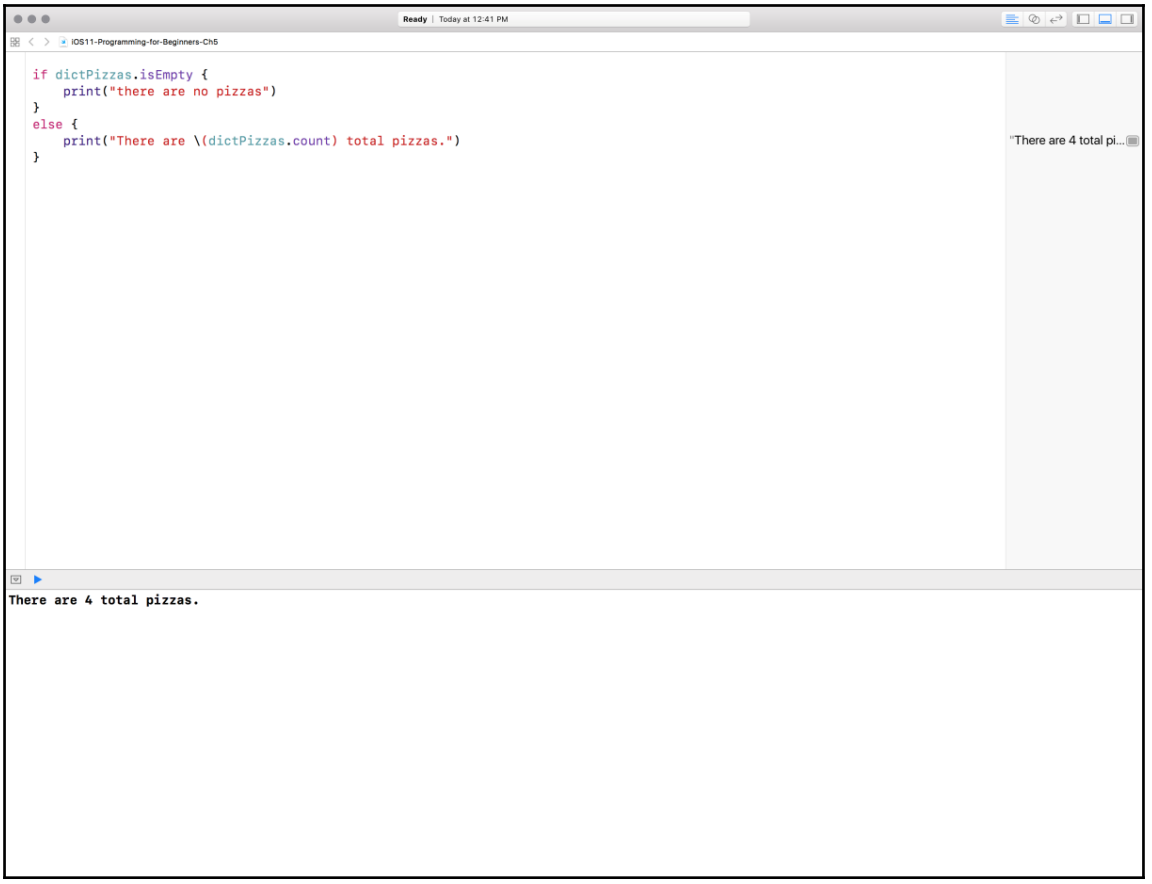

Ready | Today at 12:38 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
var dictPizzas = ["veggie": 14.99]
dictPizzas["meat"] = 17.99
dictPizzas["meat"] = 16.99
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
dictPizzas["specialty"] = 18.99
dictPizzas["chicken"] = 16.99
if let numChickenPrice = dictPizzas["chicken"] {
print(numChickenPrice)
}
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
"16.99\n"
old value 16.99
16.99

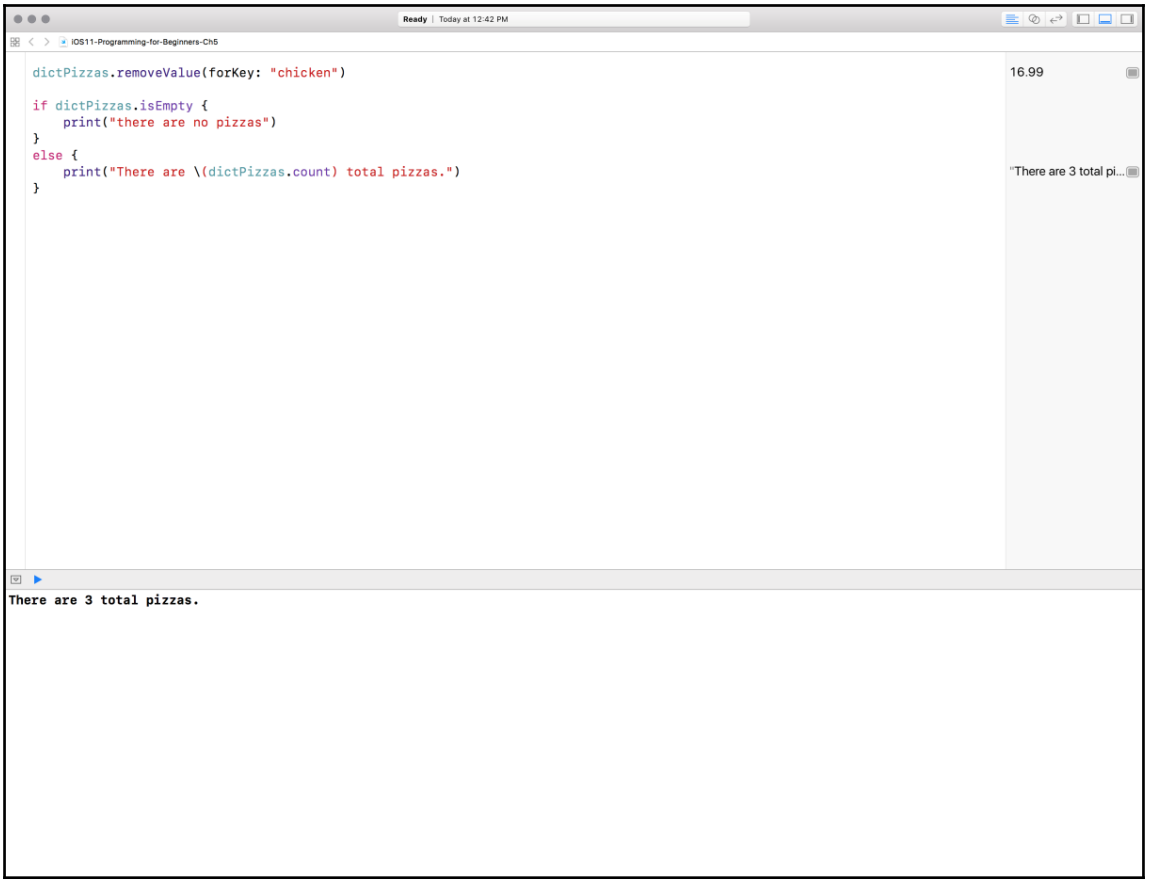

Ready | Today at 12:38 PM
IOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
var dictPizzas = ["veggie": 14.99]
dictPizzas["meat"] = 17.99
dictPizzas["meat"] = 16.99
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
dictPizzas["specialty"] = 18.99
dictPizzas["chicken"] = 16.99
if let numChickenPrice = dictPizzas["chicken"] {
print(numChickenPrice)
}
for value in dictPizzas.values {
print(value)
}
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
"16.99\n"
(4 times)
old value 16.99
16.99
16.99
15.99
14.99
18.99

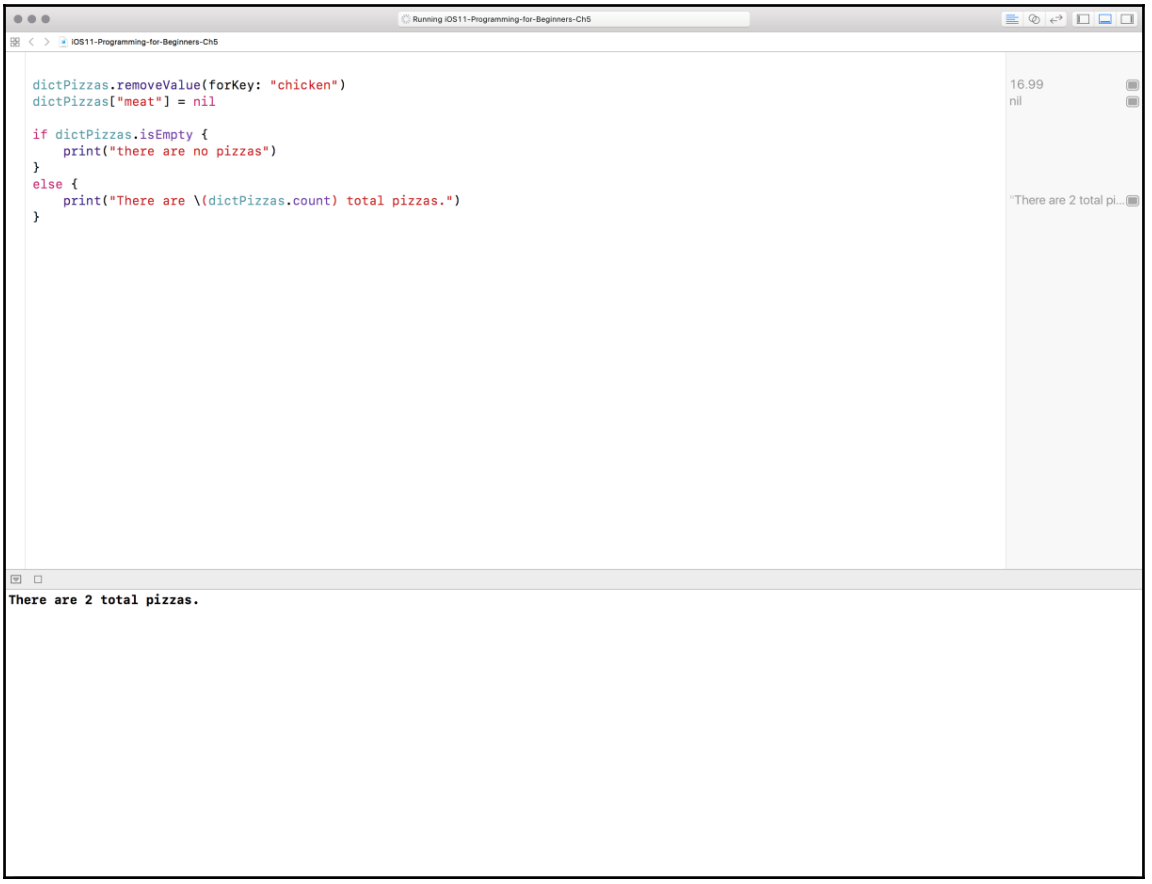

Ready | Today at 12:39 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
var dictPizzas = ["veggie": 14.99]
dictPizzas["meat"] = 17.99
dictPizzas["meat"] = 16.99
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
dictPizzas["specialty"] = 18.99
dictPizzas["chicken"] = 16.99
if let numChickenPrice = dictPizzas["chicken"] {
print(numChickenPrice)
}
for value in dictPizzas.values {
print(value)
}
for value in dictPizzas.keys {
print(value)
}
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
"16.99\n"
(4 times)
(4 times)
old value 16.99
16.99
16.99
15.99
14.99
18.99
chicken
meat
veggie
specialty

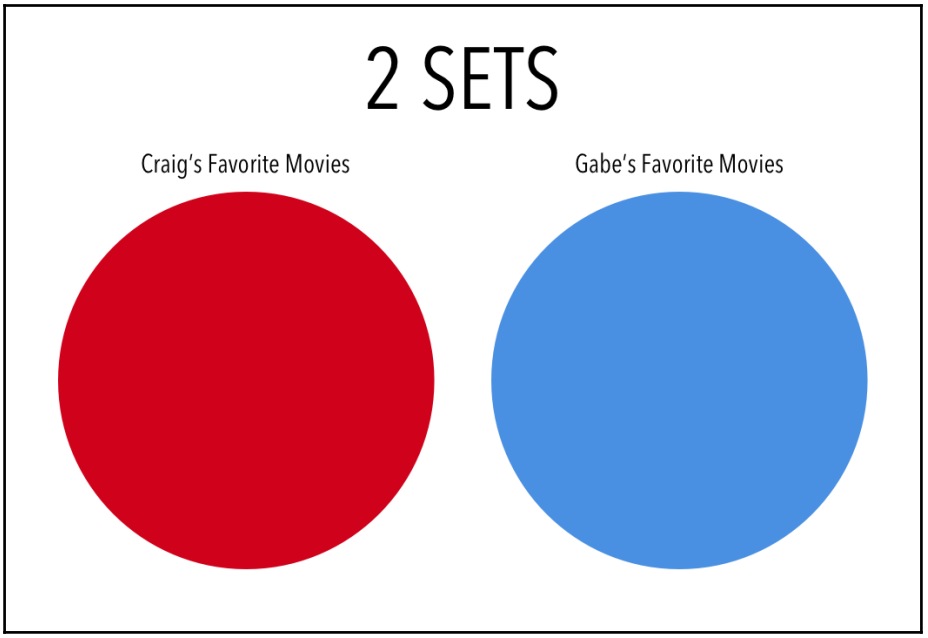

Ready | Today at 1:12 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
var dictPizzas = ["veggie": 14.99]
dictPizzas["meat"] = 17.99
dictPizzas["meat"] = 16.99
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
dictPizzas["specialty"] = 18.99
dictPizzas["chicken"] = 16.99
if let numChickenPrice = dictPizzas["chicken"] {
print(numChickenPrice)
}
for value in dictPizzas.values {
print(value)
}
for value in dictPizzas.keys {
print(value)
}
for (key, value) in dictPizzas {
print("\(key): \(value)")
}
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
"16.99\n"
(4 times)
(4 times)
(4 times)
old value 16.99
16.99
16.99
15.99
14.99
18.99
chicken
meat
veggie
specialty
chicken: 16.99
meat: 15.99
veggie: 14.99
specialty: 18.99


Ready | Today at 1:14 PM
iOS11-Programming-for-Beginners-Ch5
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
var dictPizzas = ["veggie": 14.99]
dictPizzas["meat"] = 17.99
dictPizzas["meat"] = 16.99
if let oldValue = dictPizzas.updateValue(15.99, forKey: "meat") {
print("old value \(oldValue)")
}
dictPizzas["specialty"] = 18.99
dictPizzas["chicken"] = 16.99
if let numChickenPrice = dictPizzas["chicken"] {
print(numChickenPrice)
}
for value in dictPizzas.values {
print(value)
}
for value in dictPizzas.keys {
print(value)
}
for (key, value) in dictPizzas {
print("\(key): \(value)")
}
print("There are \(dictPizzas.count) total pizzas.")
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
"16.99\n"
(4 times)
(4 times)
(4 times)
"There are 4 total pi...
old value 16.99
16.99
16.99
15.99
14.99
18.99
chicken
meat
veggie
specialty
chicken: 16.99
meat: 15.99
veggie: 14.99
specialty: 18.99
There are 4 total pizzas.


Ready | Today at 12:41 PM
iOS11-Programming-for-Beginners-Ch5
if dictPizzas.isEmpty {
print("there are no pizzas")
}
else {
print("There are \(dictPizzas.count) total pizzas.")
}
"There are 4 total pi...
There are 4 total pizzas.


Ready | Today at 12:42 PM
iOS11-Programming-for-Beginners-Ch5
dictPizzas.removeValue(forKey: "chicken")
if dictPizzas.isEmpty {
print("there are no pizzas")
}
else {
print("There are \(dictPizzas.count) total pizzas.")
}
16.99
"There are 3 total pi...
There are 3 total pizzas.


Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5
dictPizzas.removeValue(forKey: "chicken")
dictPizzas["meat"] = nil
if dictPizzas.isEmpty {
print("there are no pizzas")
}
else {
print("There are \(dictPizzas.count) total pizzas.")
}
16.99
nil
"There are 2 total pi...
There are 2 total pizzas.



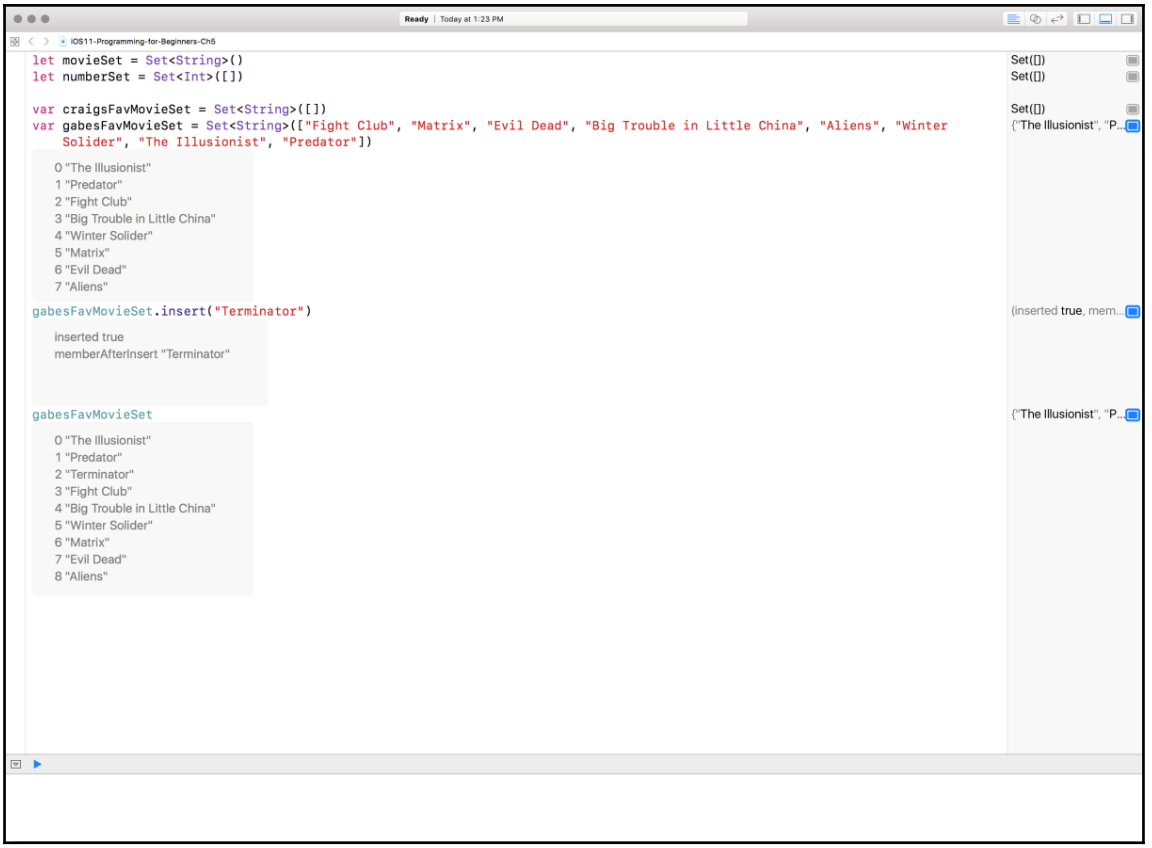

```
let movieSet = Set<String>()
```

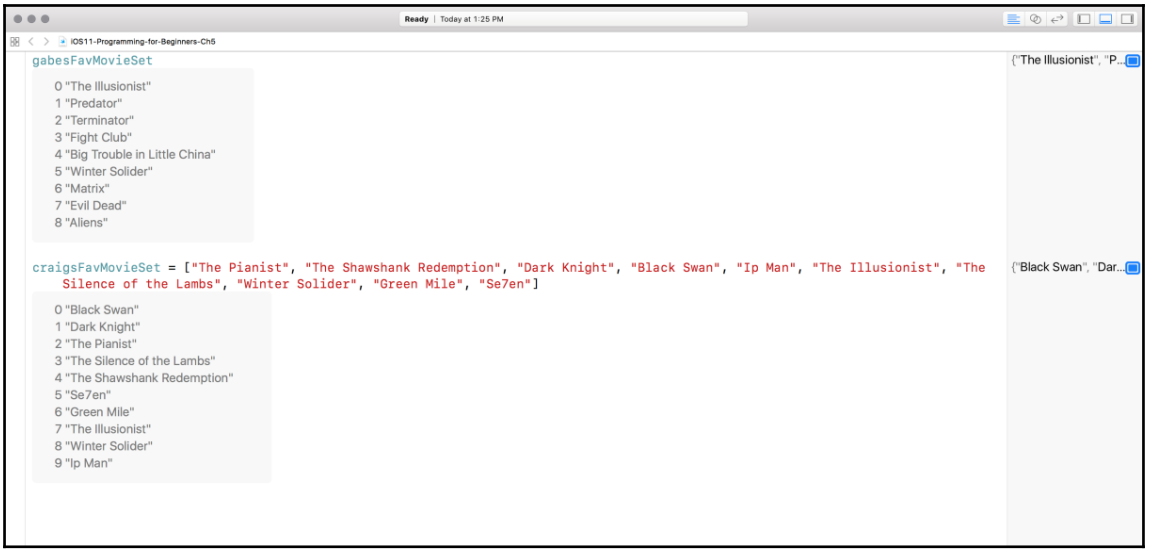
```
Set([])
```

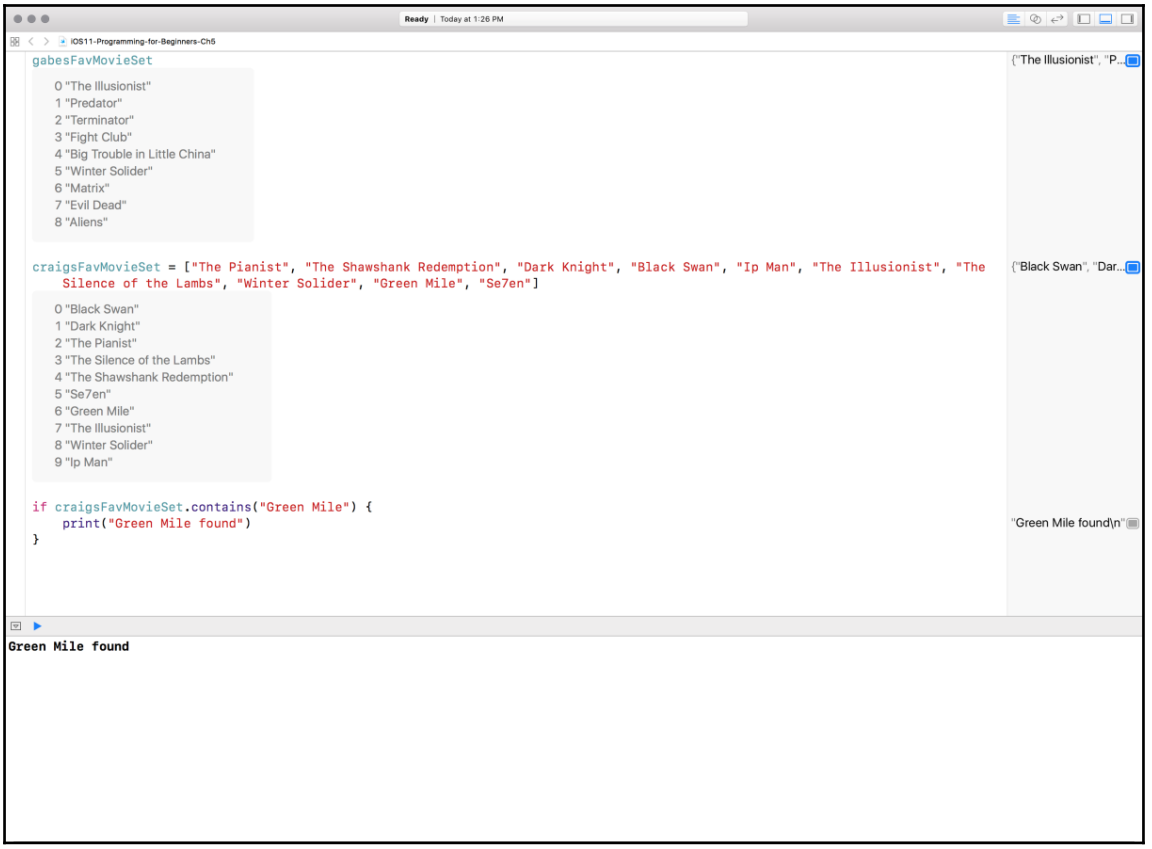
```
let movieSet = Set<String>()
let numberSet = Set<Int>([])
```

```
Set([])
Set([])
```

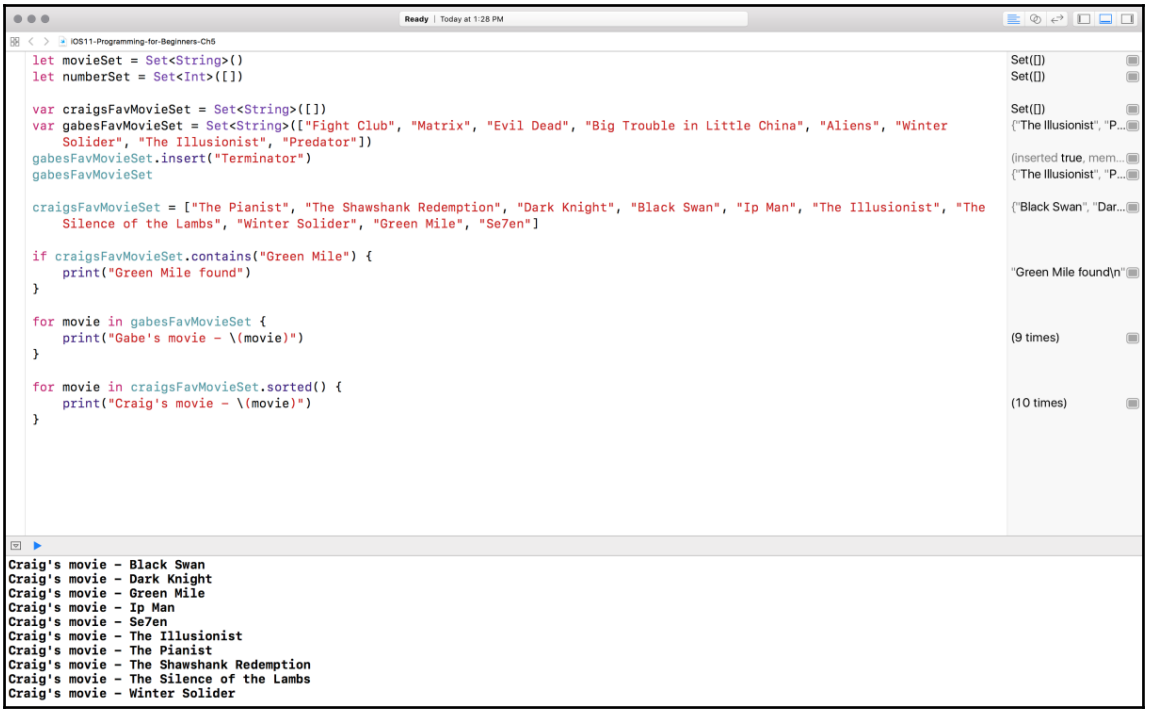

Ready | Today at 1:22 PM
iOS11-Programming-for-Beginners-Ch5
let movieSet = Set<String>()
let numberSet = Set<Int>([])
var craigsFavMovieSet = Set<String>([])
var gabesFavMovieSet = Set<String>(["Fight Club", "Matrix", "Evil Dead", "Big Trouble in Little China", "Aliens", "Winter Solider", "The Illusionist", "Predator"])
0 "The Illusionist"
1 "Predator"
2 "Fight Club"
3 "Big Trouble in Little China"
4 "Winter Solider"
5 "Matrix"
6 "Evil Dead"
7 "Aliens"
Set([])
Set([])
Set([])
{"The Illusionist", "...

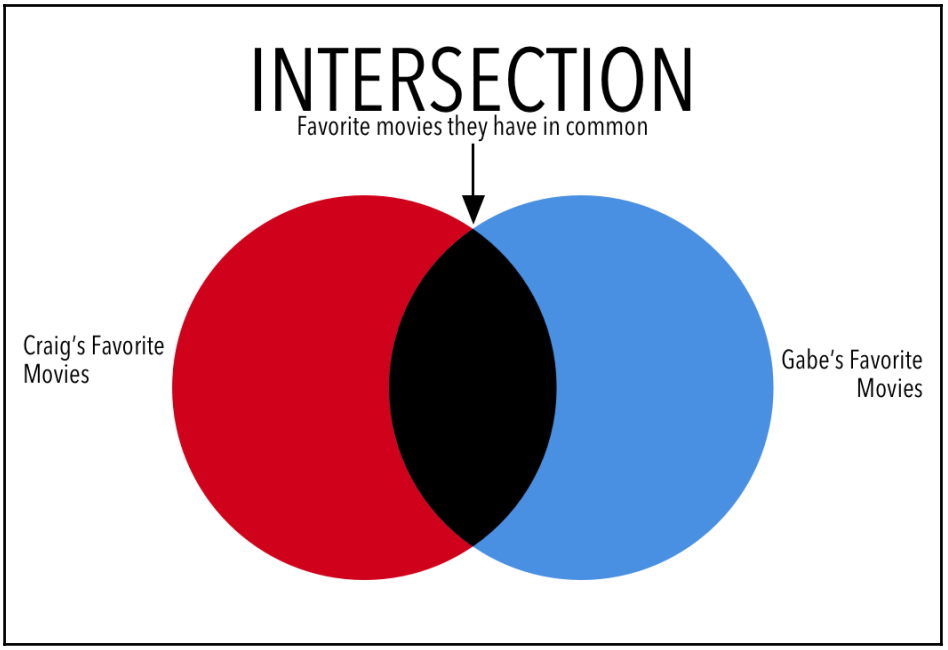

Ready | Today at 1:23 PM
iOS11-Programming-for-Beginners-Ch5
let movieSet = Set<String>()
let numberSet = Set<Int>([])
var craigsFavMovieSet = Set<String>([])
var gabesFavMovieSet = Set<String>(["Fight Club", "Matrix", "Evil Dead", "Big Trouble in Little China", "Aliens", "Winter Solider", "The Illusionist", "Predator"])
0 "The Illusionist"
1 "Predator"
2 "Fight Club"
3 "Big Trouble in Little China"
4 "Winter Solider"
5 "Matrix"
6 "Evil Dead"
7 "Aliens"
gabesFavMovieSet.insert("Terminator")
inserted true
memberAfterInsert "Terminator"
gabesFavMovieSet
0 "The Illusionist"
1 "Predator"
2 "Terminator"
3 "Fight Club"
4 "Big Trouble in Little China"
5 "Winter Solider"
6 "Matrix"
7 "Evil Dead"
8 "Aliens"
Set([])
Set([])
Set([])
{"The Illusionist", "P...
(inserted true, mem...
{"The Illusionist", "P...

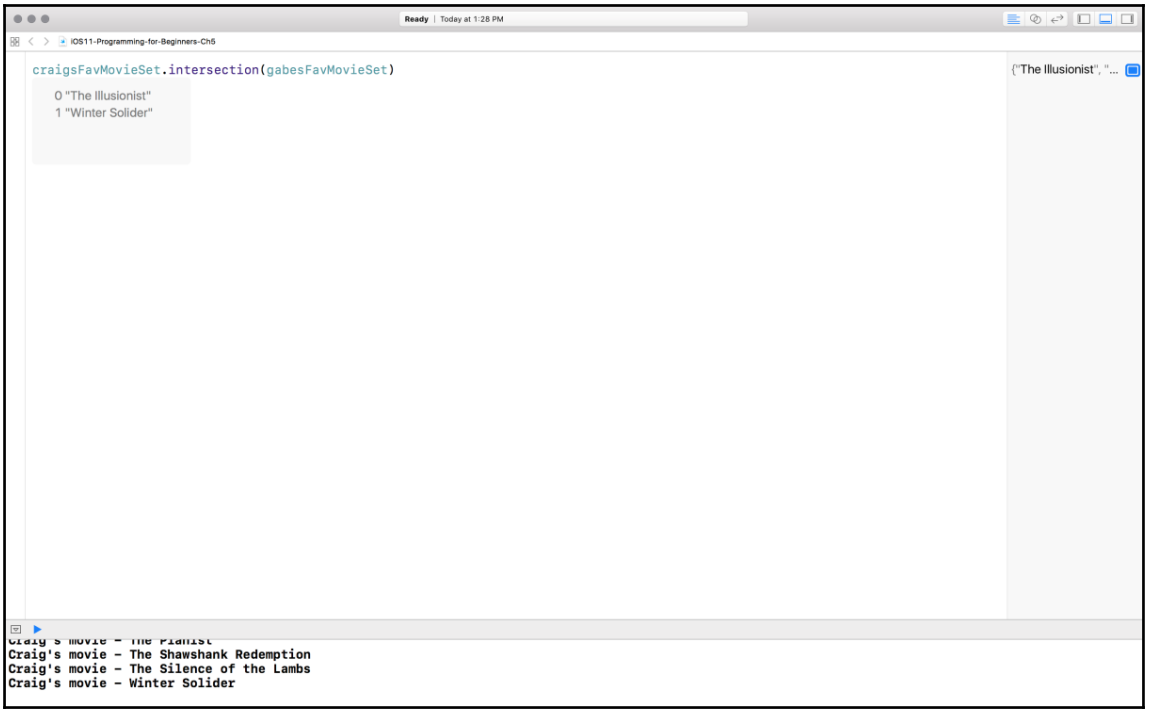

Ready | Today at 1:25 PM
iOS11-Programming-for-Beginners-Ch5
gabesFavMovieSet
0 "The Illusionist"
1 "Predator"
2 "Terminator"
3 "Fight Club"
4 "Big Trouble in Little China"
5 "Winter Solider"
6 "Matrix"
7 "Evil Dead"
8 "Aliens"
{"The Illusionist", "P...
craigsFavMovieSet = ["The Pianist", "The Shawshank Redemption", "Dark Knight", "Black Swan", "Ip Man", "The Illusionist", "The Silence of the Lambs", "Winter Solider", "Green Mile", "Se7en"]
0 "Black Swan"
1 "Dark Knight"
2 "The Pianist"
3 "The Silence of the Lambs"
4 "The Shawshank Redemption"
5 "Se7en"
6 "Green Mile"
7 "The Illusionist"
8 "Winter Solider"
9 "Ip Man"
{"Black Swan", "Dar...

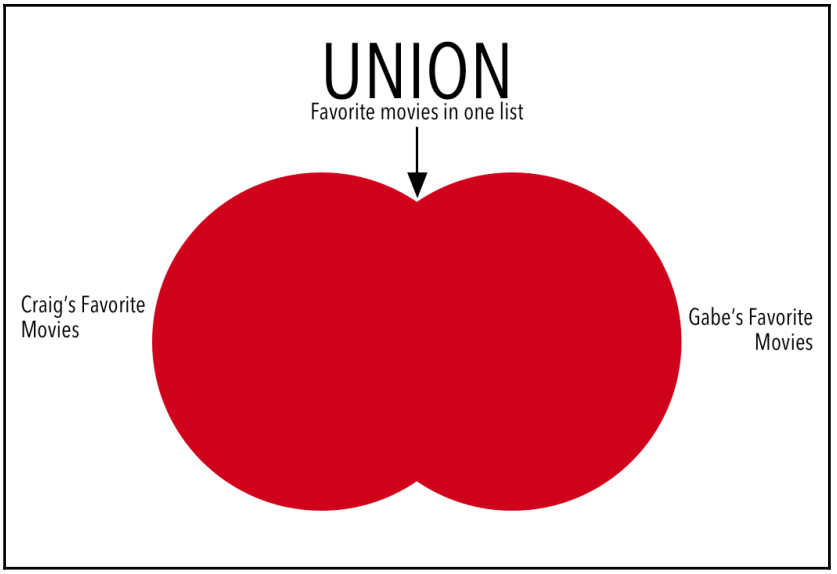

Ready | Today at 1:26 PM
iOS11-Programming-for-Beginners-Ch5
gabesFavMovieSet
0 "The Illusionist"
1 "Predator"
2 "Terminator"
3 "Fight Club"
4 "Big Trouble in Little China"
5 "Winter Solider"
6 "Matrix"
7 "Evil Dead"
8 "Aliens"
{"The Illusionist", "P...
craigsFavMovieSet = ["The Pianist", "The Shawshank Redemption", "Dark Knight", "Black Swan", "Ip Man", "The Illusionist", "The Silence of the Lambs", "Winter Solider", "Green Mile", "Se7en"]
0 "Black Swan"
1 "Dark Knight"
2 "The Pianist"
3 "The Silence of the Lambs"
4 "The Shawshank Redemption"
5 "Se7en"
6 "Green Mile"
7 "The Illusionist"
8 "Winter Solider"
9 "Ip Man"
{"Black Swan", "Dar...
if craigsFavMovieSet.contains("Green Mile") {
print("Green Mile found")
}
"Green Mile found\n"
Green Mile found

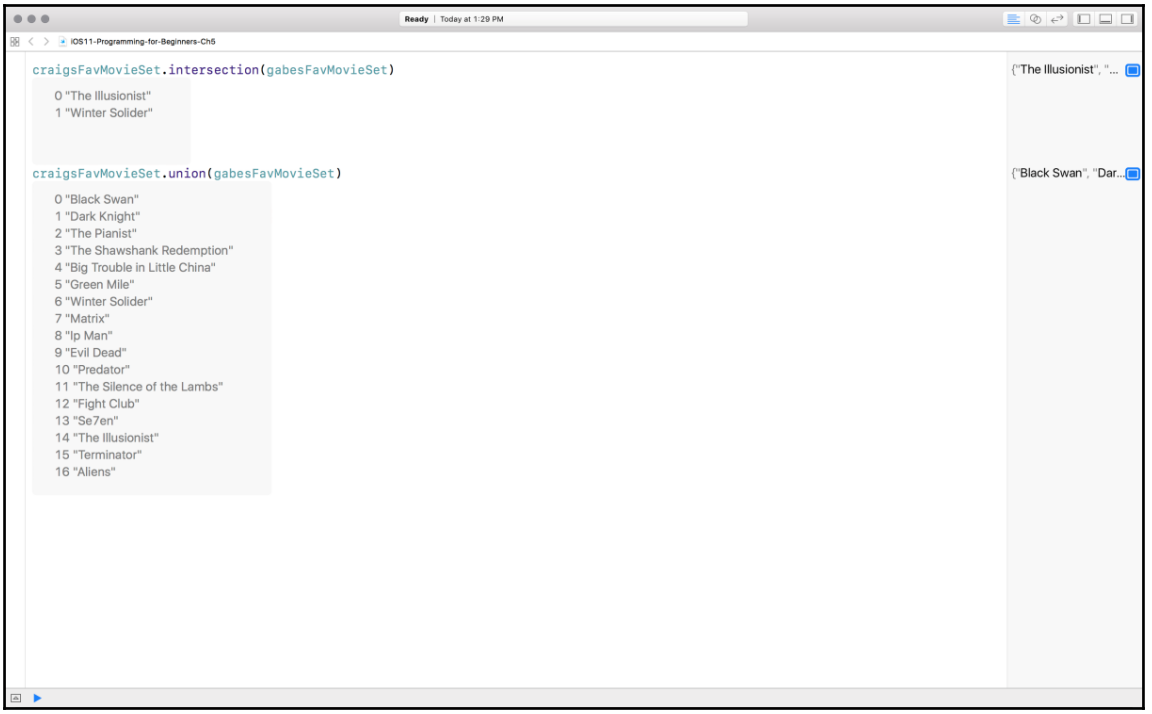

Ready | Today at 1:27 PM
iOS11-Programming-for-Beginners-Ch5
let movieSet = Set<String>()
let numberSet = Set<Int>([])
var craigsFavMovieSet = Set<String>([])
var gabesFavMovieSet = Set<String>(["Fight Club", "Matrix", "Evil Dead", "Big Trouble in Little China", "Aliens", "Winter Solider", "The Illusionist", "Predator"])
gabesFavMovieSet.insert("Terminator")
gabesFavMovieSet
craigsFavMovieSet = ["The Pianist", "The Shawshank Redemption", "Dark Knight", "Black Swan", "Ip Man", "The Illusionist", "The Silence of the Lambs", "Winter Solider", "Green Mile", "Se7en"]
if craigsFavMovieSet.contains("Green Mile") {
print("Green Mile found")
}
for movie in gabesFavMovieSet {
print("Gabe's movie - \(movie)")
}
Set([])
Set([])
Set([])
{"The Illusionist", "P...
(inserted true, mem...
{"The Illusionist", "P...
{"Black Swan", "Dar...
"Green Mile found\n"
(9 times)
Gabe's movie - The Illusionist
Gabe's movie - Predator
Gabe's movie - Terminator
Gabe's movie - Fight Club
Gabe's movie - Big Trouble in Little China
Gabe's movie - Winter Solider
Gabe's movie - Matrix
Gabe's movie - Evil Dead
Gabe's movie - Aliens

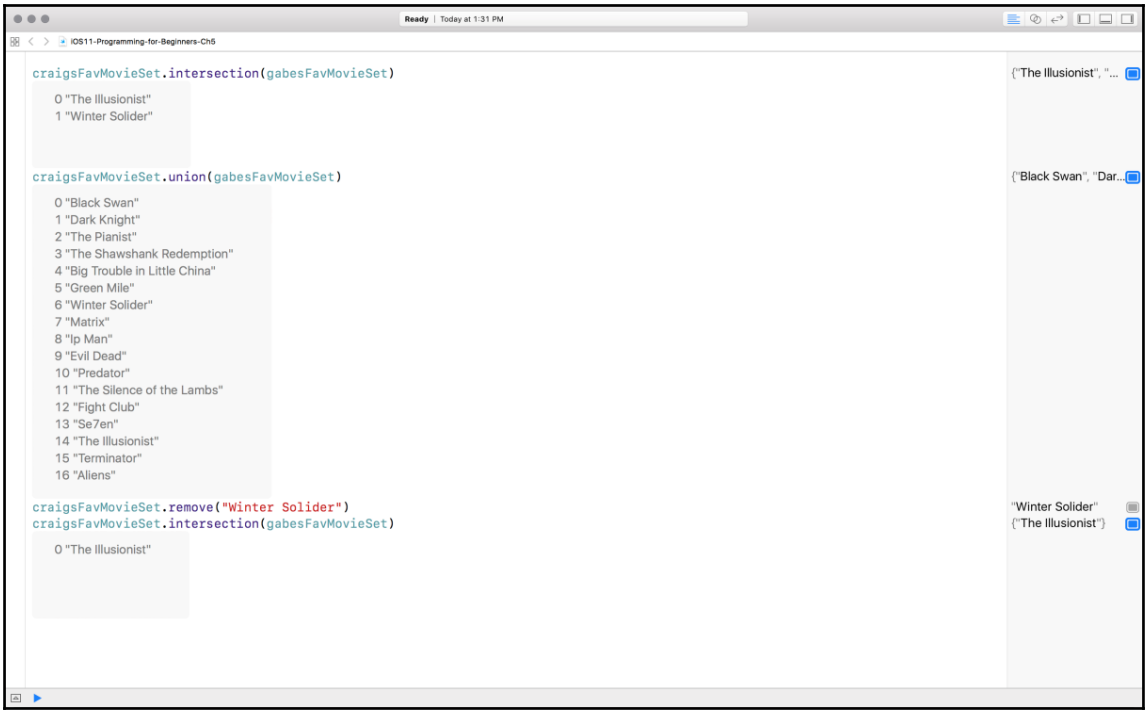

Ready | Today at 1:28 PM
iOS11-Programming-for-Beginners-Ch5
let movieSet = Set<String>()
let numberSet = Set<Int>([])
var craigsFavMovieSet = Set<String>([])
var gabesFavMovieSet = Set<String>(["Fight Club", "Matrix", "Evil Dead", "Big Trouble in Little China", "Aliens", "Winter Solider", "The Illusionist", "Predator"])
gabesFavMovieSet.insert("Terminator")
gabesFavMovieSet
craigsFavMovieSet = ["The Pianist", "The Shawshank Redemption", "Dark Knight", "Black Swan", "Ip Man", "The Illusionist", "The Silence of the Lambs", "Winter Solider", "Green Mile", "Se7en"]
if craigsFavMovieSet.contains("Green Mile") {
print("Green Mile found")
}
for movie in gabesFavMovieSet {
print("Gabe's movie - \(movie)")
}
for movie in craigsFavMovieSet.sorted() {
print("Craig's movie - \(movie)")
}
Set([])
Set([])
Set([])
{"The Illusionist", "P...
(inserted true, mem...
{"The Illusionist", "P...
{"Black Swan", "Dar...
"Green Mile found\n"
(9 times)
(10 times)
Craig's movie - Black Swan
Craig's movie - Dark Knight
Craig's movie - Green Mile
Craig's movie - Ip Man
Craig's movie - Se7en
Craig's movie - The Illusionist
Craig's movie - The Pianist
Craig's movie - The Shawshank Redemption
Craig's movie - The Silence of the Lambs
Craig's movie - Winter Solider

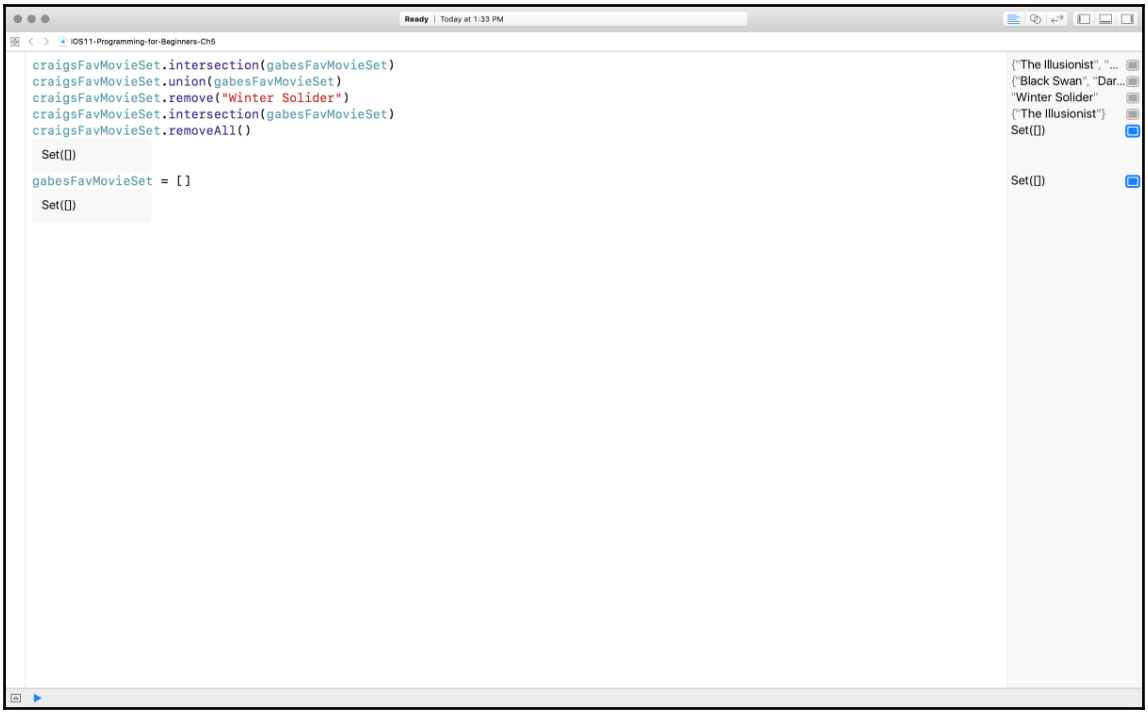

INTERSECTION
Favorite movies they have in common
Craig's Favorite Movies
Gabe's Favorite Movies


Ready | Today at 1:28 PM
iOS11-Programming-for-Beginners-Ch5
craigsFavMovieSet.intersection(gabesFavMovieSet)
0 "The Illusionist"
1 "Winter Solider"
{"The Illusionist", "...
Craig's movie - The Shawshank Redemption
Craig's movie - The Silence of the Lambs
Craig's movie - Winter Solider

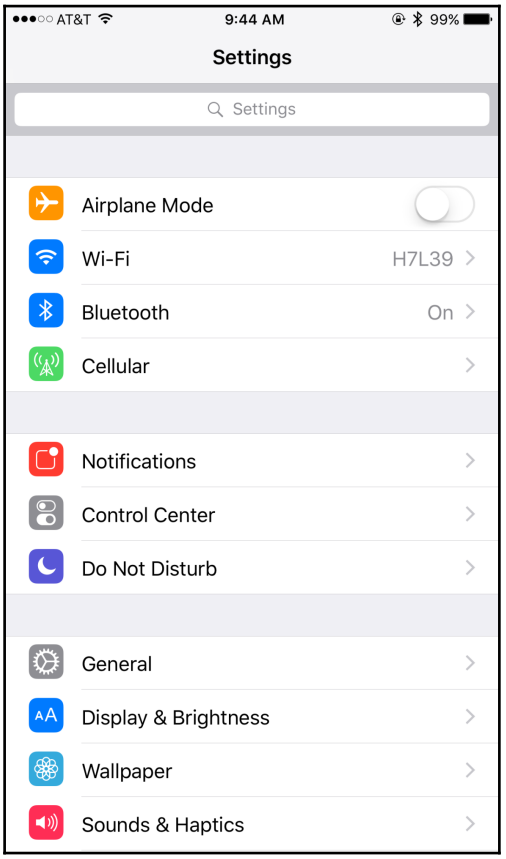

UNION
Favorite movies in one list
Craig's Favorite Movies
Gabe's Favorite Movies

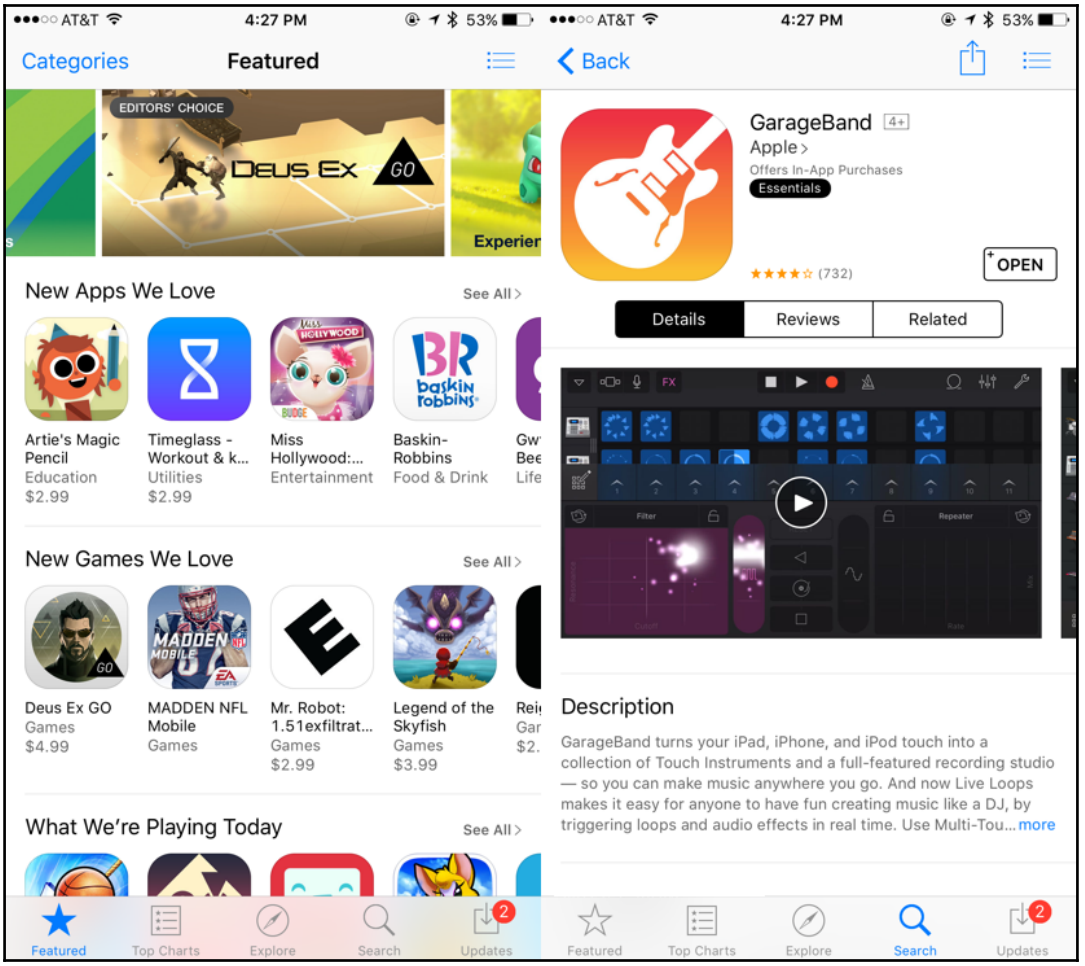

Ready | Today at 1:29 PM
iOS11-Programming-for-Beginners-Ch5
craigsFavMovieSet.intersection(gabesFavMovieSet)
{"The Illusionist", "...
0 "The Illusionist"
1 "Winter Solider"
craigsFavMovieSet.union(gabesFavMovieSet)
{"Black Swan", "Dar...
0 "Black Swan"
1 "Dark Knight"
2 "The Pianist"
3 "The Shawshank Redemption"
4 "Big Trouble in Little China"
5 "Green Mile"
6 "Winter Solider"
7 "Matrix"
8 "Ip Man"
9 "Evil Dead"
10 "Predator"
11 "The Silence of the Lambs"
12 "Fight Club"
13 "Se7en"
14 "The Illusionist"
15 "Terminator"
16 "Aliens"

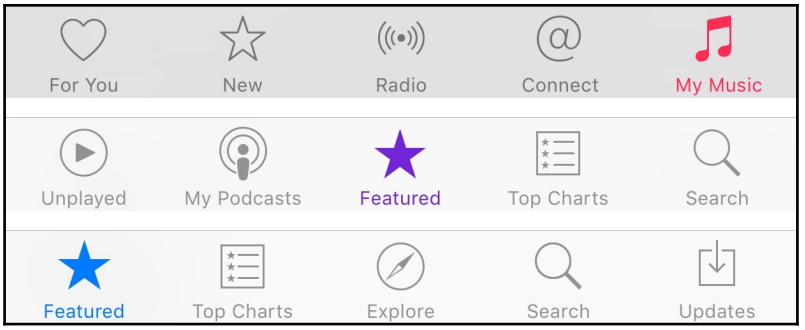

Ready | Today at 1:31 PM
iOS11-Programming-for-Beginners-Ch5
craigsFavMovieSet.intersection(gabesFavMovieSet)
{"The Illusionist", "...
0 "The Illusionist"
1 "Winter Solider"
craigsFavMovieSet.union(gabesFavMovieSet)
{"Black Swan", "Dar...
0 "Black Swan"
1 "Dark Knight"
2 "The Pianist"
3 "The Shawshank Redemption"
4 "Big Trouble in Little China"
5 "Green Mile"
6 "Winter Solider"
7 "Matrix"
8 "Ip Man"
9 "Evil Dead"
10 "Predator"
11 "The Silence of the Lambs"
12 "Fight Club"
13 "Se7en"
14 "The Illusionist"
15 "Terminator"
16 "Aliens"
craigsFavMovieSet.remove("Winter Solider")
"Winter Solider"
craigsFavMovieSet.intersection(gabesFavMovieSet)
{"The Illusionist"}
0 "The Illusionist"

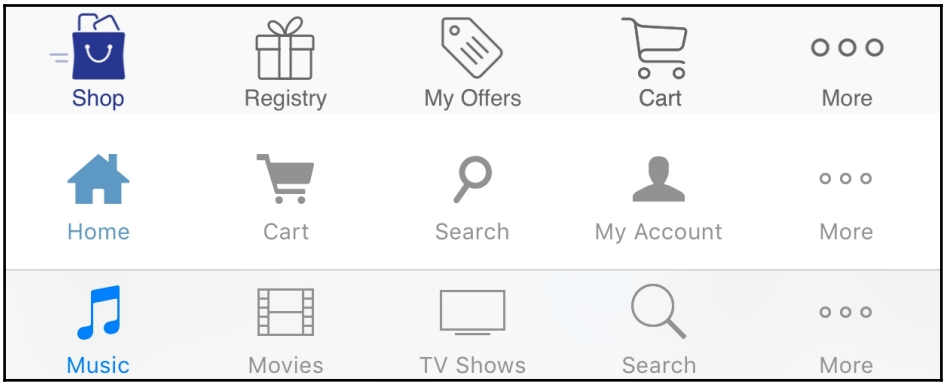

Ready | Today at 1:33 PM
iOS11-Programming-for-Beginners-Ch5
craigsFavMovieSet.intersection(gabesFavMovieSet)
craigsFavMovieSet.union(gabesFavMovieSet)
craigsFavMovieSet.remove("Winter Solider")
craigsFavMovieSet.intersection(gabesFavMovieSet)
craigsFavMovieSet.removeAll()
Set([])
gabesFavMovieSet = []
Set([])
{"The Illusionist", "...
{"Black Swan", "Dar...
"Winter Solider"
{"The Illusionist"}
Set([])
Set([])

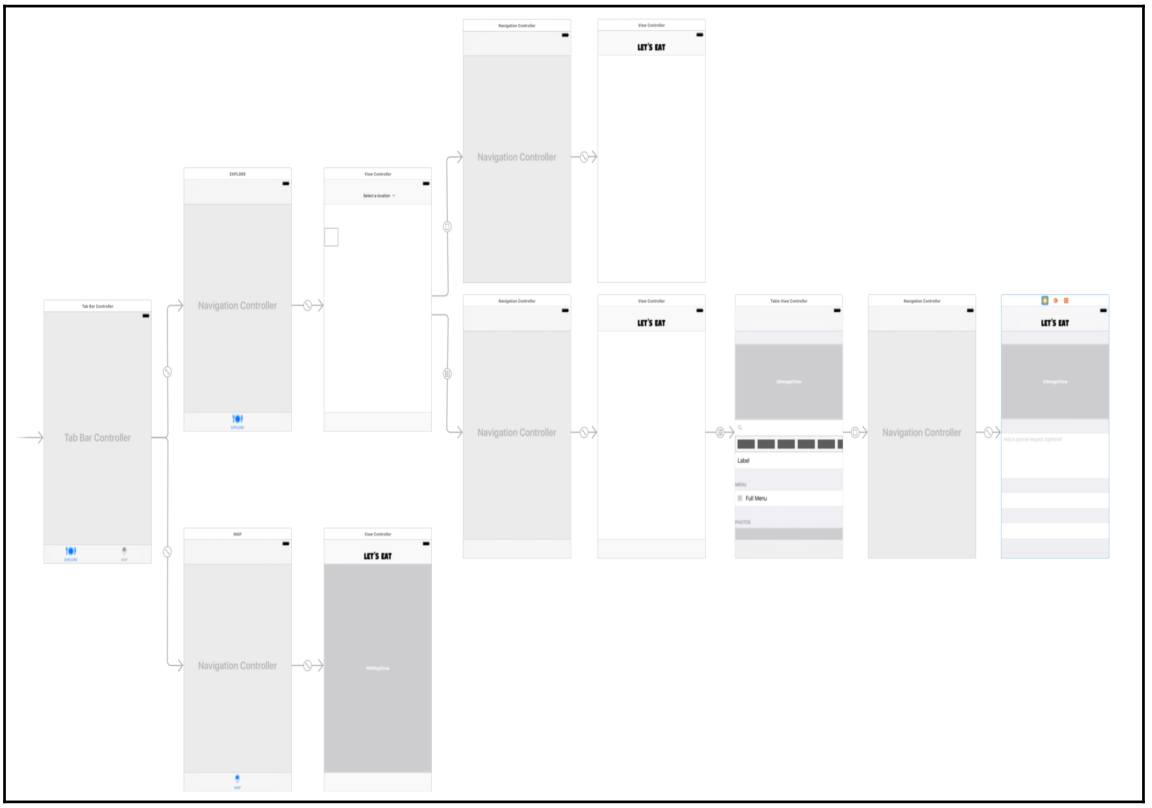
Chapter 6: Starting the UI Setup

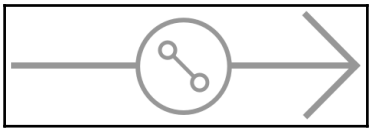


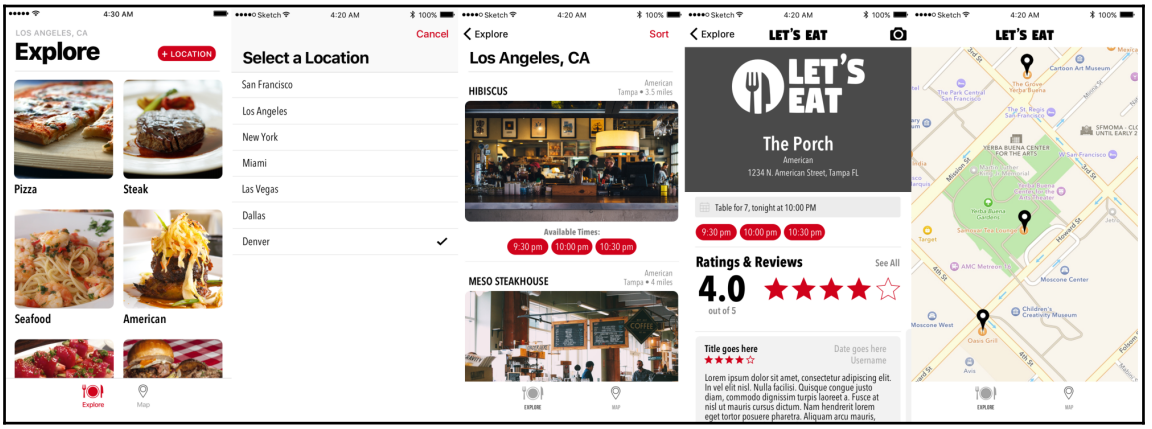

AT&T 4:27 PM 53%
Categories
Featured
EDITORS' CHOICE
Deus Ex GO
New Apps We Love
See All
Artie's Magic Pencil
Education
$2.99
Timeglass - Workout & k...
Utilities
$2.99
Miss Hollywood:...
Entertainment
Baskin-Robbins
Food & Drink
New Games We Love
See All
Deus Ex GO
Games
$4.99
MADDEN NFL Mobile
Games
Mr. Robot: 1.51exfiltrat...
Games
$2.99
Legend of the Skyfish
Games
$3.99
What We're Playing Today
See All
Featured
Top Charts
Explore
Search
Updates
Back
GarageBand
4+
Apple
Offers In-App Purchases
Essentials
(732)
OPEN
Details
Reviews
Related
Description
GarageBand turns your iPad, iPhone, and iPod touch into a collection of Touch Instruments and a full-featured recording studio — so you can make music anywhere you go. And now Live Loops makes it easy for anyone to have fun creating music like a DJ, by triggering loops and audio effects in real time. Use Multi-Tou... more


For You
New
Radio
Connect
My Music
Unplayed
My Podcasts
Featured
Top Charts
Search
Featured
Top Charts
Explore
Search
Updates

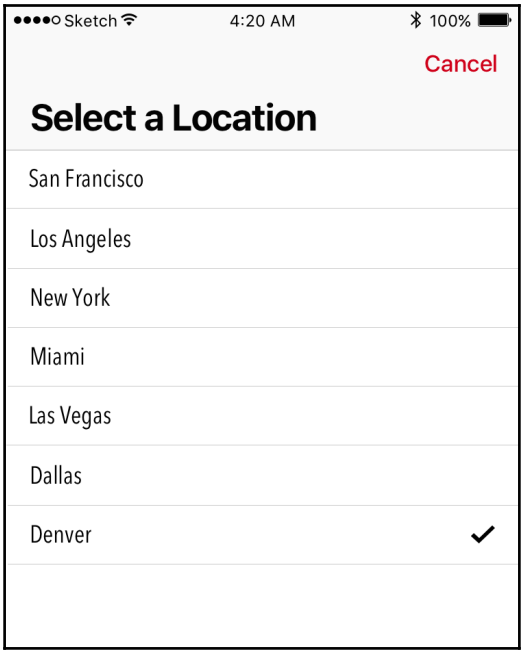

Shop
Registry
My Offers
Cart
More
Home
Cart
Search
My Account
More
Music
Movies
TV Shows
Search
More

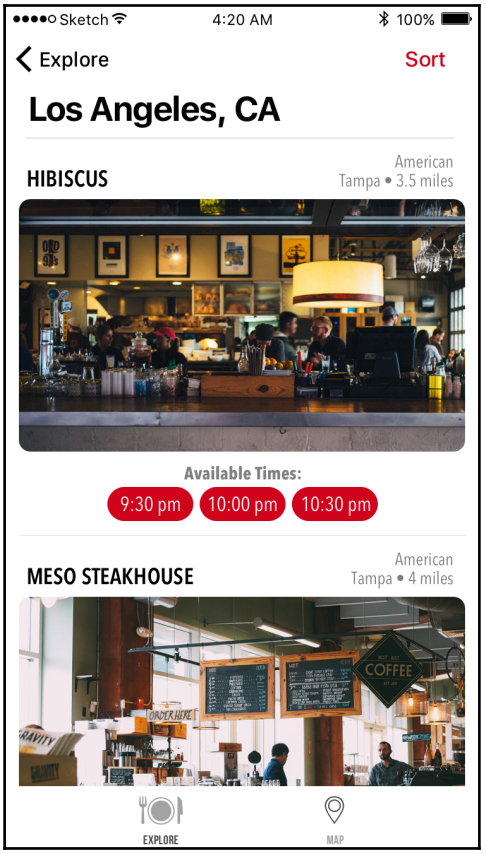

Tab Bar Controller
Navigation Controller
Navigation Controller
Navigation Controller
Navigation Controller
Navigation Controller
LET'S EAT
LET'S EAT
LET'S EAT
LET'S EAT
Label
Full Menu

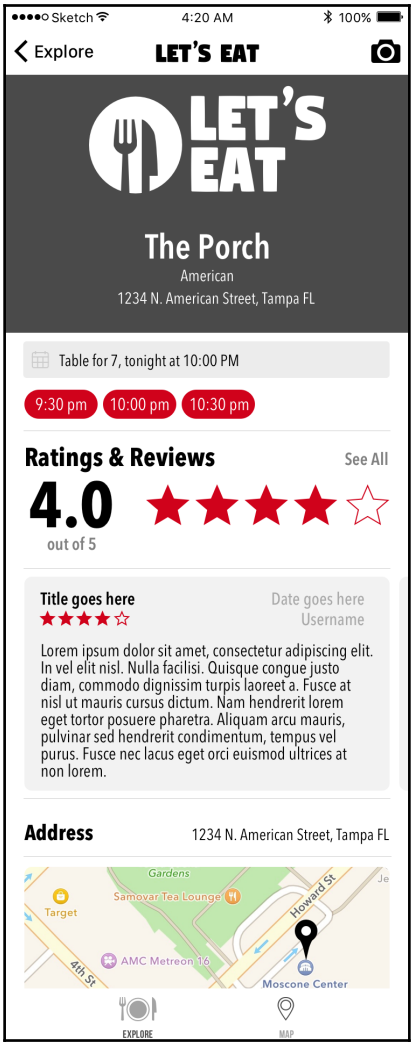


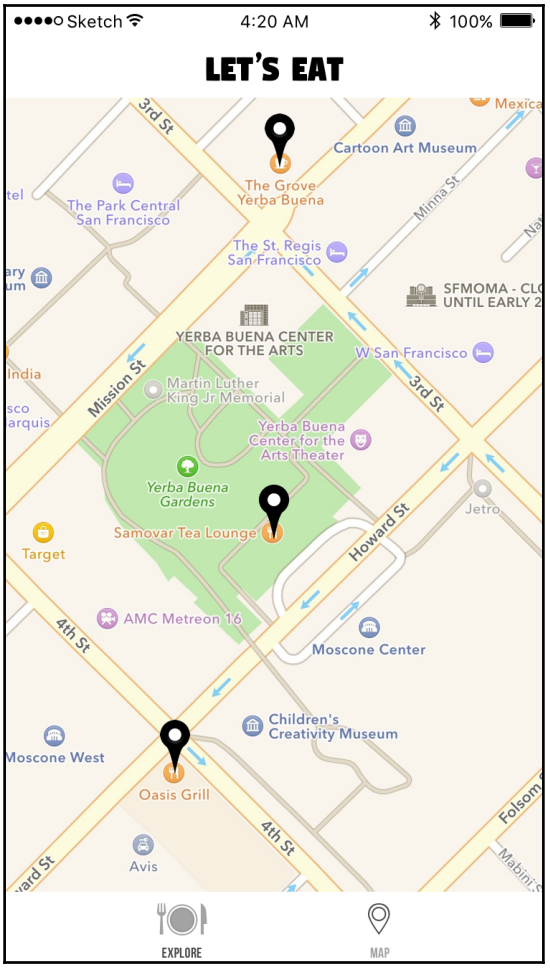

LOS ANGELES, CA
Explore
+ LOCATION
Pizza
Steak
Seafood
American
Explore
Map
Cancel
Select a Location
San Francisco
Los Angeles
New York
Miami
Las Vegas
Dallas
Denver
Explore
Sort
Los Angeles, CA
HIBISCUS
American
Tampa • 3.5 miles
Available Times:
9:30 pm
10:00 pm
10:30 pm
MESO STEAKHOUSE
American
Tampa • 4 miles
Explore
LET'S EAT
LET'S EAT
The Porch
American
1234 N. American Street, Tampa FL
Table for 7, tonight at 10:00 PM
9:30 pm
10:00 pm
10:30 pm
Ratings & Reviews
See All
4.0
out of 5
Title goes here
Date goes here
Username
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vel elit nisl. Nulla facilisi. Quisque congue justo diam, commodo dignissim turpis laoreet a. Fusce at nisl ut mauris cursus dictum. Nam hendrerit lorem eget tortor posuere pharetra. Aliquam arcu mauris,
LET'S EAT
EXPLORE
MAP

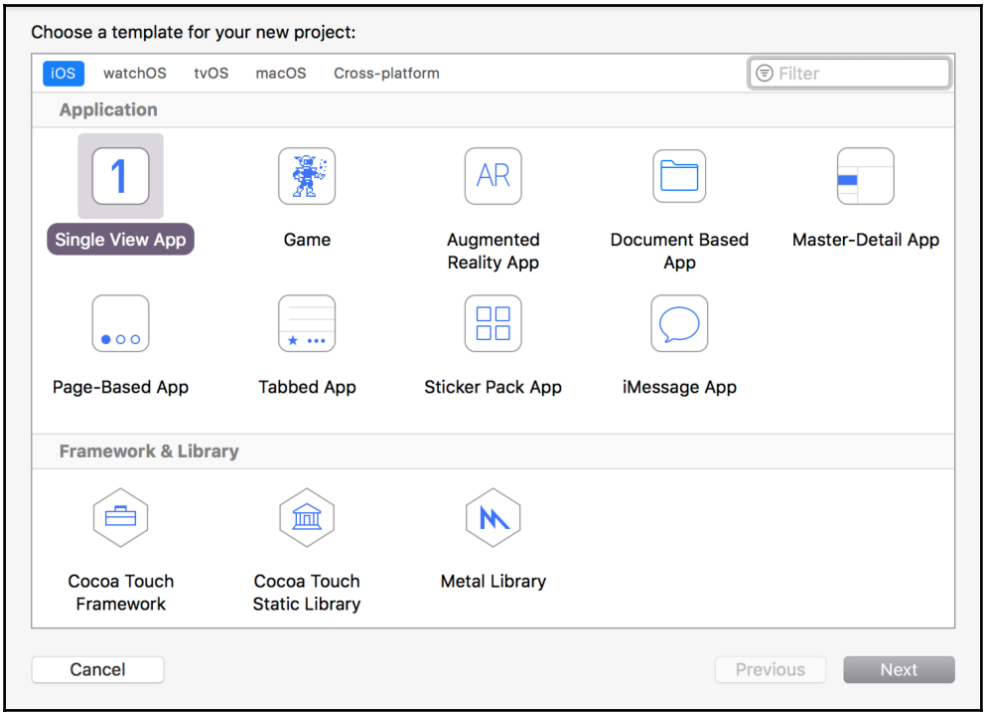

4:30 AM
LOS ANGELES, CA
Explore
+ LOCATION
Pizza
Steak
Seafood
American
Explore
Map

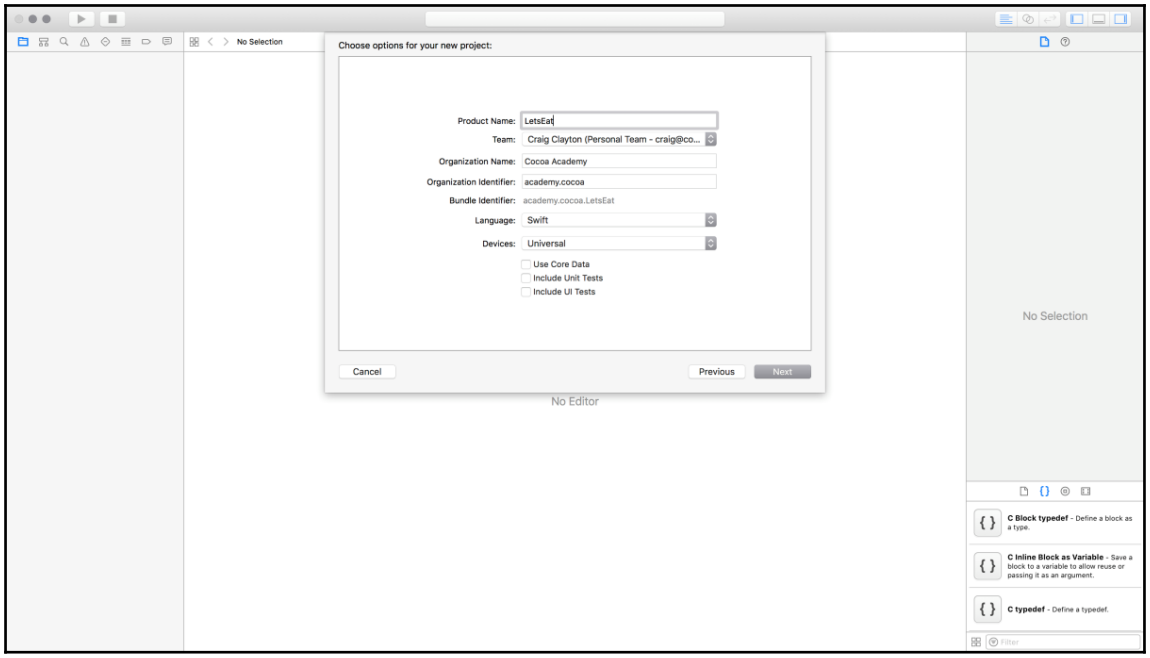

Sketch
4:20 AM
100%
Cancel
Select a Location
San Francisco
Los Angeles
New York
Miami
Las Vegas
Dallas
Denver

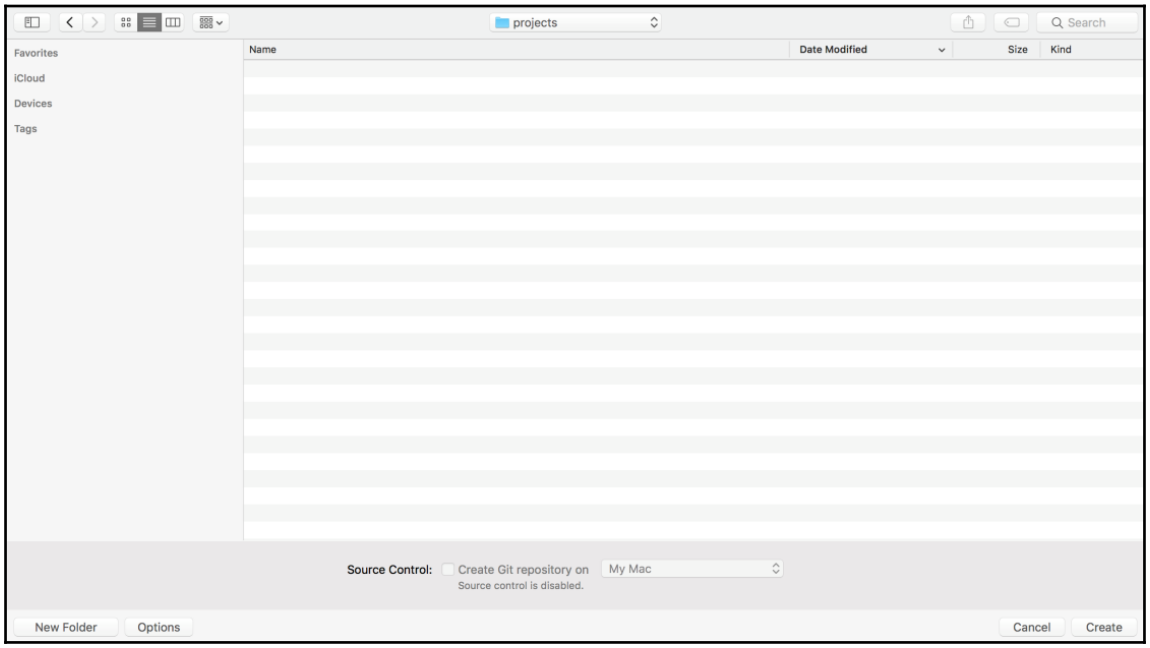

Sketch
4:20 AM
100%
Explore
Sort
Los Angeles, CA
HIBISCUS
American
Tampa • 3.5 miles
Available Times:
9:30 pm
10:00 pm
10:30 pm
MESO STEAKHOUSE
American
Tampa • 4 miles
COFFEE
ORDER HERE
EXPLORE
MAP

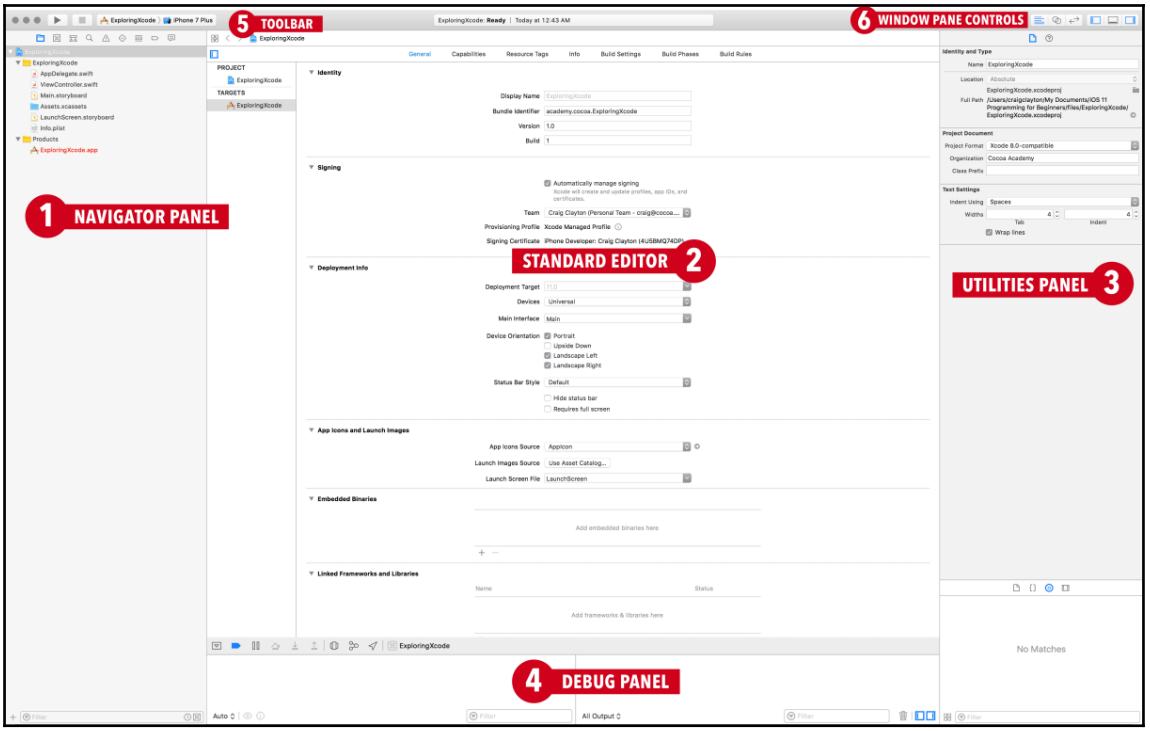

Sketch
4:20 AM
100%
Explore
LET'S EAT
LET'S EAT
The Porch
American
1234 N. American Street, Tampa FL
Table for 7, tonight at 10:00 PM
9:30 pm
10:00 pm
10:30 pm
Ratings & Reviews
See All
4.0
out of 5
Title goes here
Date goes here
Username
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vel elit nisl. Nulla facilisi. Quisque congue justo diam, commodo dignissim turpis laoreet a. Fusce at nisl ut mauris cursus dictum. Nam hendrerit lorem eget tortor posuere pharetra. Aliquam arcu mauris, pulvinar sed hendrerit condimentum, tempus vel purus. Fusce nec lacus eget orci euismod ultrices at non lorem.
Address
1234 N. American Street, Tampa FL
Gardens
Samovar Tea Lounge
Target
Howard St
AMC Metreon 16
4th St
Moscone Center
EXPLORE
MAP


Sketch
4:20 AM
100%
LET'S EAT
3rd St
Cartoon Art Museum
The Grove
Yerba Buena
The Park Central
San Francisco
Minna St
The St. Regis
San Francisco
SFMOMA - CLO
UNTIL EARLY 2
YERBA BUENA CENTER
FOR THE ARTS
W San Francisco
Mission St
Martin Luther
King Jr Memorial
3rd St
Yerba Buena
Center for the
Arts Theater
Yerba Buena
Gardens
Jetro
Samovar Tea Lounge
Target
Howard St
AMC Metreon 16
4th St
Moscone Center
Children's
Creativity Museum
Moscone West
Oasis Grill
4th St
Folsom
Avis
EXPLORE
MAP

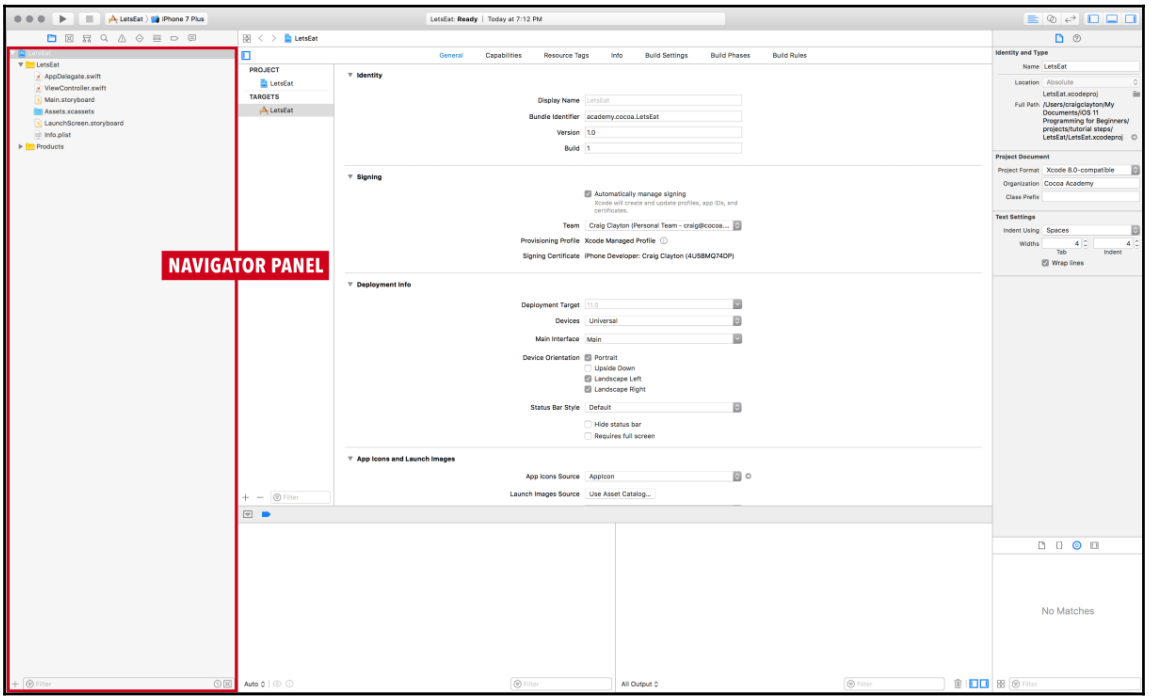

Choose a template for your new project:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application
Single View App
Game
Augmented Reality App
Document Based App
Master-Detail App
Page-Based App
Tabbed App
Sticker Pack App
iMessage App
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Metal Library
Cancel
Previous
Next

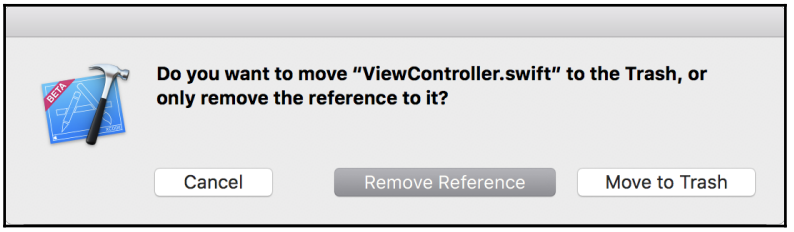

No Selection
Choose options for your new project:
Product Name: LetsEat
Team: Craig Clayton (Personal Team - craig@co...
Organization Name: Cocoa Academy
Organization Identifier: academy.cocoa
Bundle Identifier: academy.cocoa.LetsEat
Language: Swift
Devices: Universal
Use Core Data
Include Unit Tests
Include UI Tests
Cancel
Previous
Next
No Editor
No Selection
C Block typedef - Define a block as a type.
C Inline Block as Variable - Save a block to a variable to allow reuse or passing it as an argument.
C typedef - Define a typedef.

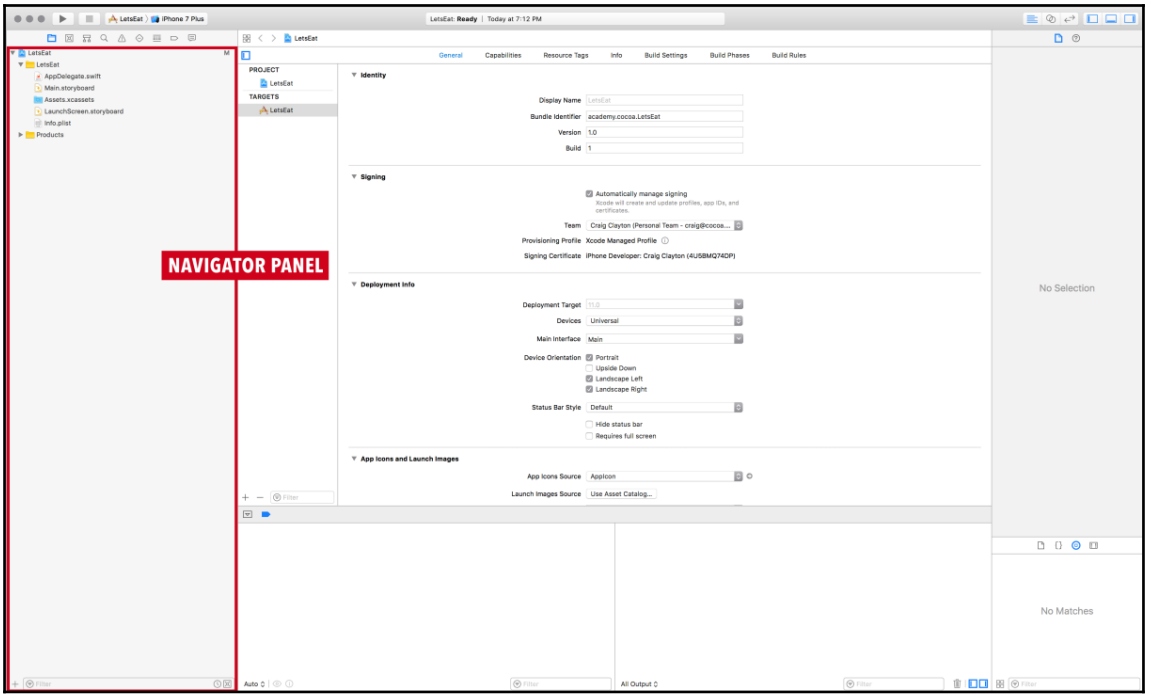

projects
Search
Favorites
iCloud
Devices
Tags
Name
Date Modified
Size
Kind
Source Control:
Create Git repository on
My Mac
Source control is disabled.
New Folder
Options
Cancel
Create

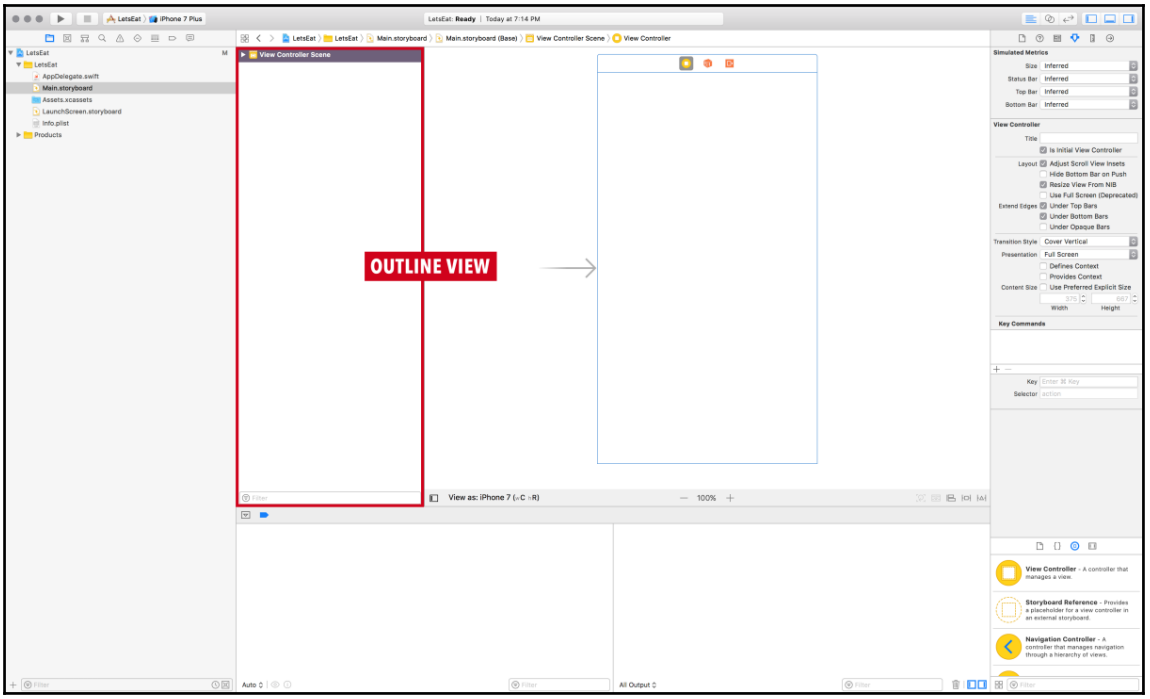

5 TOOLBAR
6 WINDOW PANE CONTROLS
1 NAVIGATOR PANEL
STANDARD EDITOR 2
UTILITIES PANEL 3
4 DEBUG PANEL

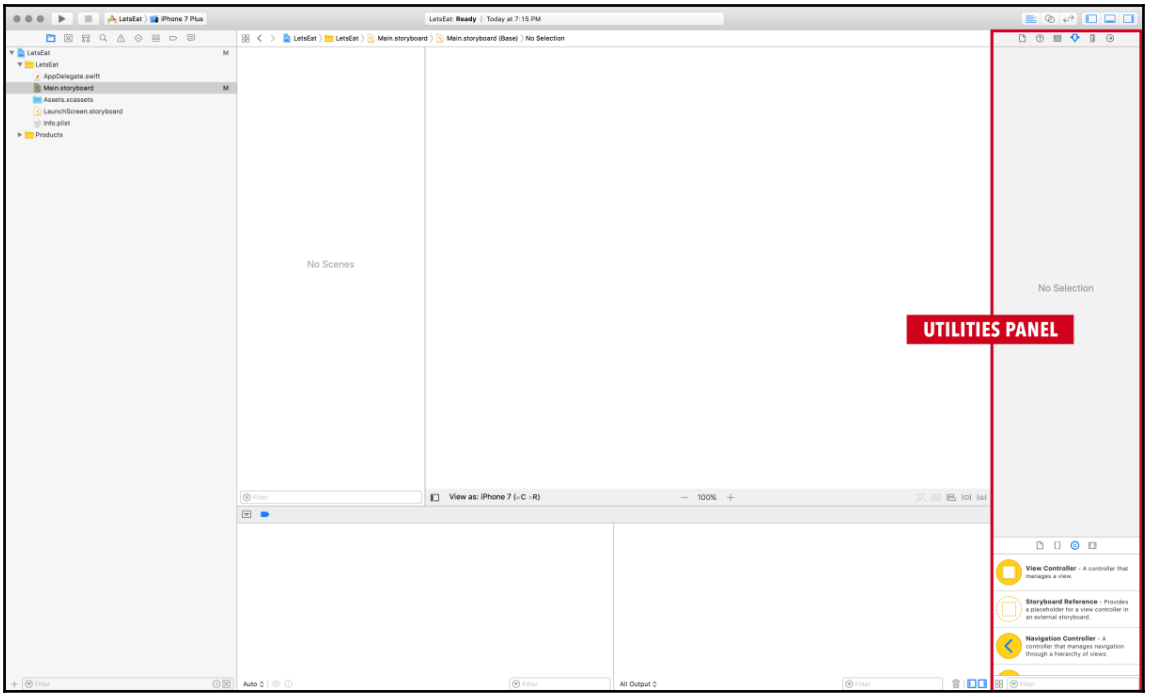
Chapter 7: Setting Up the Basic Structure

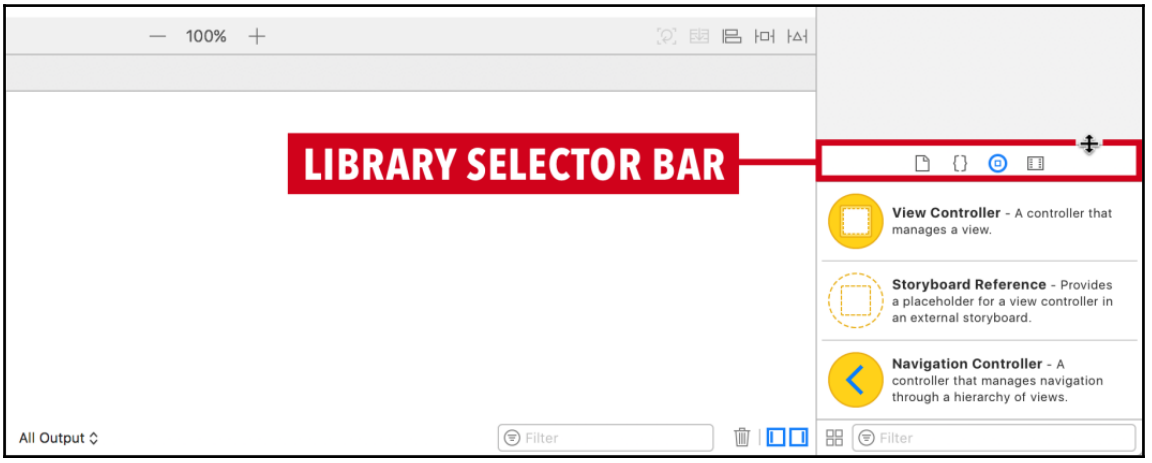


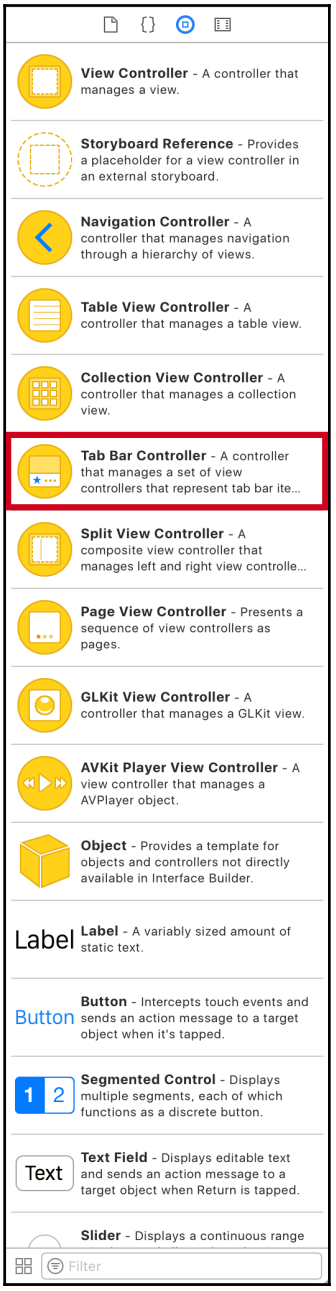


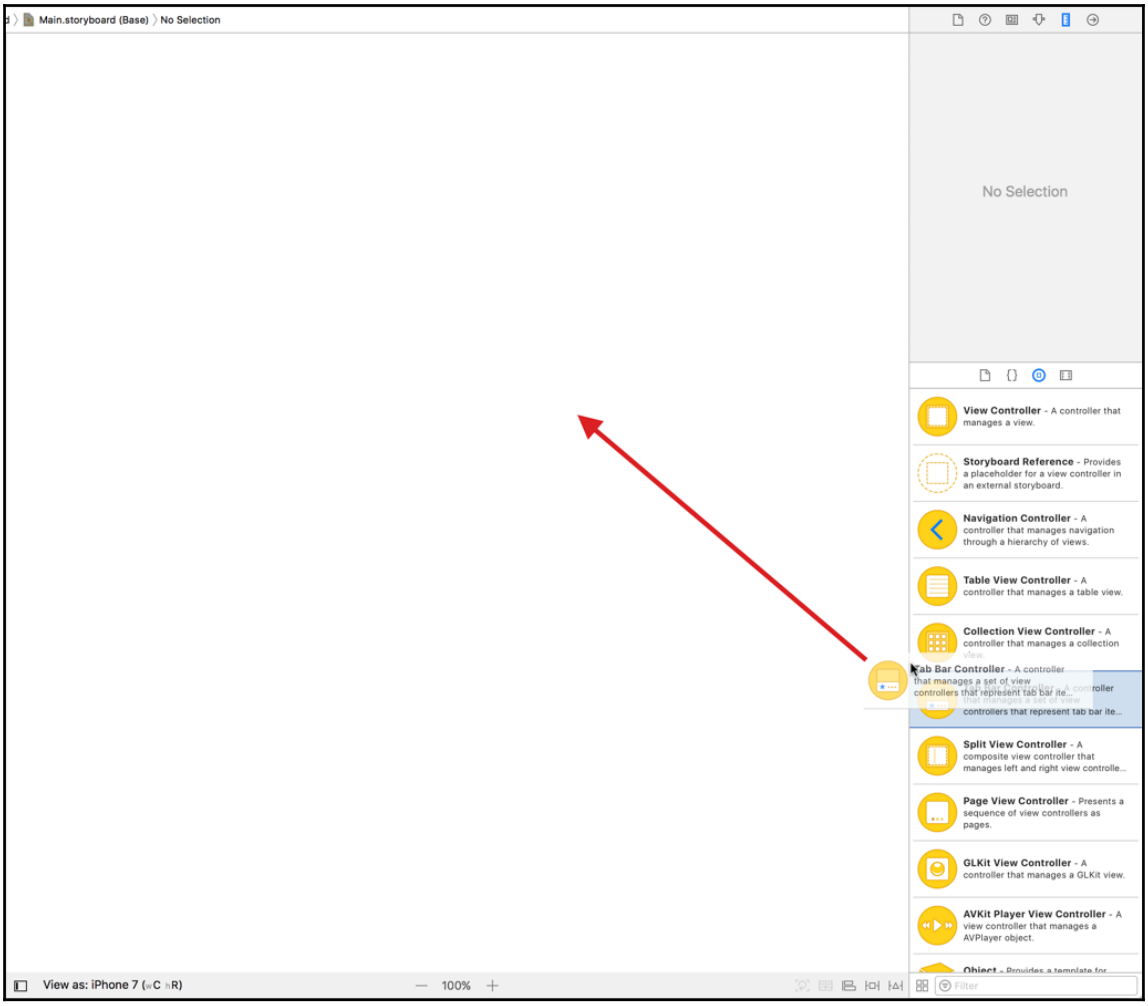

NAVIGATOR PANEL
No Selection
No Matches

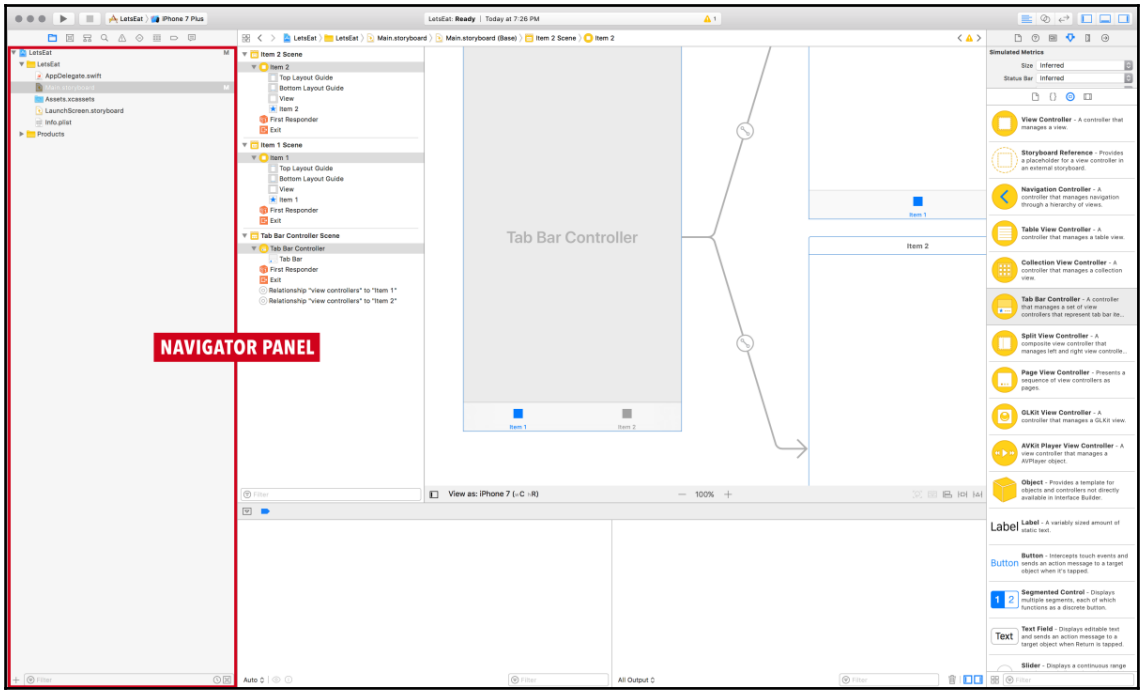

OUTLINE VIEW

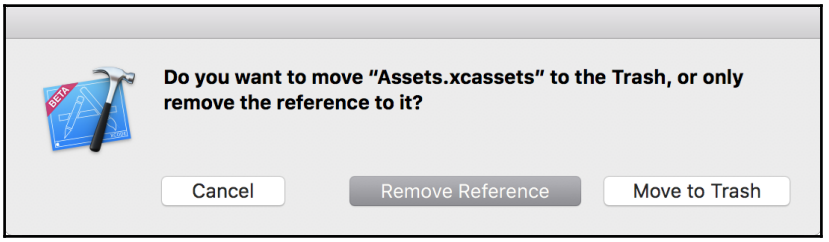

No Scenes
No Selection
UTILITIES PANEL

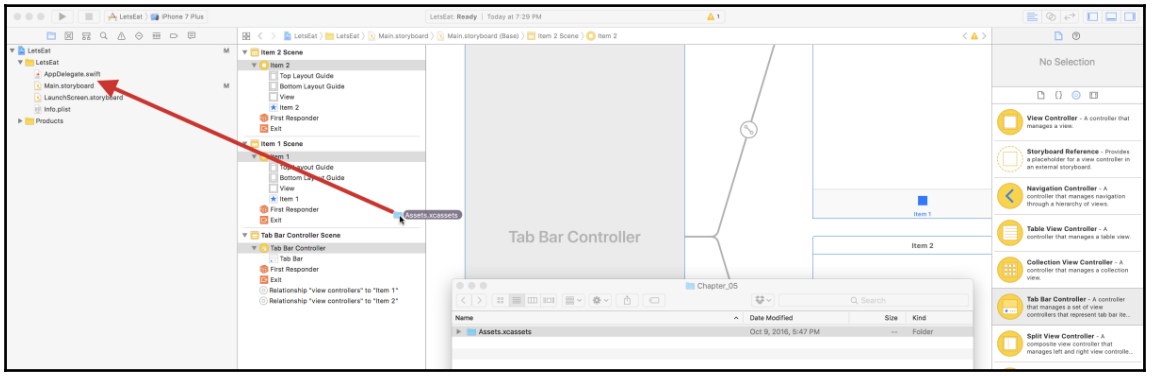

100%
LIBRARY SELECTOR BAR
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
All Output
Filter
Filter

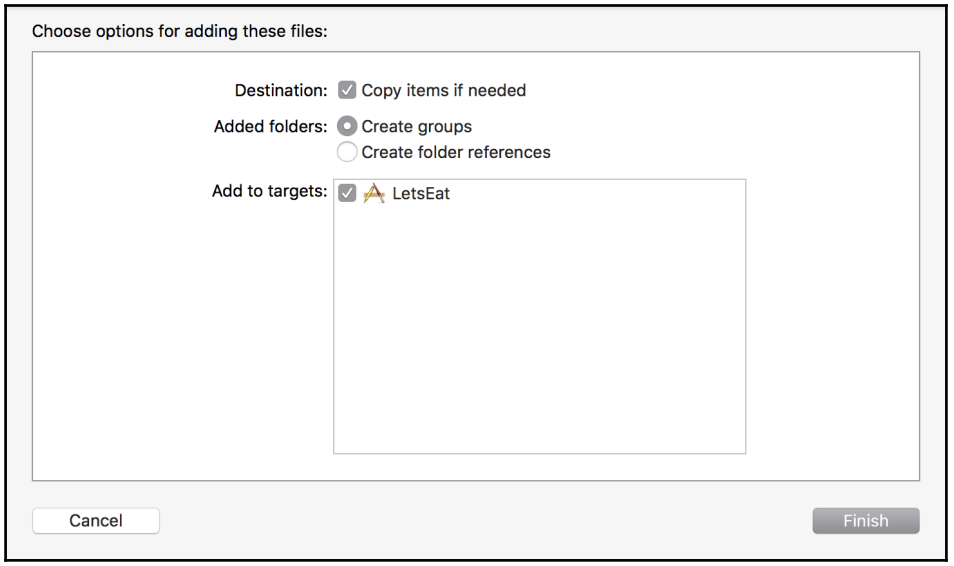

View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar ite...
Split View Controller - A composite view controller that manages left and right view controlle...
Page View Controller - Presents a sequence of view controllers as pages.
GLKit View Controller - A controller that manages a GLKit view.
AVKit Player View Controller - A view controller that manages a AVPlayer object.
Object - Provides a template for objects and controllers not directly available in Interface Builder.
Label
Label - A variably sized amount of static text.
Button
Button - Intercepts touch events and sends an action message to a target object when it's tapped.
1
2
Segmented Control - Displays multiple segments, each of which functions as a discrete button.
Text
Text Field - Displays editable text and sends an action message to a target object when Return is tapped.
Slider - Displays a continuous range
Filter

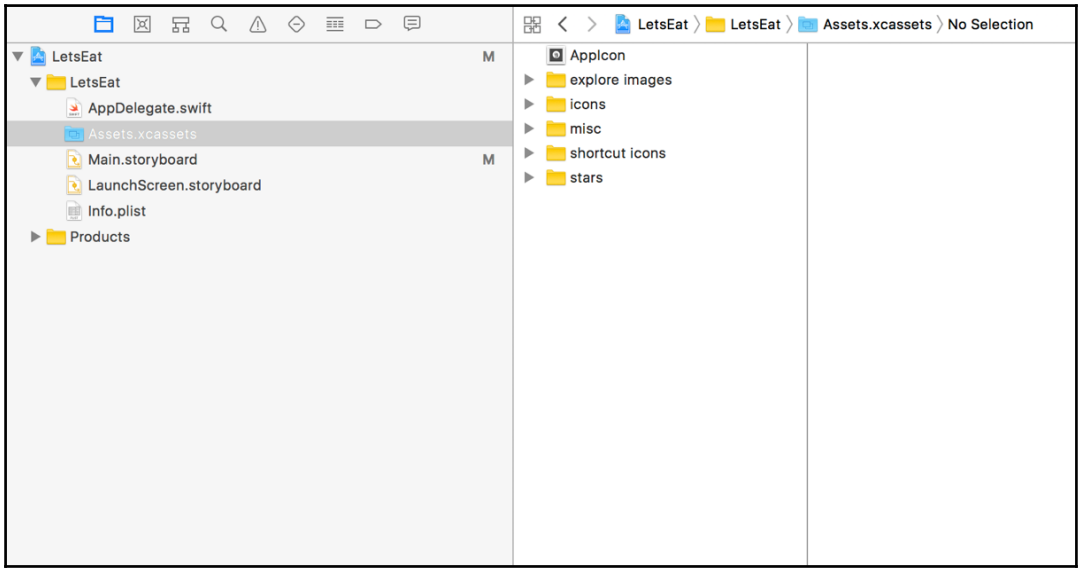

Main.storyboard (Base)
No Selection
No Selection
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar ite...
Split View Controller - A composite view controller that manages left and right view controlle...
Page View Controller - Presents a sequence of view controllers as pages.
GLKit View Controller - A controller that manages a GLKit view.
AVKit Player View Controller - A view controller that manages a AVPlayer object.
View as: iPhone 7
100%
Filter

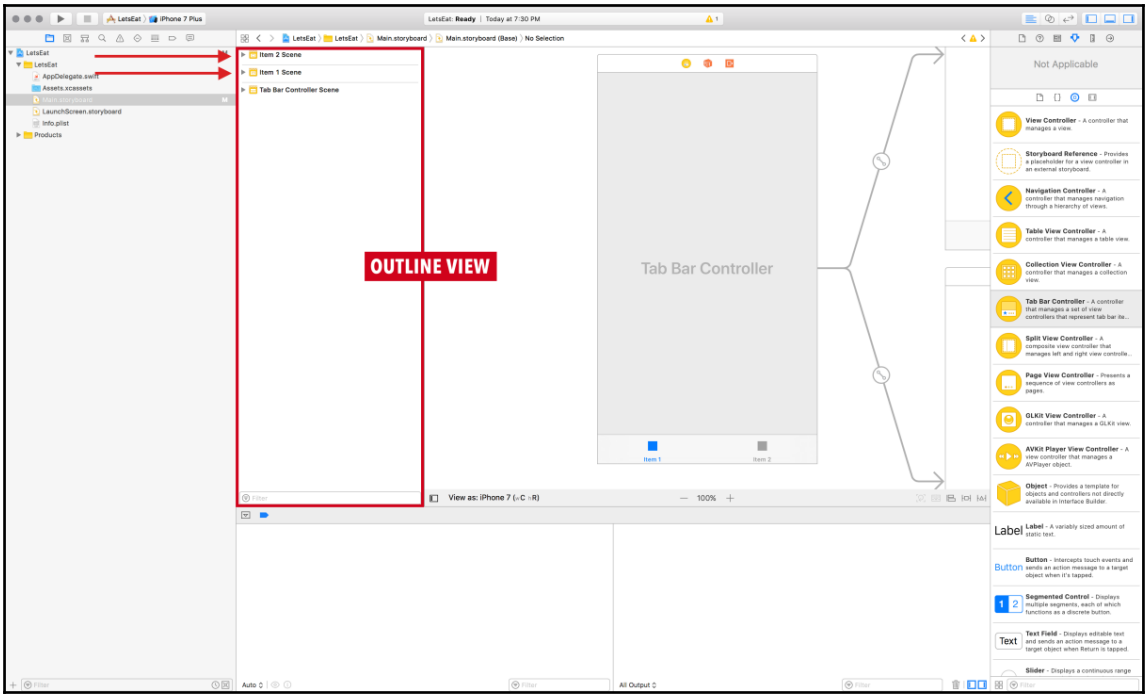

NAVIGATOR PANEL
Tab Bar Controller
Item 1
Item 2

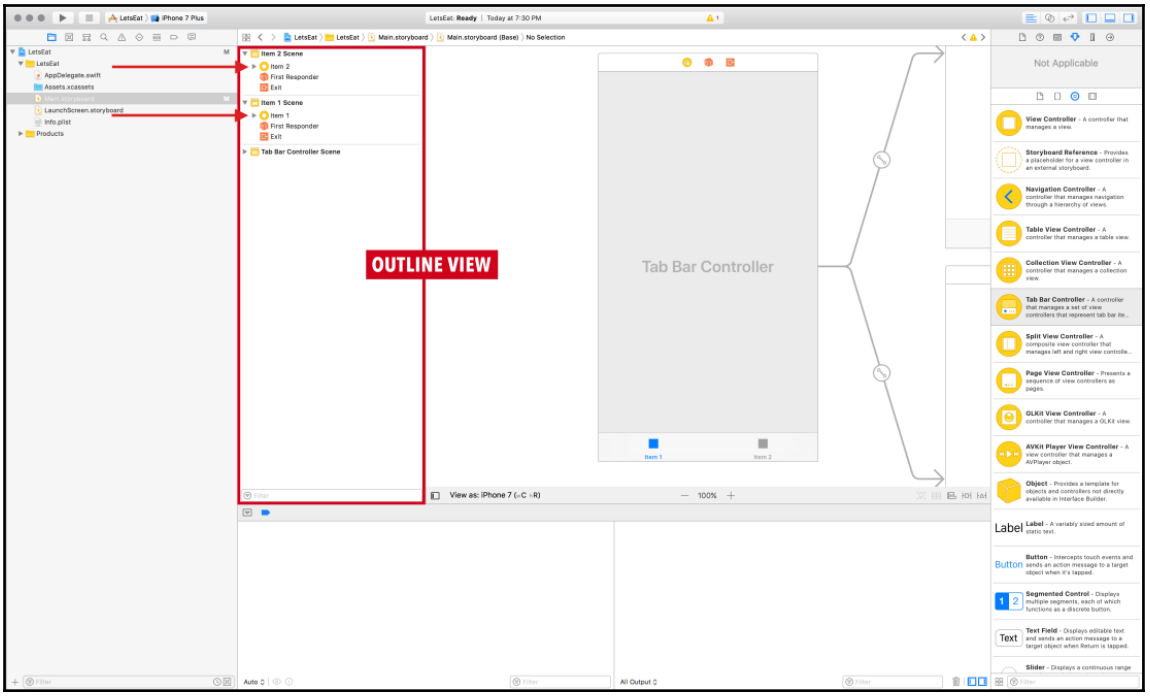

Do you want to move "Assets.xcassets" to the Trash, or only remove the reference to it?
Cancel
Remove Reference
Move to Trash

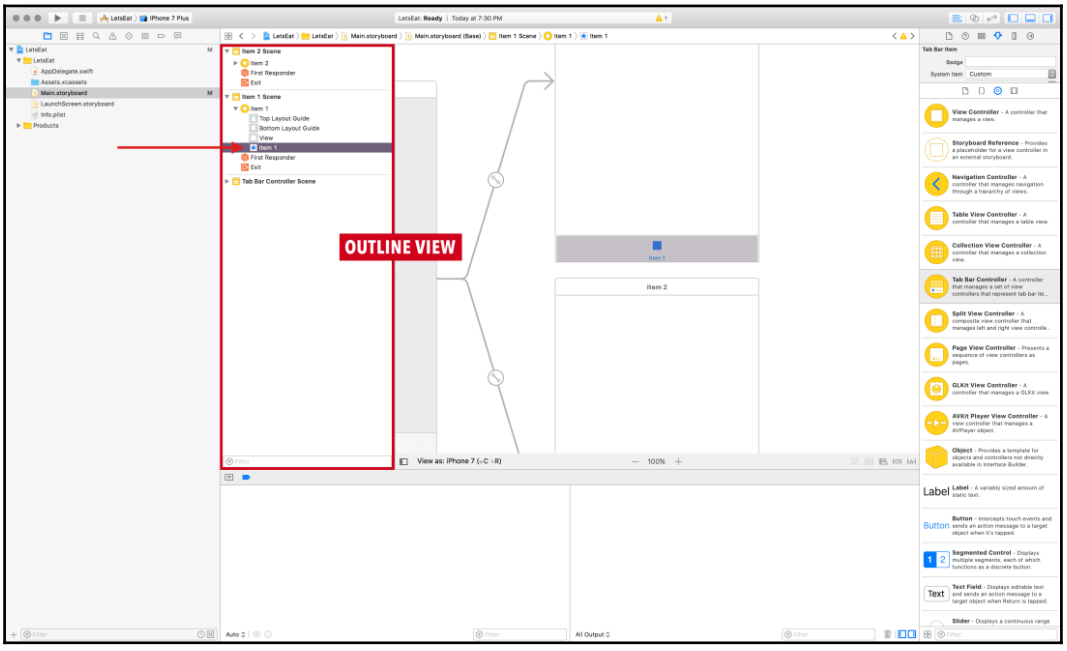

Tab Bar Controller
Item 1
Item 2
No Selection

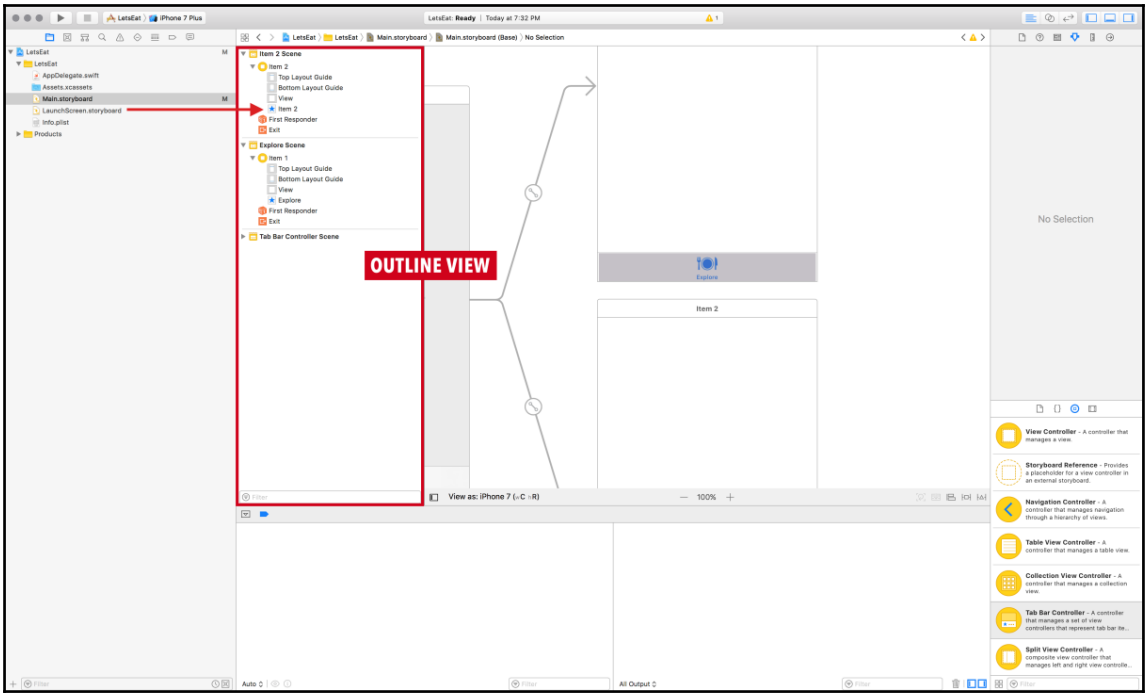

Choose options for adding these files:
Destination:
Copy items if needed
Added folders:
Create groups
Create folder references
Add to targets:
LetsEat
Cancel
Finish

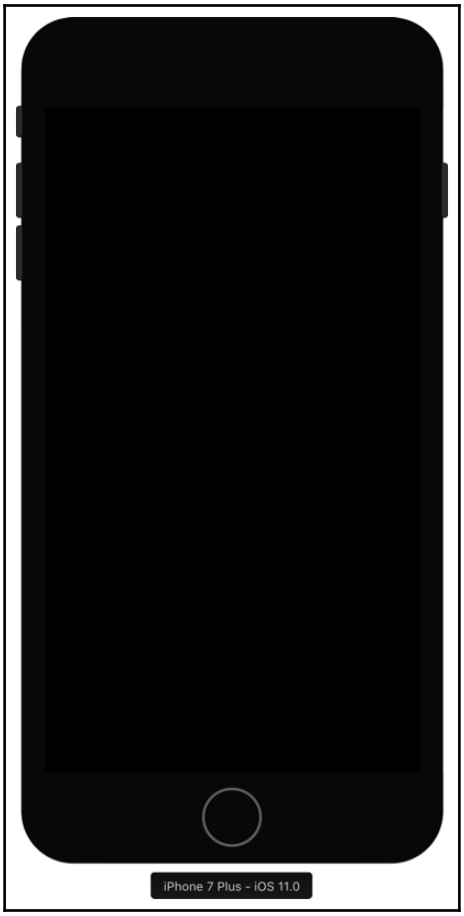

LetsEat
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
LaunchScreen.storyboard
Info.plist
Products
M
M
LetsEat
LetsEat
Assets.xcassets
No Selection
AppIcon
explore images
icons
misc
shortcut icons
stars

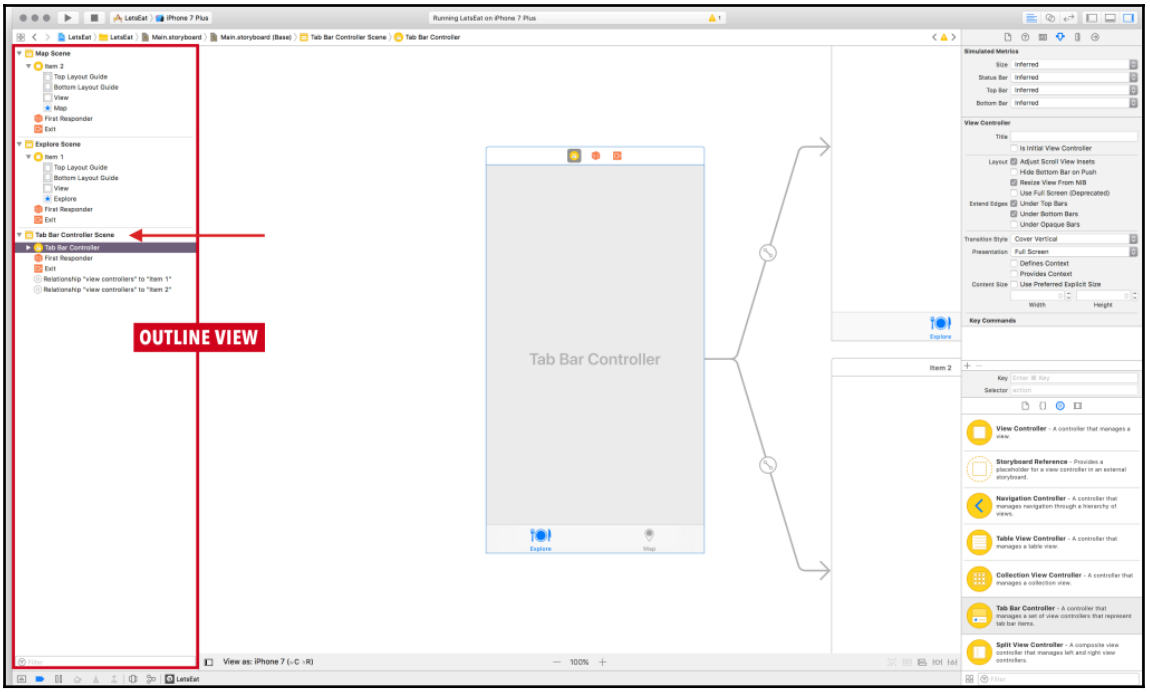

OUTLINE VIEW
Item 2 Scene
Item 1 Scene
Tab Bar Controller Scene
Tab Bar Controller
Not Applicable

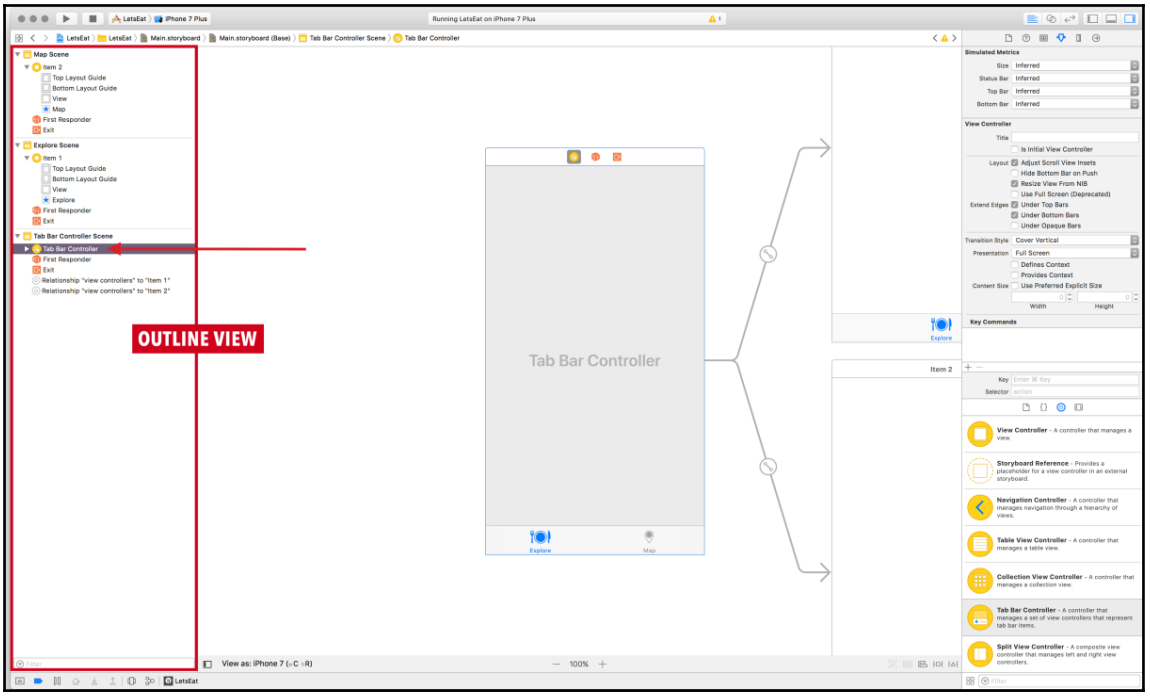

OUTLINE VIEW
Tab Bar Controller

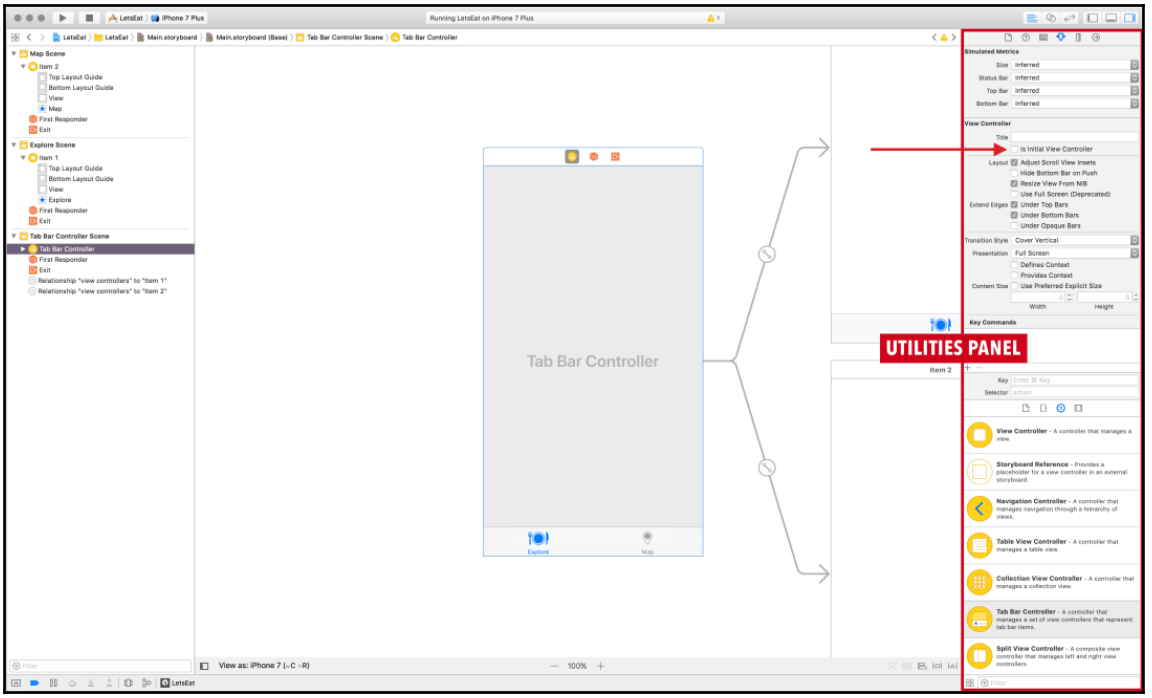

OUTLINE VIEW

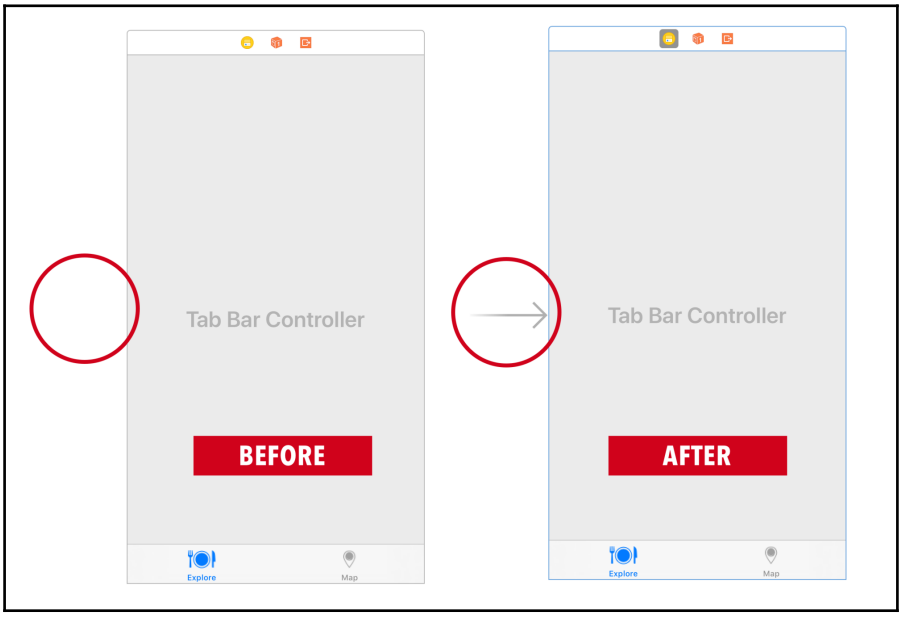

OUTLINE VIEW
No Selection

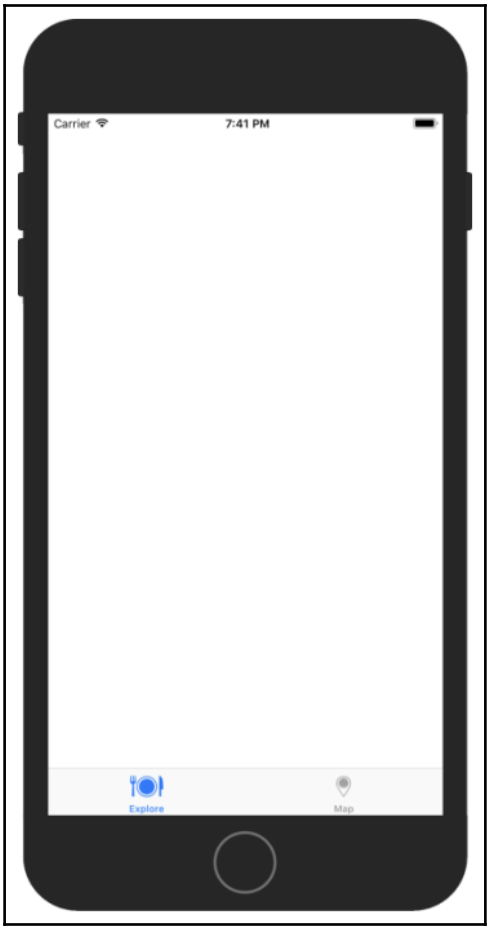

iPhone 7 Plus - iOS 11.0

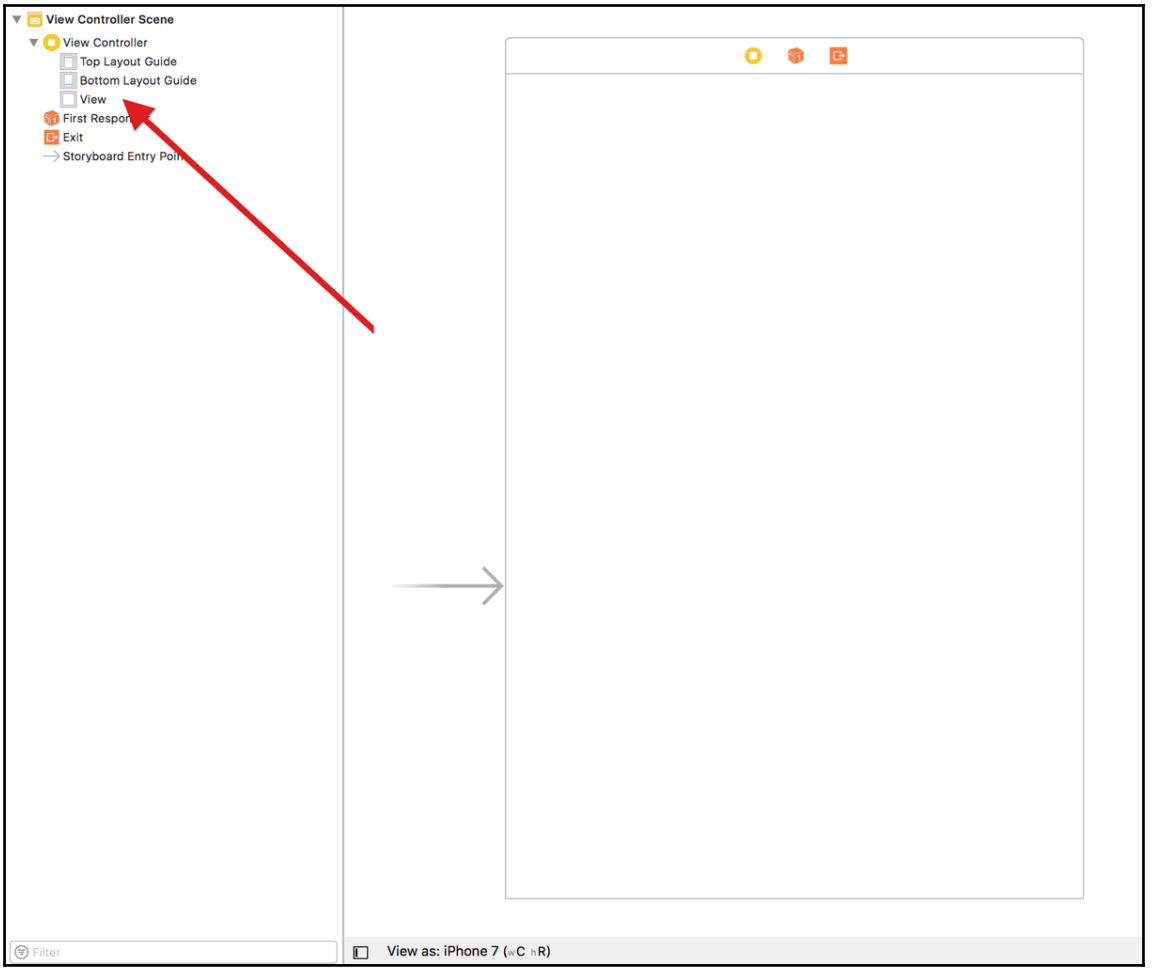

OUTLINE VIEW
Tab Bar Controller

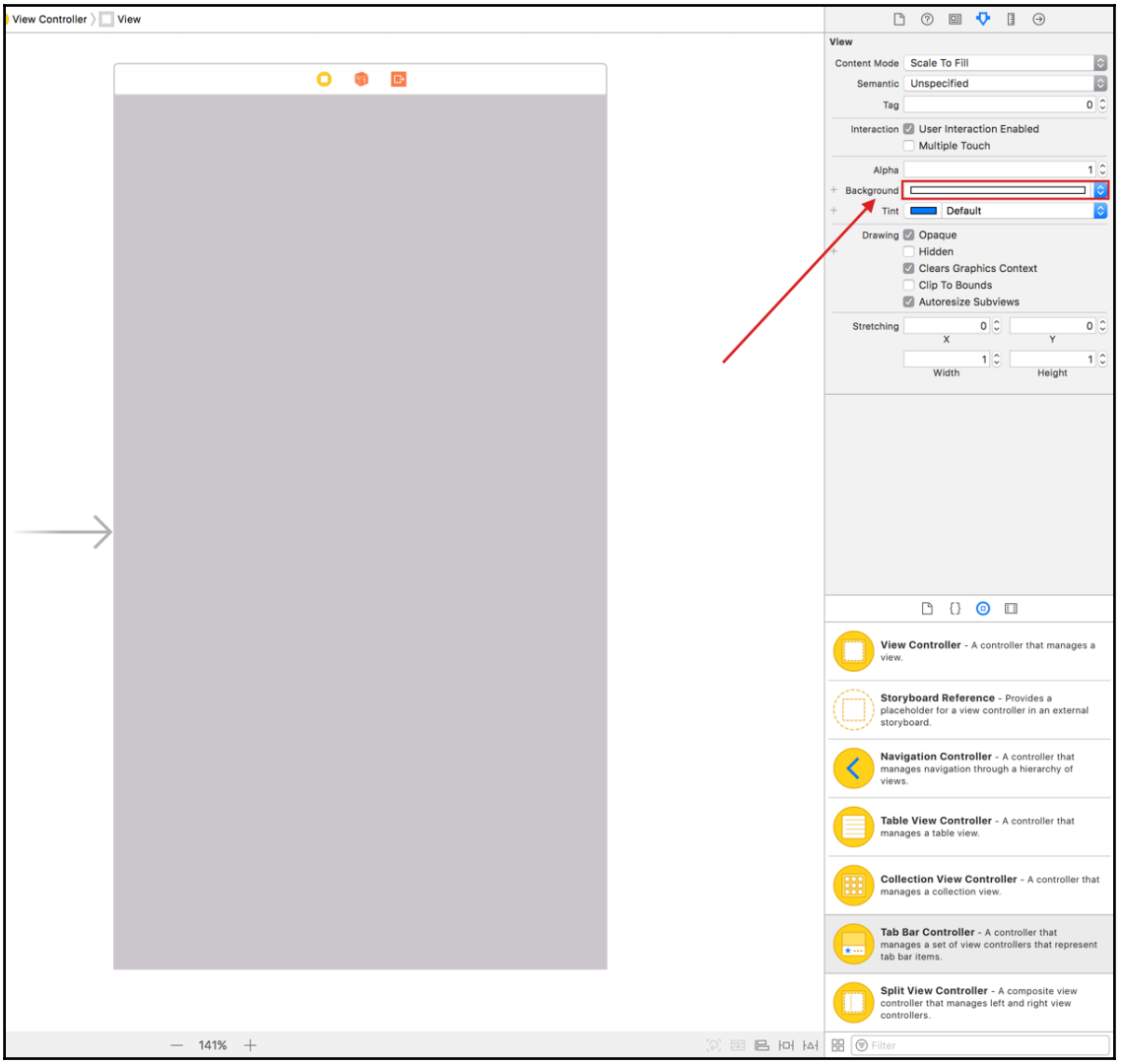

OUTLINE VIEW
Tab Bar Controller


Tab Bar Controller
UTILITIES PANEL
Explore
Map

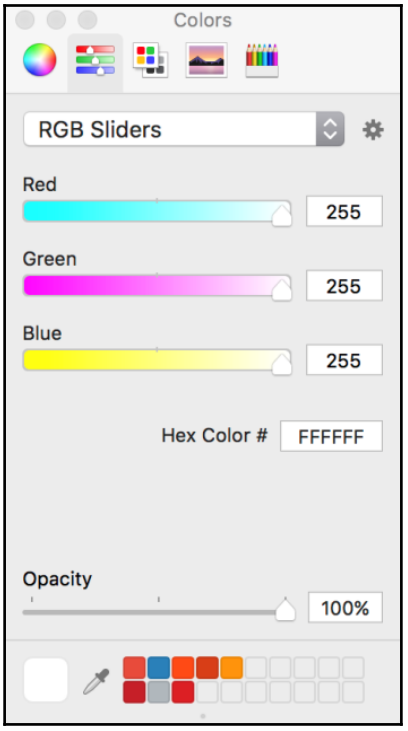

Tab Bar Controller
BEFORE
Explore
Map
Tab Bar Controller
AFTER
Explore
Map

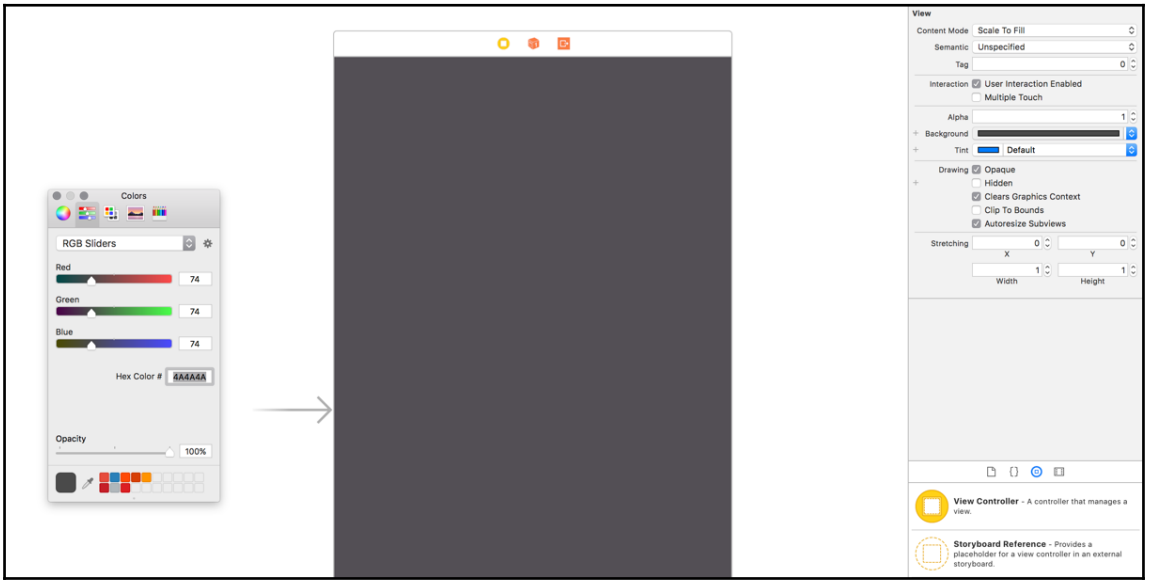

Carrier
7:41 PM
Explore
Map

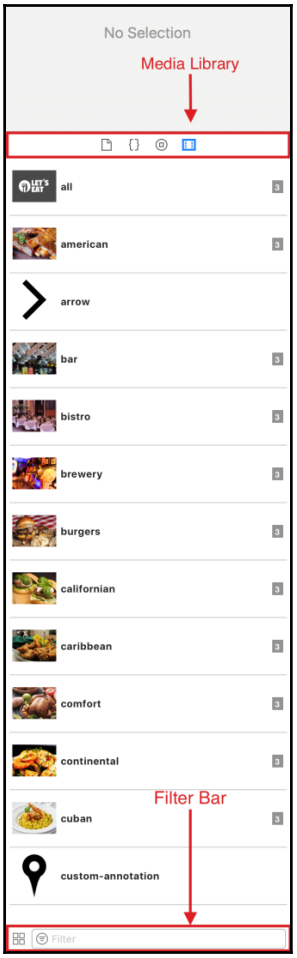

View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
First Respon
Exit
Storyboard Entry Poin
Filter
View as: iPhone 7 (wC hR)

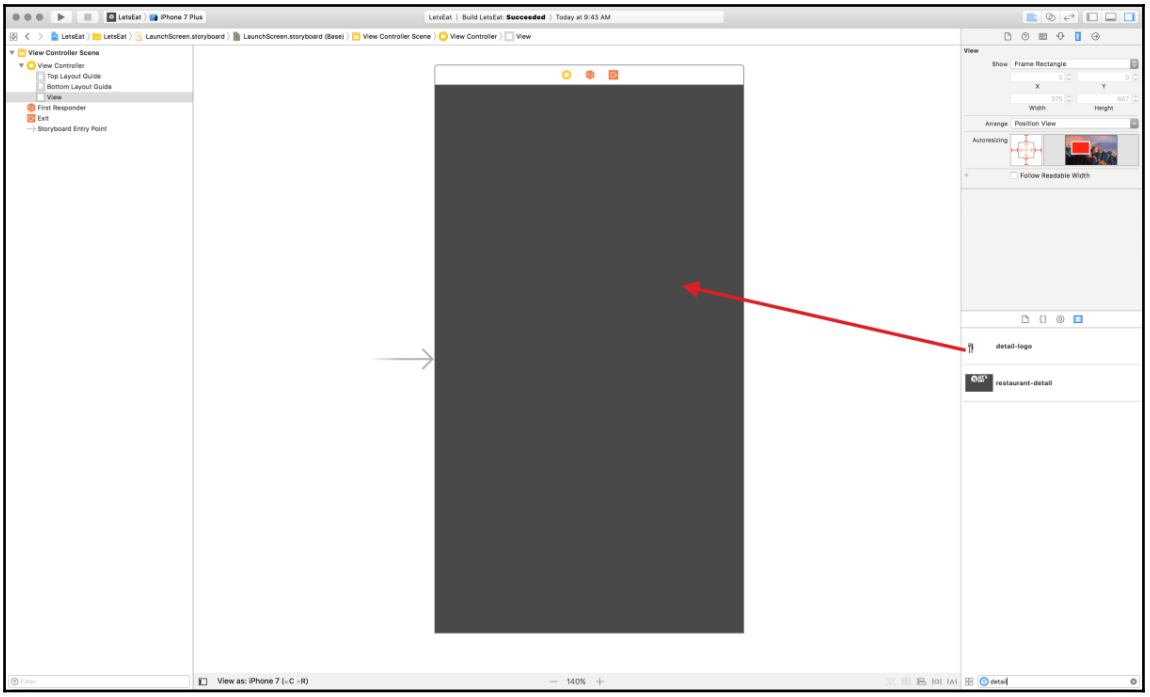

View Controller
View
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
0
Interaction
User Interaction Enabled
Multiple Touch
Alpha
1
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
0
X
0
Y
1
Width
1
Height
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.
Split View Controller - A composite view controller that manages left and right view controllers.
141%
Filter

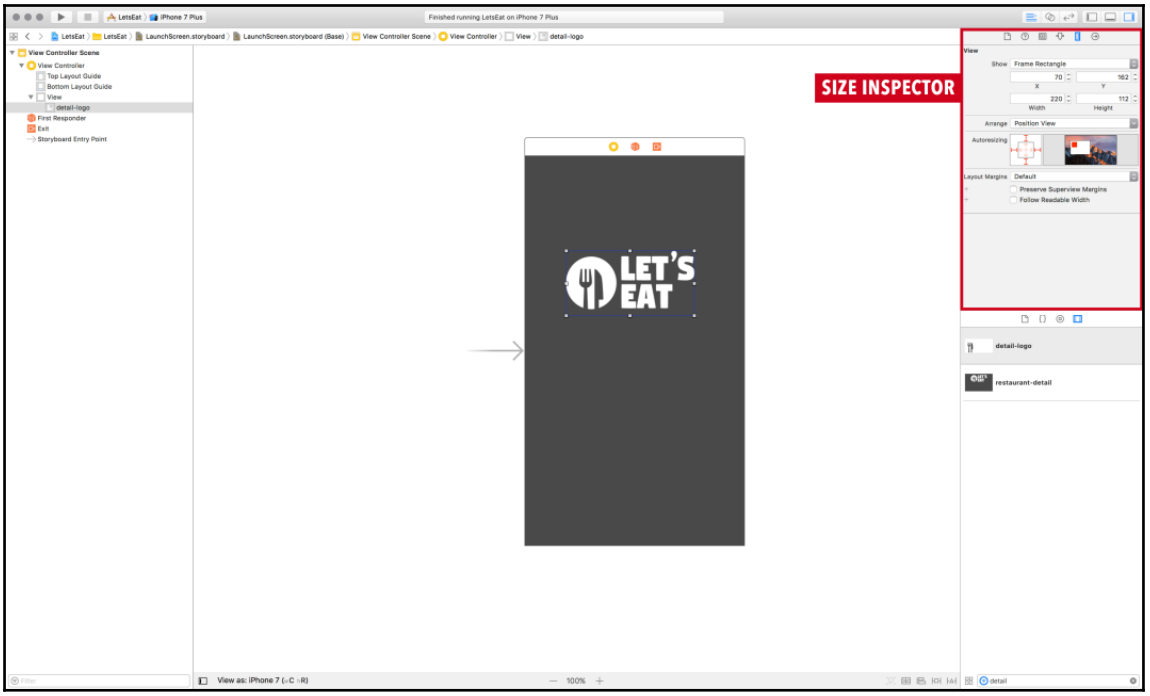

Colors
Opacity
100%

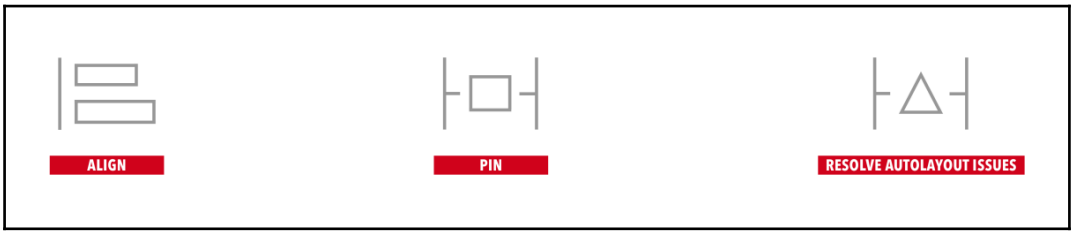

Colors
RGB Sliders
Red
255
Green
255
Blue
255
Hex Color #
FFFFFF
Opacity
100%


Colors
RGB Sliders
Red
74
Green
74
Blue
74
Hex Color #
AAAAAA
Opacity
100%
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
Width
Height
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

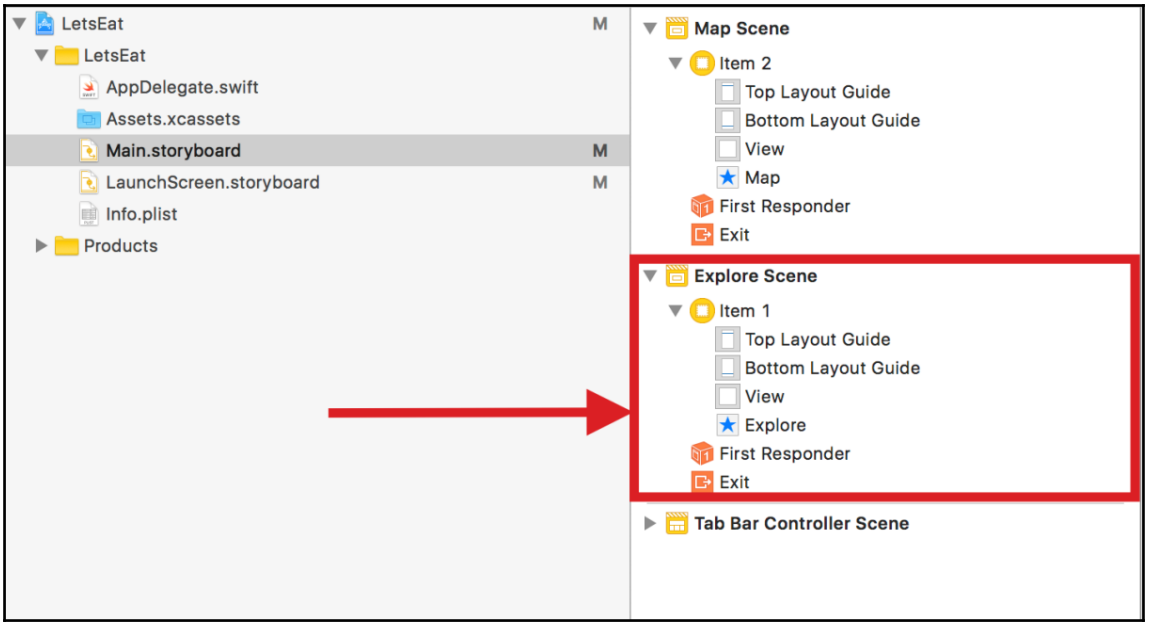

No Selection
Media Library
all
american
arrow
bar
bistro
brewery
burgers
californian
caribbean
comfort
continental
cuban
custom-annotation
Filter Bar
Filter

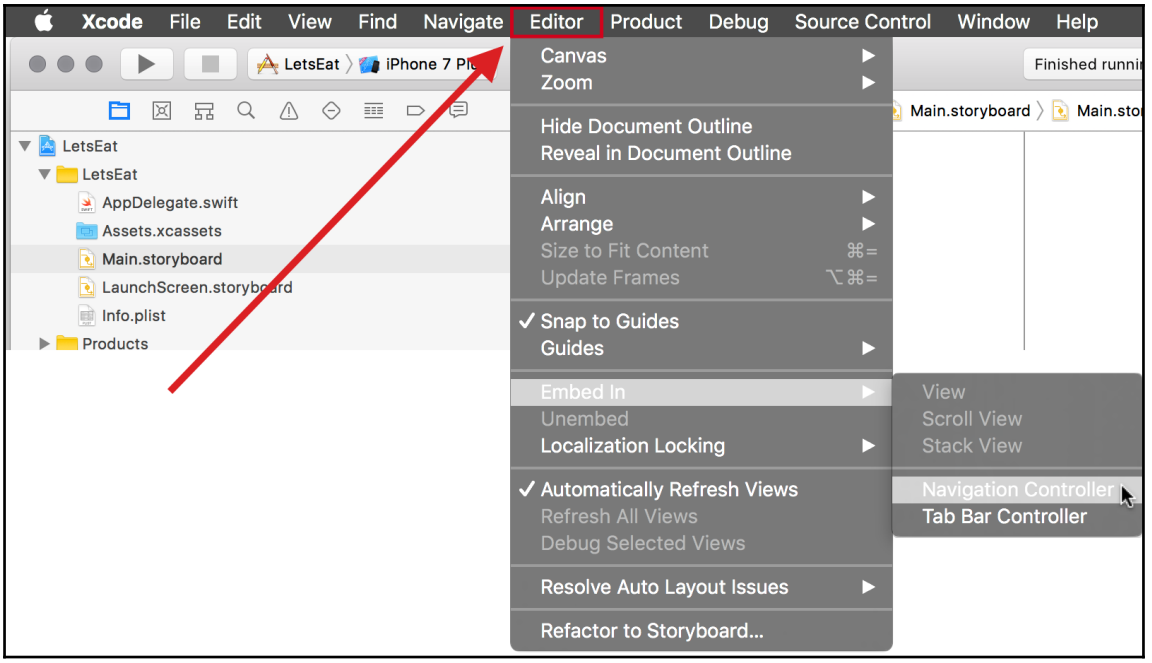

View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
First Responder
Exit
Storyboard Entry Point
Show
Frame Rectangle
Arrange
Position View
Autoresizing
Follow Readable Width
detail-logo
restaurant-detail
View as: iPhone 7 (wC hR)
140%

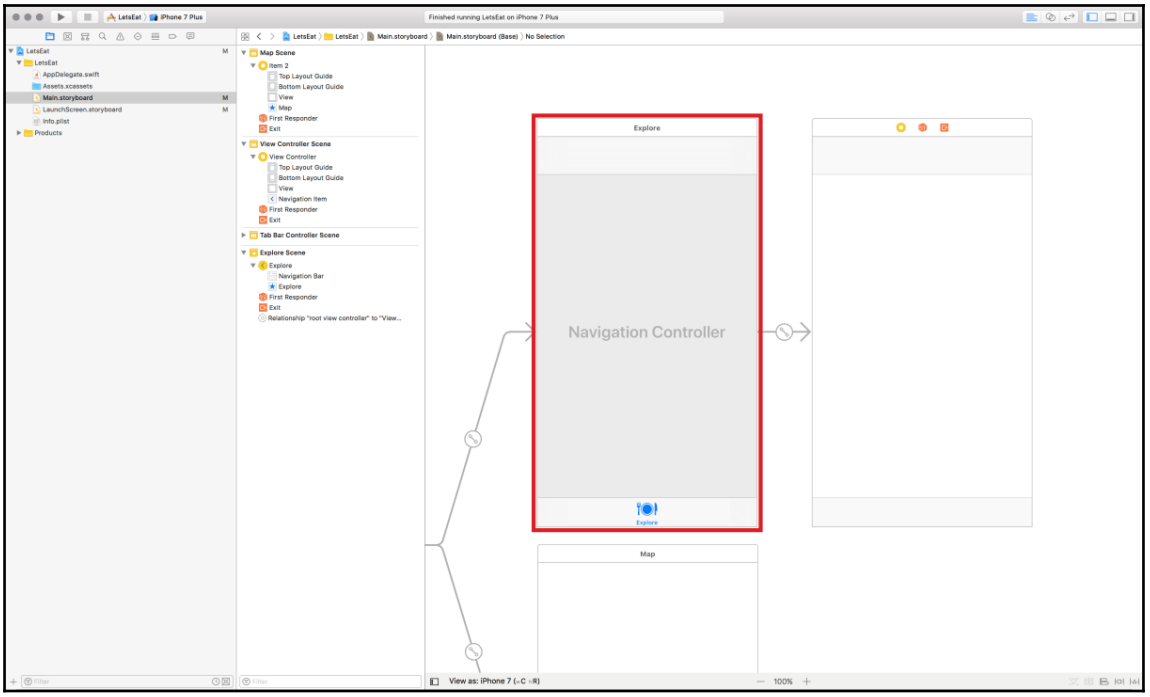

SIZE INSPECTOR
LET'S EAT
detail-logo
restaurant-detail

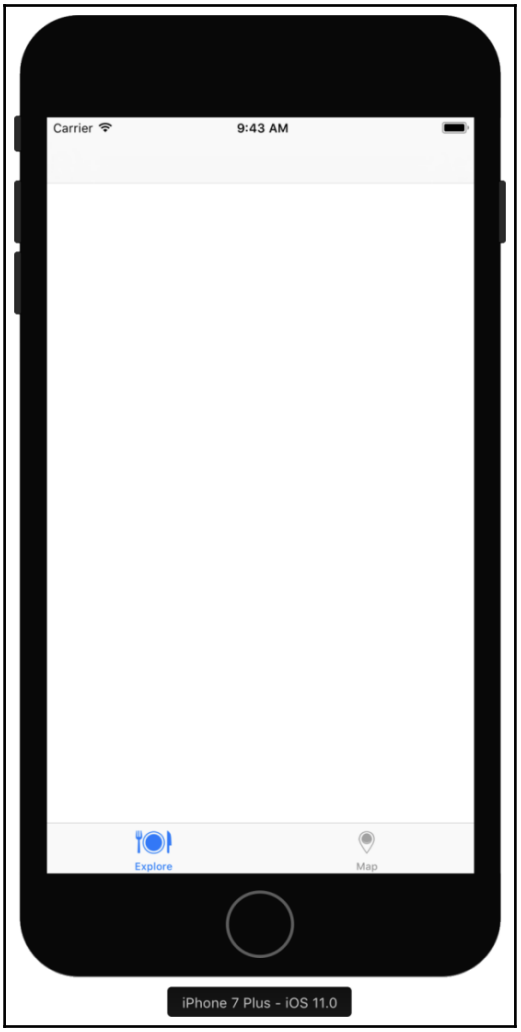

ALIGN
PIN
RESOLVE AUTOLAYOUT ISSUES


LET'S
EAT

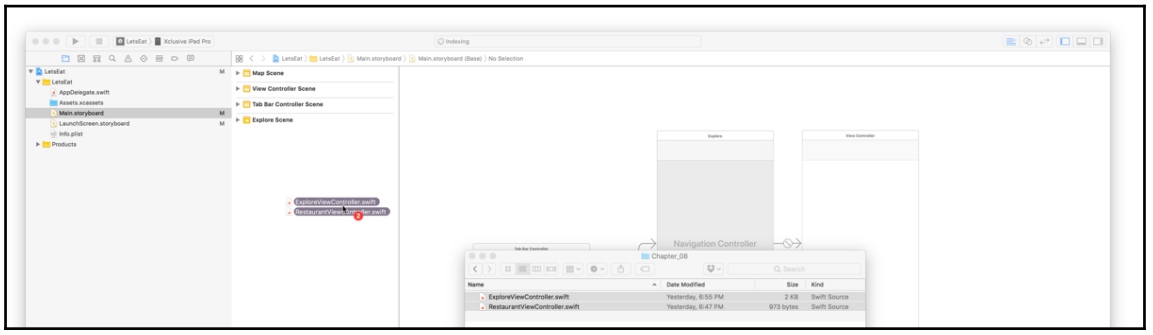

LetsEat
M
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
M
LaunchScreen.storyboard
M
Info.plist
Products
Map Scene
Item 2
Top Layout Guide
Bottom Layout Guide
View
Map
First Responder
Exit
Explore Scene
Item 1
Top Layout Guide
Bottom Layout Guide
View
Explore
First Responder
Exit
Tab Bar Controller Scene

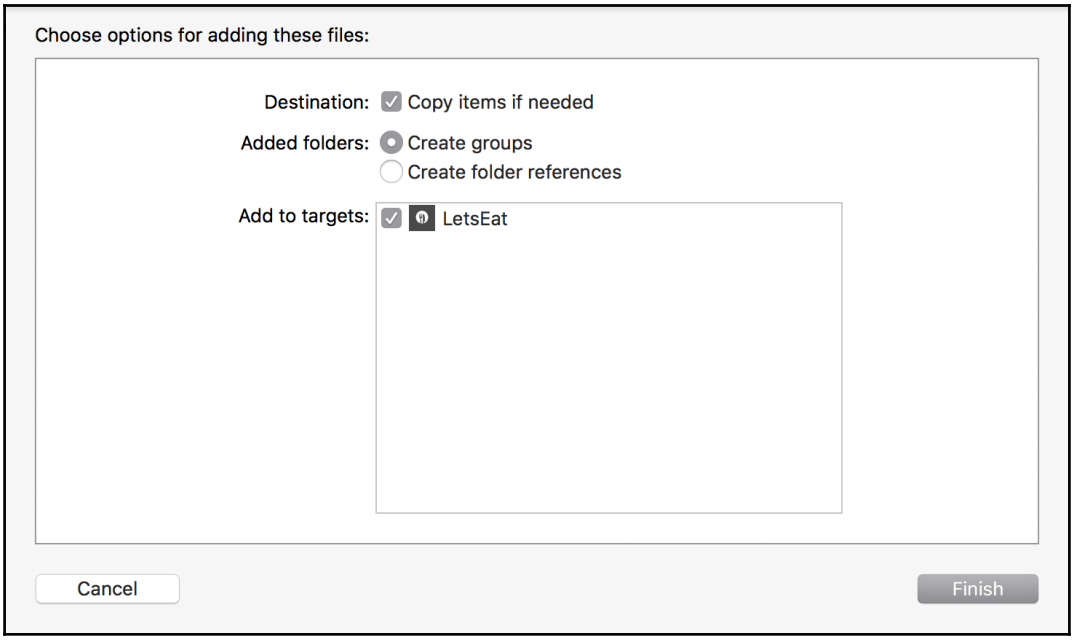

Xcode
File
Edit
View
Find
Navigate
Editor
Product
Debug
Source Control
Window
Help
LetsEat
iPhone 7 Pl
Finished runni
Main.storyboard
Main.sto
LetsEat
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
LaunchScreen.storyboard
Info.plist
Products
Canvas
Zoom
Hide Document Outline
Reveal in Document Outline
Align
Arrange
Size to Fit Content
Update Frames
Snap to Guides
Guides
Embed In
Unembed
Localization Locking
Automatically Refresh Views
Refresh All Views
Debug Selected Views
Resolve Auto Layout Issues
Refactor to Storyboard...
View
Scroll View
Stack View
Navigation Controller
Tab Bar Controller

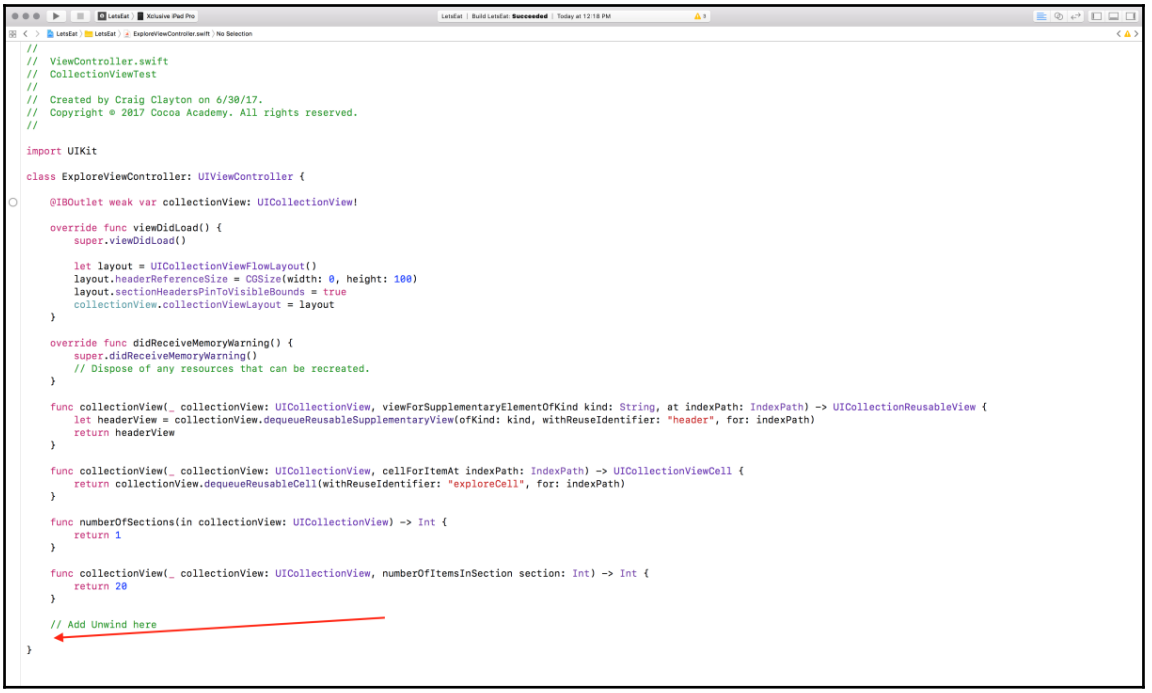

Explore
Navigation Controller
Map


Carrier
9:43 AM
Explore
Map
iPhone 7 Plus - iOS 11.0

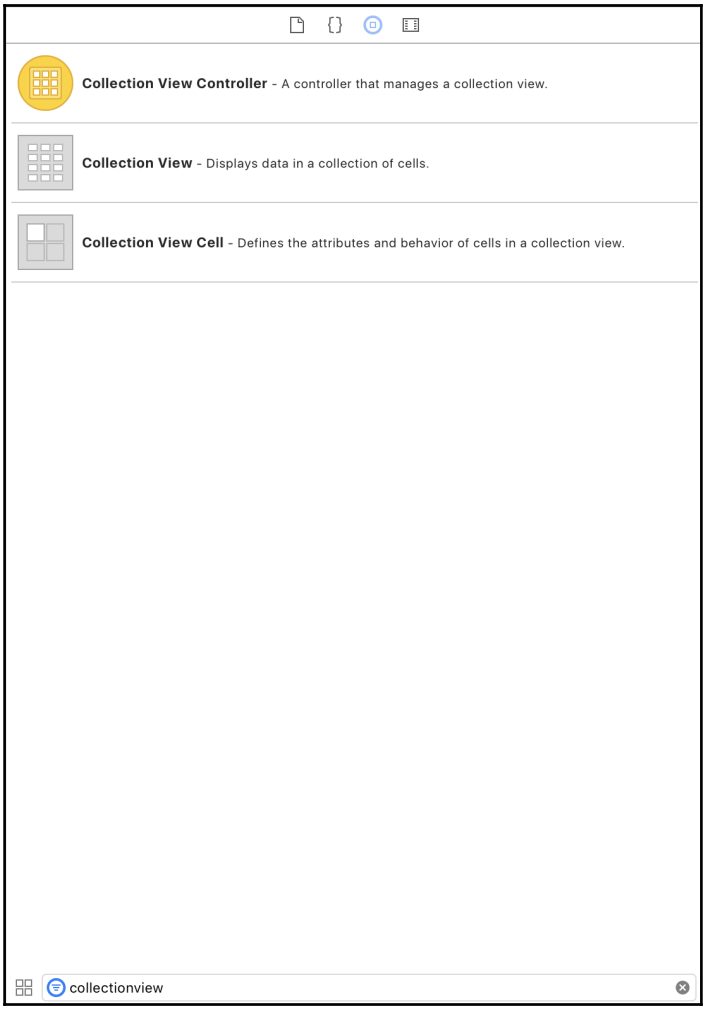
Chapter 8: Building Our App Structure in Storyboard

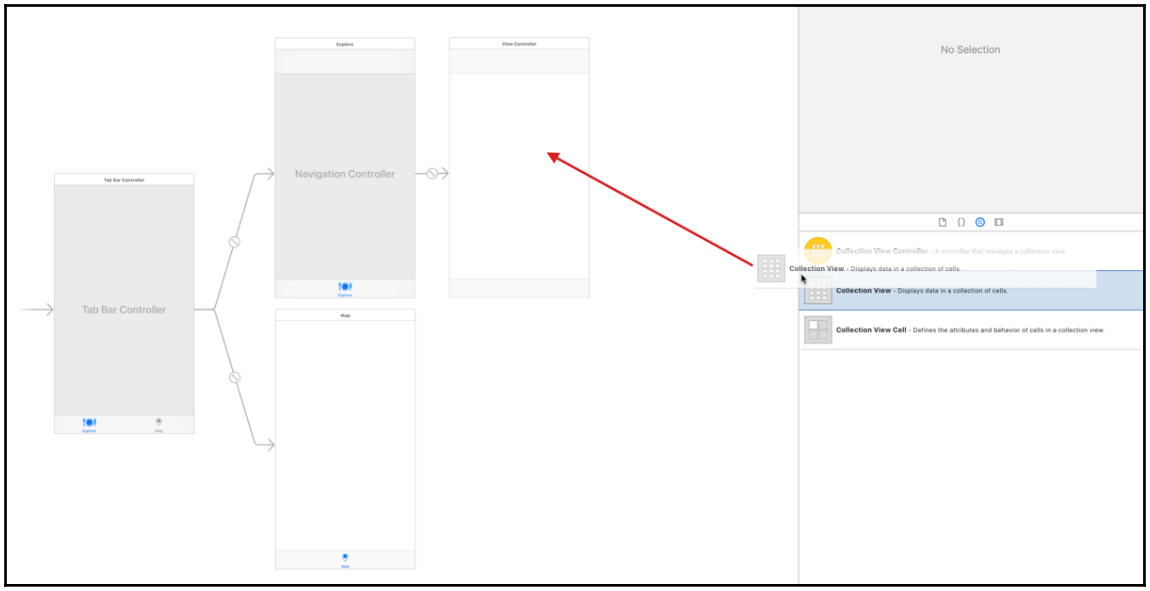


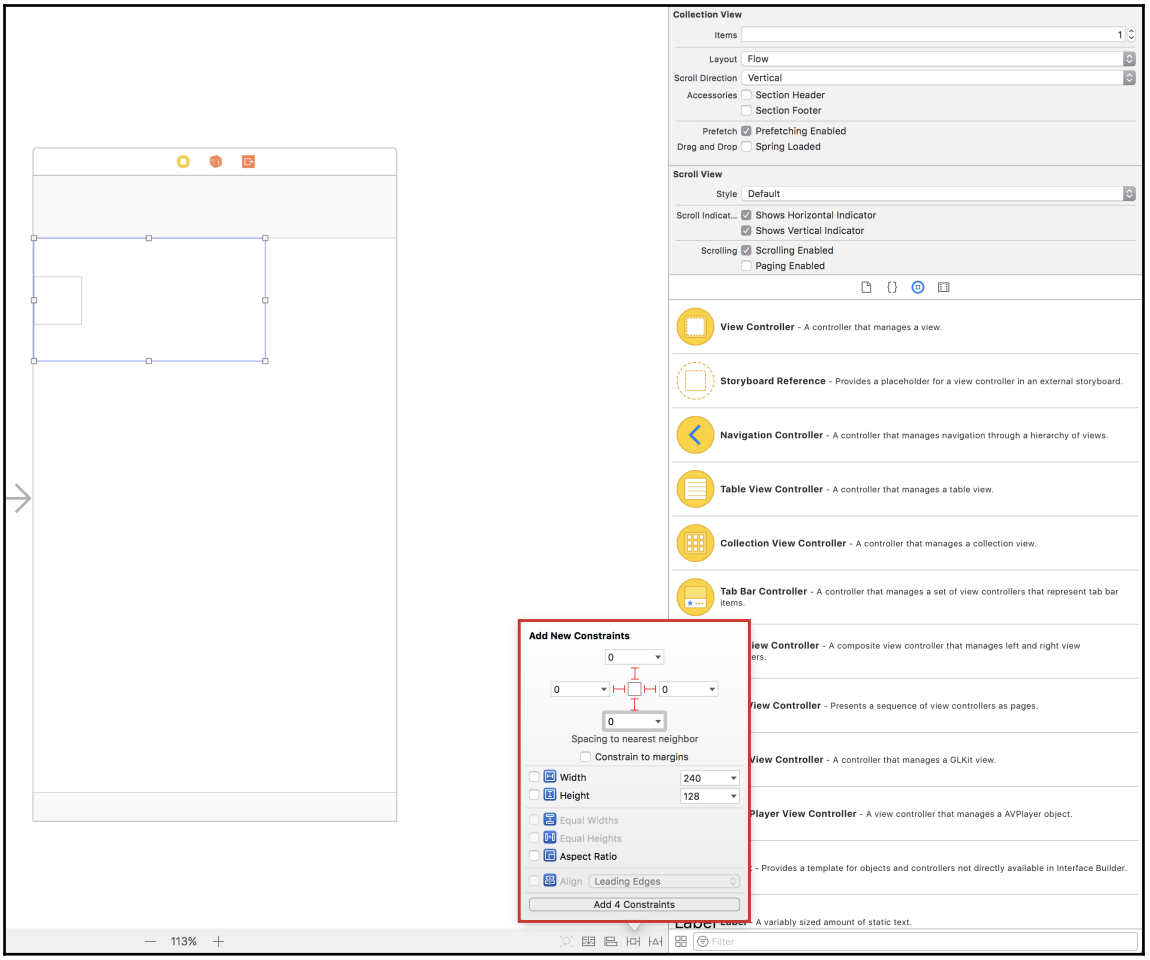


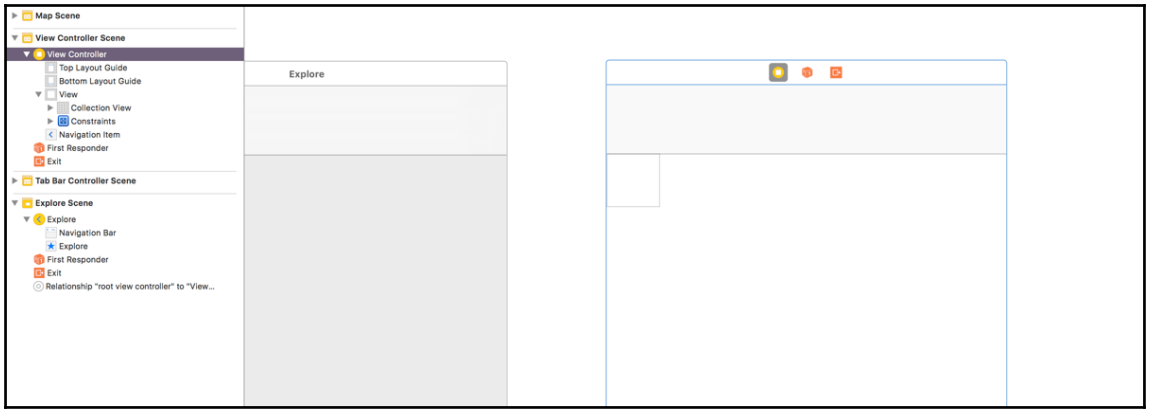


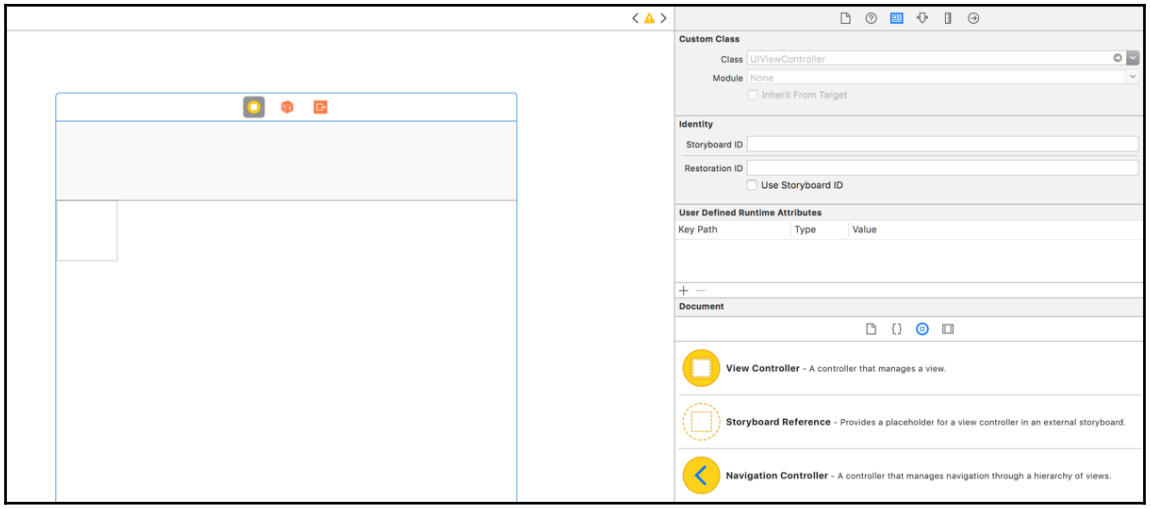
```swift
//
//  ViewController.swift
//  CollectionViewTest
//
//  Created by Craig Clayton on 6/30/17.
//  Copyright © 2017 Cocoa Academy. All rights reserved.
//

import UIKit

class ExploreViewController: UIViewController {

    @IBOutlet weak var collectionView: UICollectionView!

    override func viewDidLoad() {
        super.viewDidLoad()

        let layout = UICollectionViewFlowLayout()
        layout.headerReferenceSize = CGSize(width: 0, height: 100)
        layout.sectionHeadersPinToVisibleBounds = true
        collectionView.collectionViewLayout = layout
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func collectionView(_ collectionView: UICollectionView, viewForSupplementaryElementOfKind kind: String, at indexPath: IndexPath) -> UICollectionReusableView {
        let headerView = collectionView.dequeueReusableSupplementaryView(ofKind: kind, withReuseIdentifier: "header", for: indexPath)
        return headerView
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        return collectionView.dequeueReusableCell(withReuseIdentifier: "exploreCell", for: indexPath)
    }

    func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return 20
    }

    // Add Unwind here
}
```

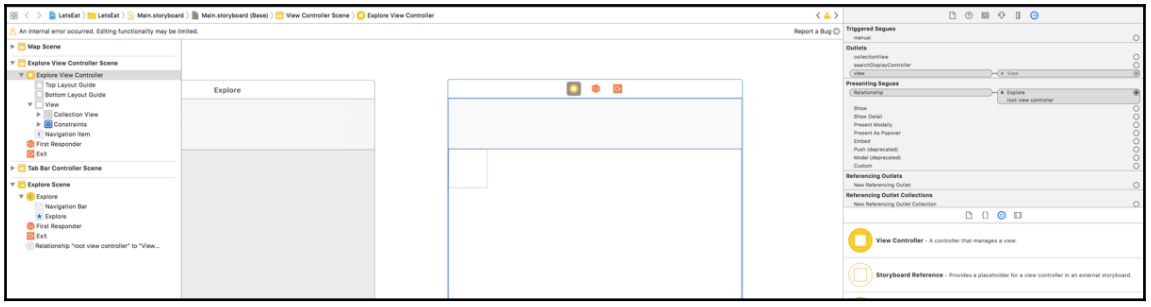

Collection View Controller - A controller that manages a collection view.
Collection View - Displays data in a collection of cells.
Collection View Cell - Defines the attributes and behavior of cells in a collection view.
collectionview

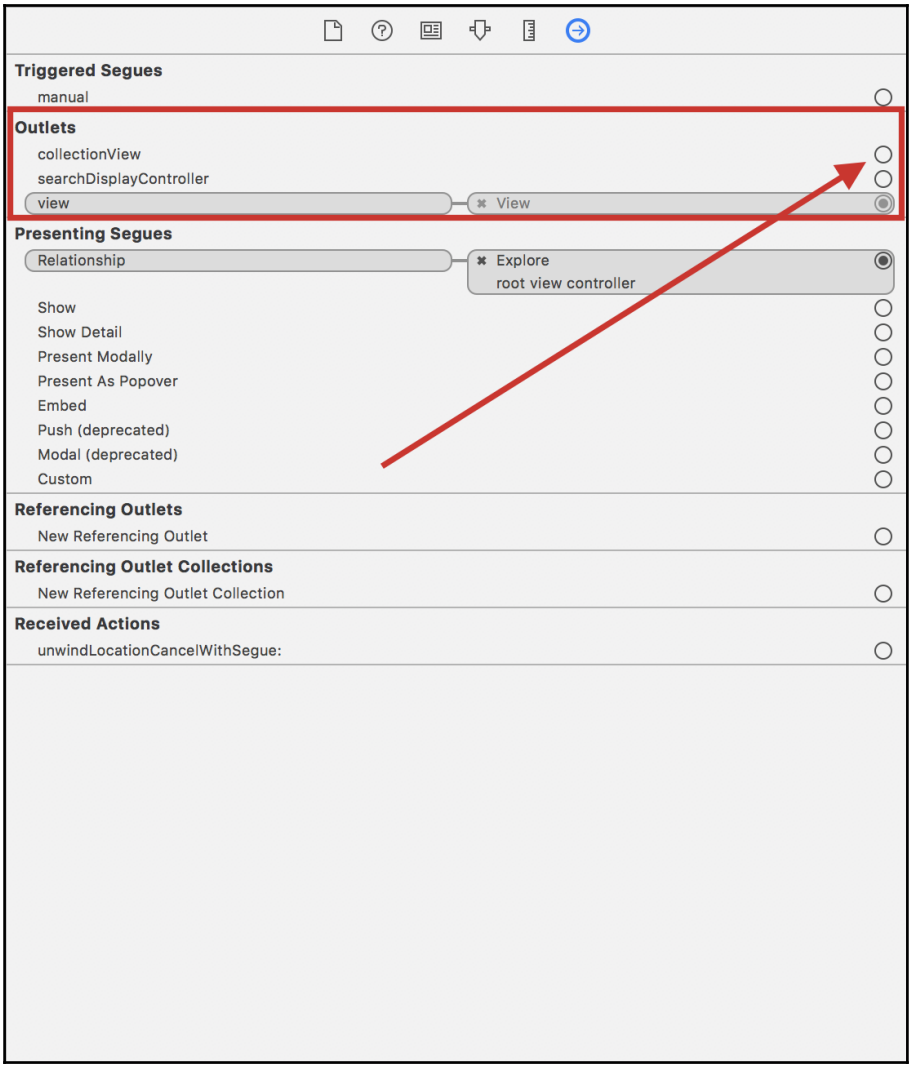

Tab Bar Controller
Navigation Controller
No Selection
Collection View Controller - A controller that manages a collection view.
Collection View - Displays data in a collection of cells.
Collection View - Displays data in a collection of cells.
Collection View Cell - Defines the attributes and behavior of cells in a collection view.

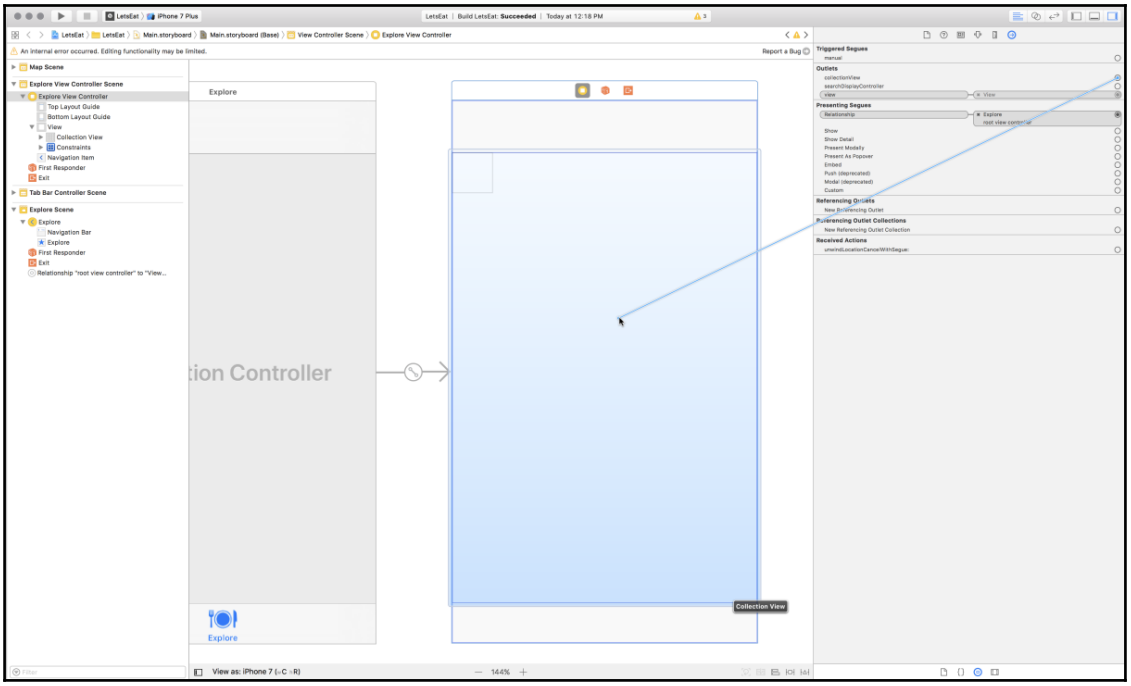

Collection View
Items
1
Layout
Flow
Scroll Direction
Vertical
Accessories
Section Header
Section Footer
Prefetch
Prefetching Enabled
Drag and Drop
Spring Loaded
Scroll View
Style
Default
Scroll Indicat...
Shows Horizontal Indicator
Shows Vertical Indicator
Scrolling
Scrolling Enabled
Paging Enabled
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Collection View Controller - A controller that manages a collection view.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.
Add New Constraints
0
0
0
0
Spacing to nearest neighbor
Constrain to margins
Width
240
Height
128
Equal Widths
Equal Heights
Aspect Ratio
Align
Leading Edges
Add 4 Constraints
113%
Filter

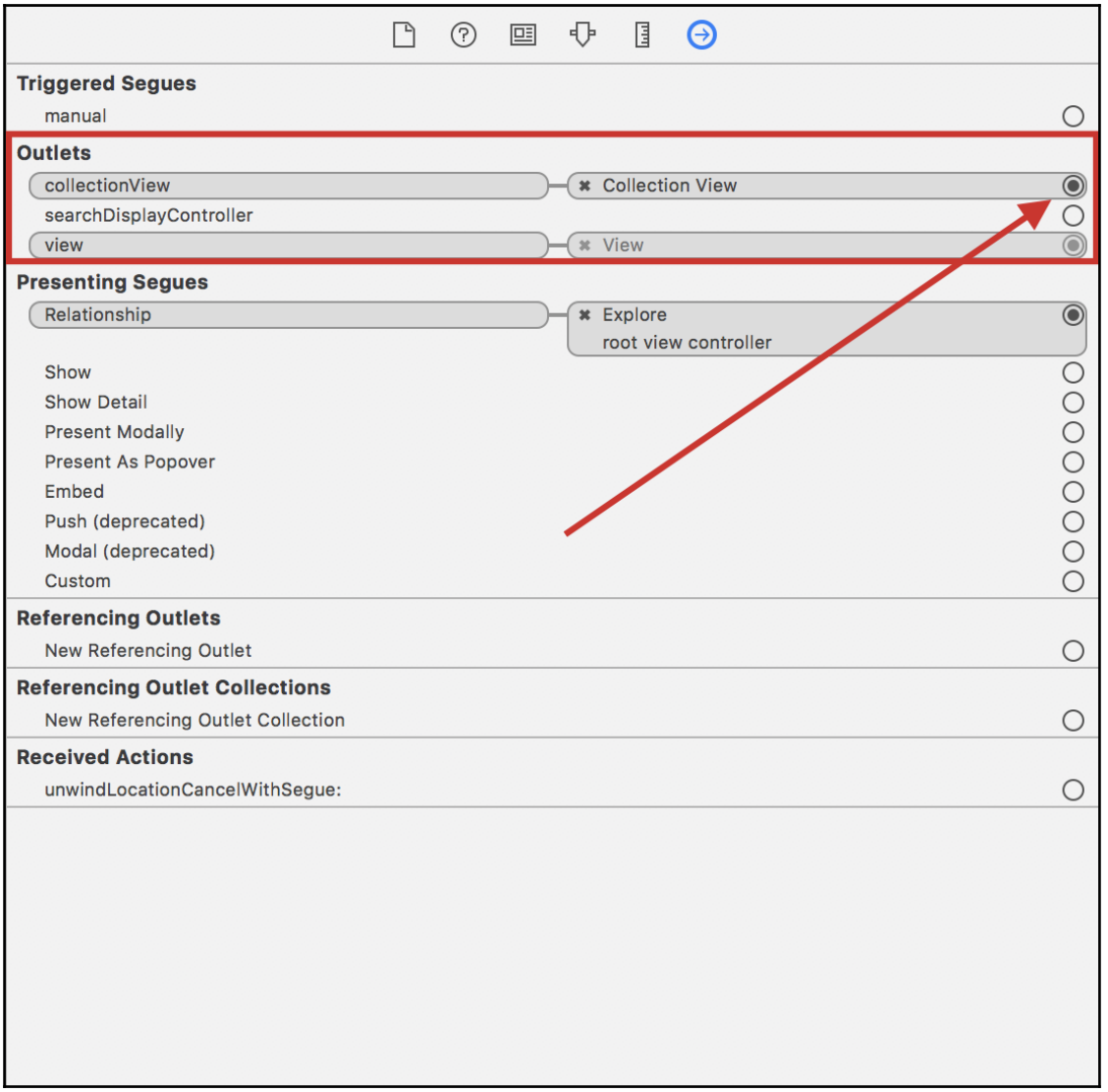

Map Scene
View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
Collection View
Constraints
Navigation Item
First Responder
Exit
Tab Bar Controller Scene
Explore Scene
Explore
Navigation Bar
Explore
First Responder
Exit
Relationship "root view controller" to "View...
Explore

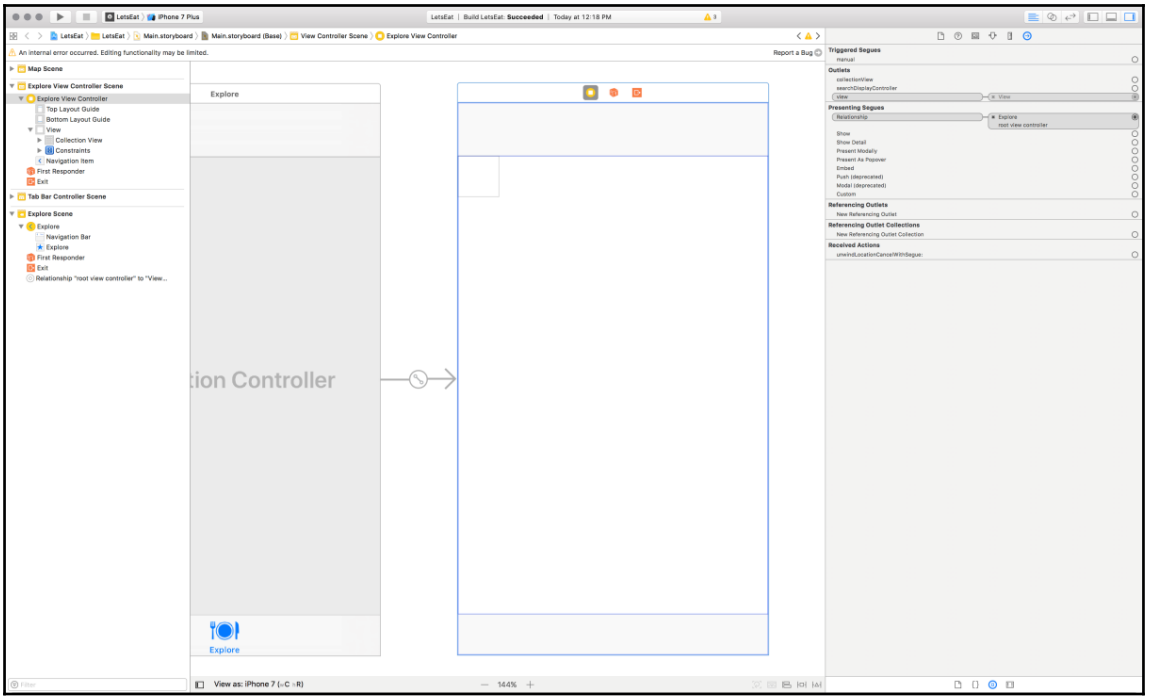

Custom Class
Class UIViewController
Module None
Inherit From Target
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
View Controller - A controller that manages a view.
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.
Navigation Controller - A controller that manages navigation through a hierarchy of views.

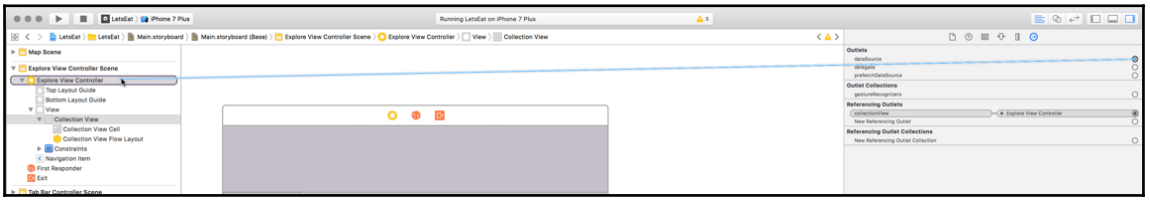

Explore

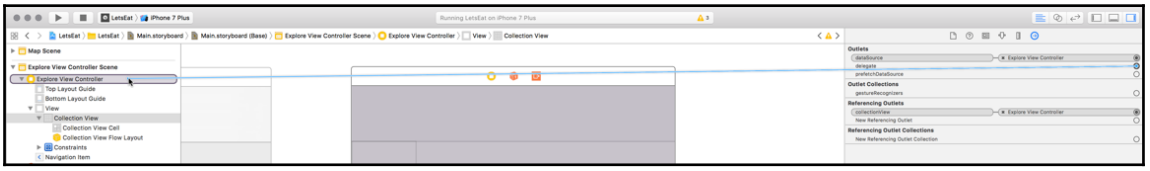

Triggered Segues
manual
Outlets
collectionView
searchDisplayController
view
View
Presenting Segues
Relationship
Explore
root view controller
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
unwindLocationCancelWithSegue:

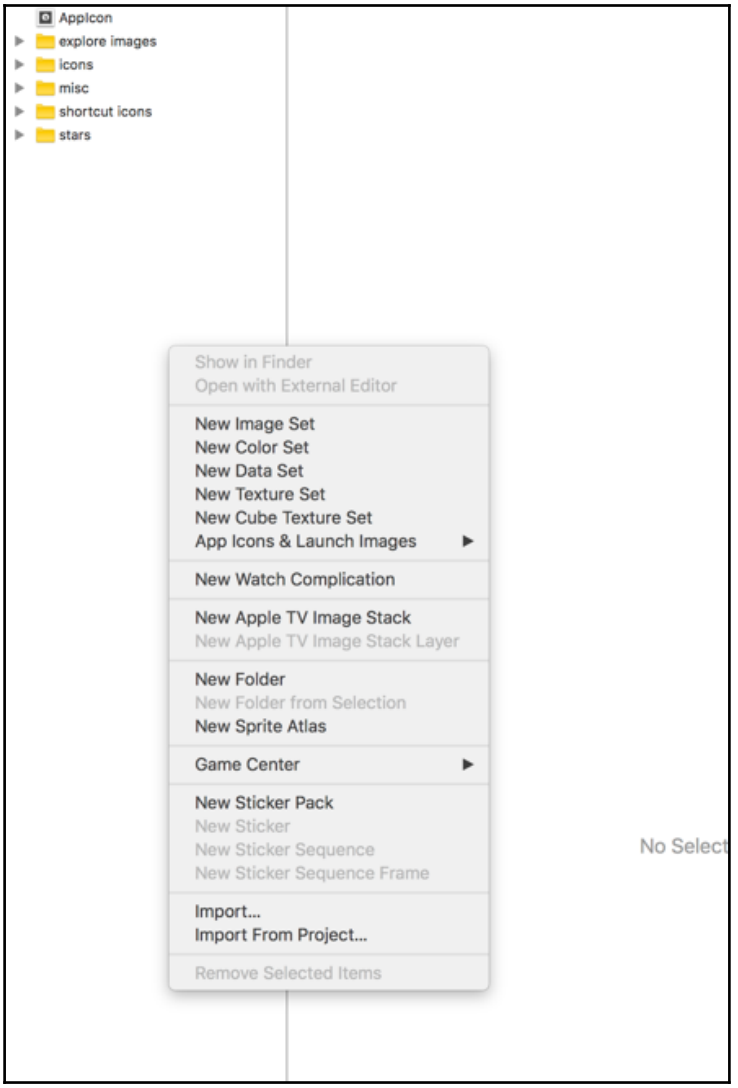

Explore
tion Controller
Explore
Collection View


Triggered Segues
manual
Outlets
collectionView
Collection View
searchDisplayController
view
View
Presenting Segues
Relationship
Explore
root view controller
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
unwindLocationCancelWithSegue:


Explore
ion Controller
Explore

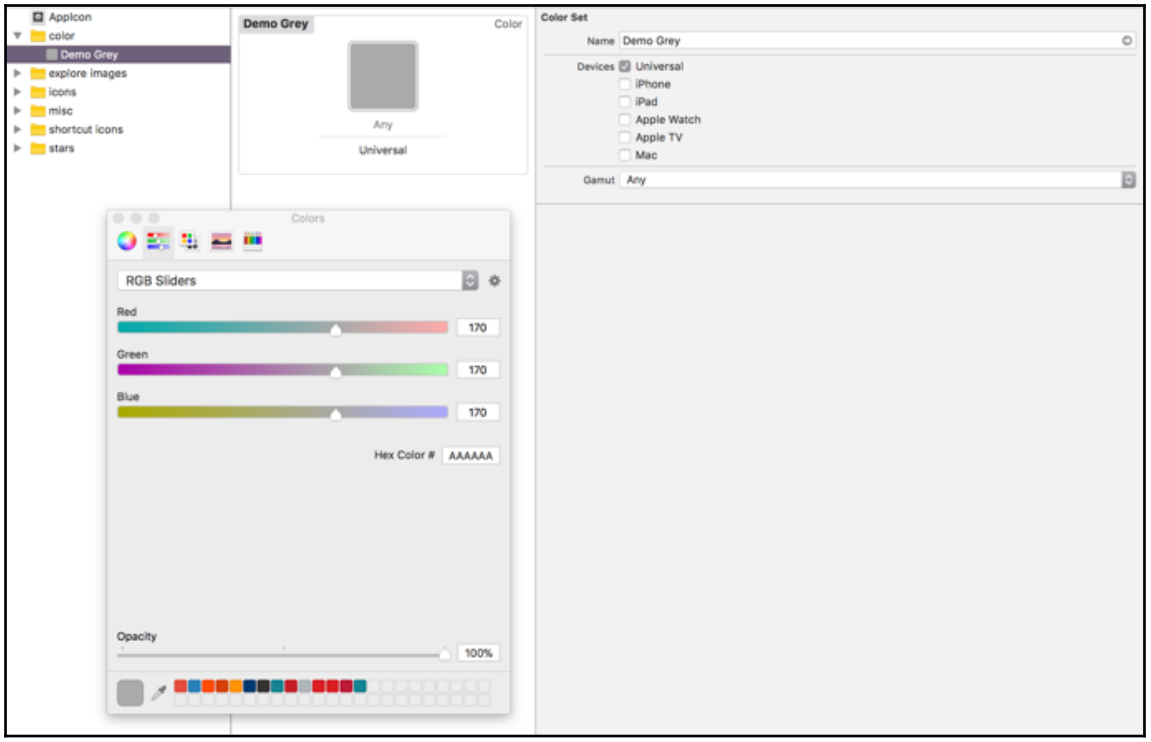


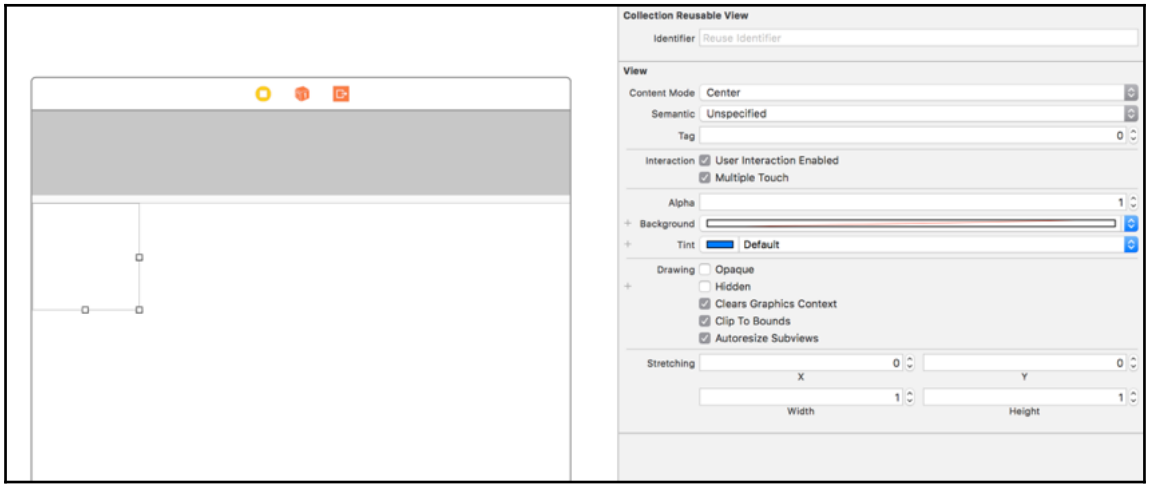


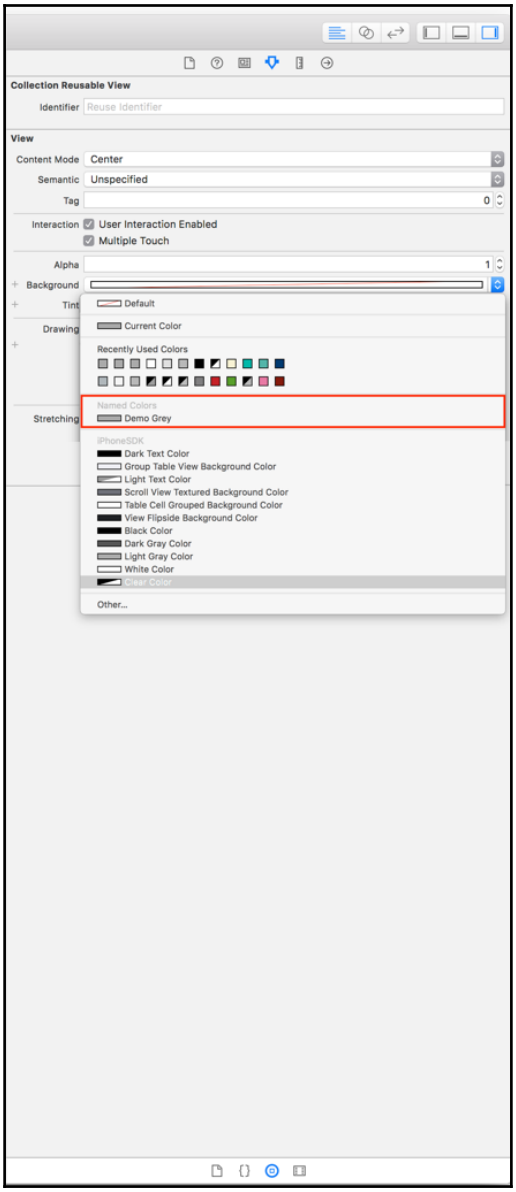
2. 
AppIcon
explore images
icons
misc
shortcut icons
stars
Show in Finder
Open with External Editor
New Image Set
New Color Set
New Data Set
New Texture Set
New Cube Texture Set
App Icons & Launch Images
New Watch Complication
New Apple TV Image Stack
New Apple TV Image Stack Layer
New Folder
New Folder from Selection
New Sprite Atlas
Game Center
New Sticker Pack
New Sticker
New Sticker Sequence
New Sticker Sequence Frame
Import...
Import From Project...
Remove Selected Items
No Select


Applcon
color
Color
explore images
icons
misc
shortcut icons
stars
Color
Color
Any
Universal
Color Set
Name Color
Devices Universal
iPhone
iPad
Apple Watch
Apple TV
Mac
Gamut Any
Color
Color Space sRGB
Color
Input Method Floating point (0.0-1.0)
Red 1.000
Green 1.000
Blue 1.000
Opacity 100.0%

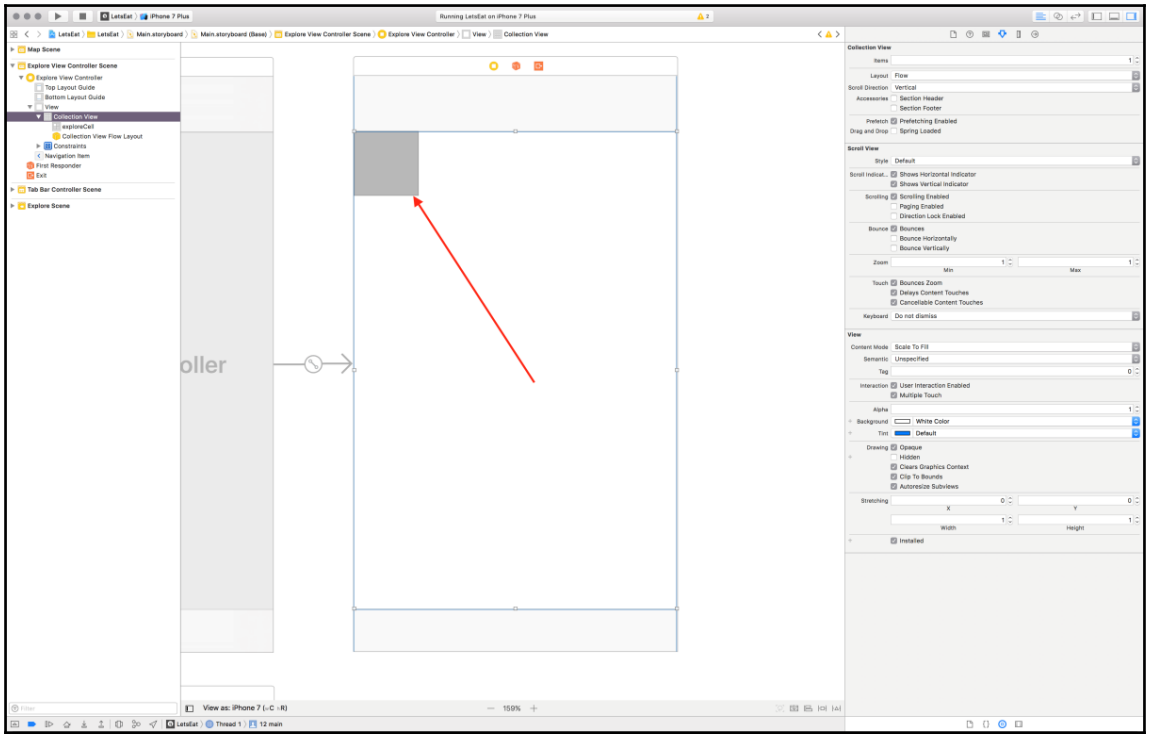

Colors
Opacity
100%

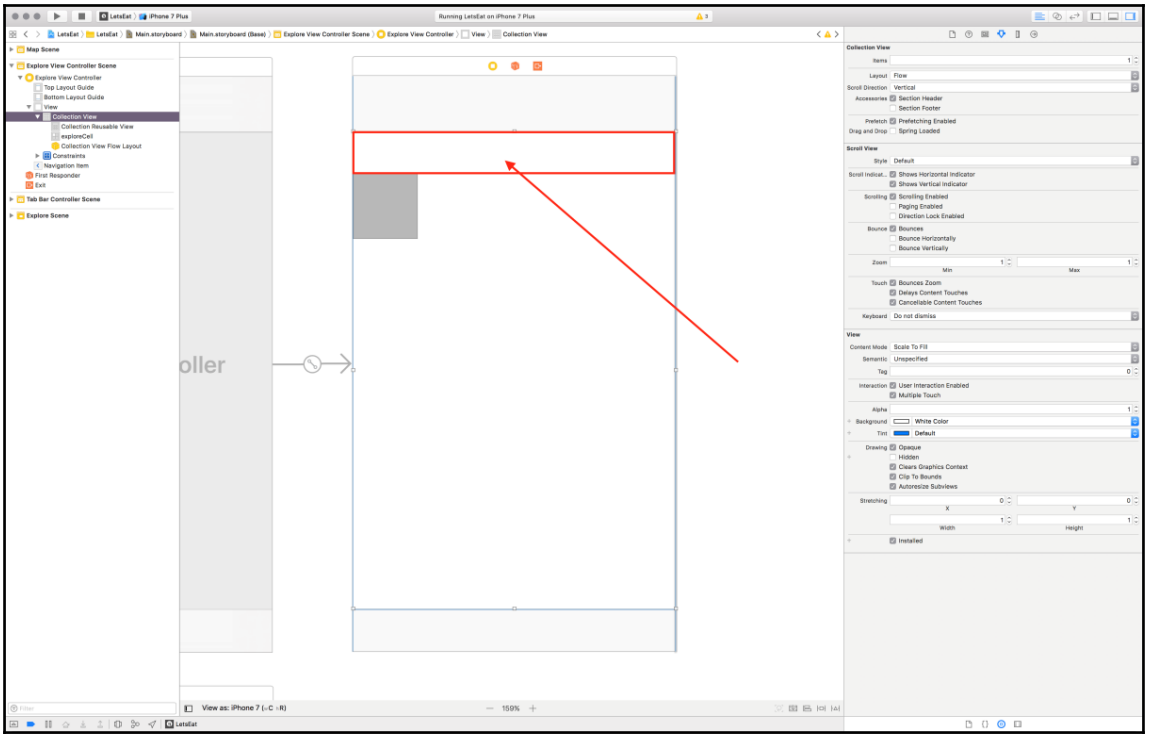

AppIcon
color
Demo Grey
explore images
icons
misc
shortcut icons
stars
Demo Grey
Color
Any
Universal
Color Set
Name
Demo Grey
Devices
Universal
iPhone
iPad
Apple Watch
Apple TV
Mac
Gamut
Any
Colors
RGB Sliders
Red
170
Green
170
Blue
170
Hex Color #
AAAAAA
Opacity
100%

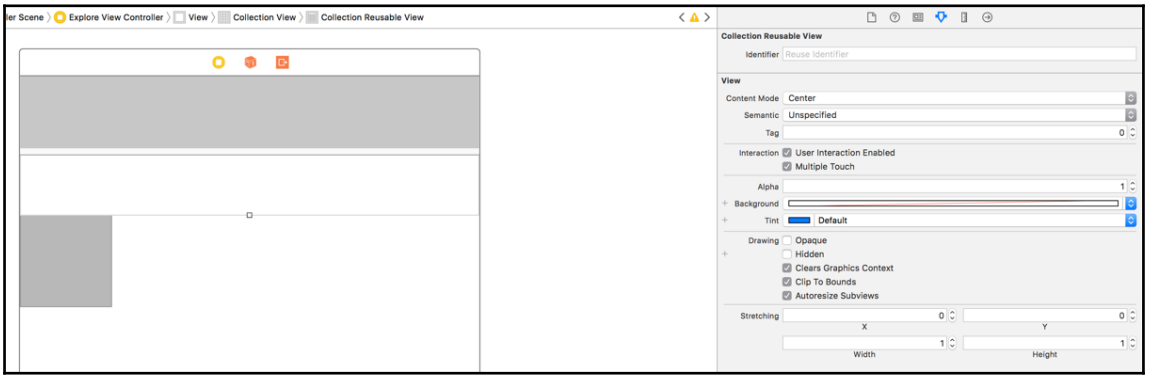

Collection Reusable View
Identifier
Reuse Identifier
View
Content Mode
Center
Semantic
Unspecified
Tag
0
Interaction
User Interaction Enabled
Multiple Touch
Alpha
1
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
0
X
0
Y
1
Width
1
Height

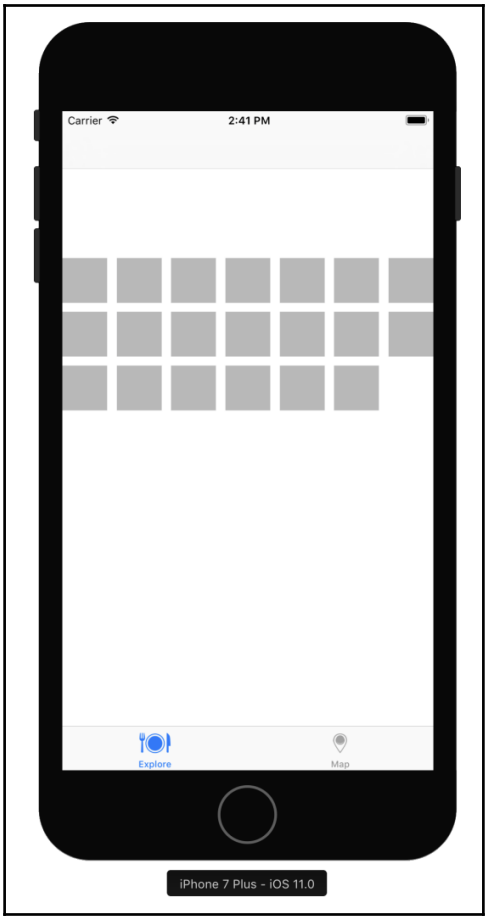

Collection Reusable View
Identifier
Reuse Identifier
View
Content Mode
Center
Semantic
Unspecified
Tag
0
Interaction
User Interaction Enabled
Multiple Touch
Alpha
1
Background
Tint
Drawing
Stretching
Default
Current Color
Recently Used Colors
Named Colors
Demo Grey
iPhoneSDK
Dark Text Color
Group Table View Background Color
Light Text Color
Scroll View Textured Background Color
Table Cell Grouped Background Color
View Flipside Background Color
Black Color
Dark Gray Color
Light Gray Color
White Color
Clear Color
Other...


Collection Reusable View
Identifier
exploreCell
View
Content Mode
Center
Semantic
Unspecified
Tag
0
Interaction
User Interaction Enabled
Multiple Touch
Alpha
1
Background
Light Gray Color
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
0
X
0
Y
1
Width
1
Height


oller



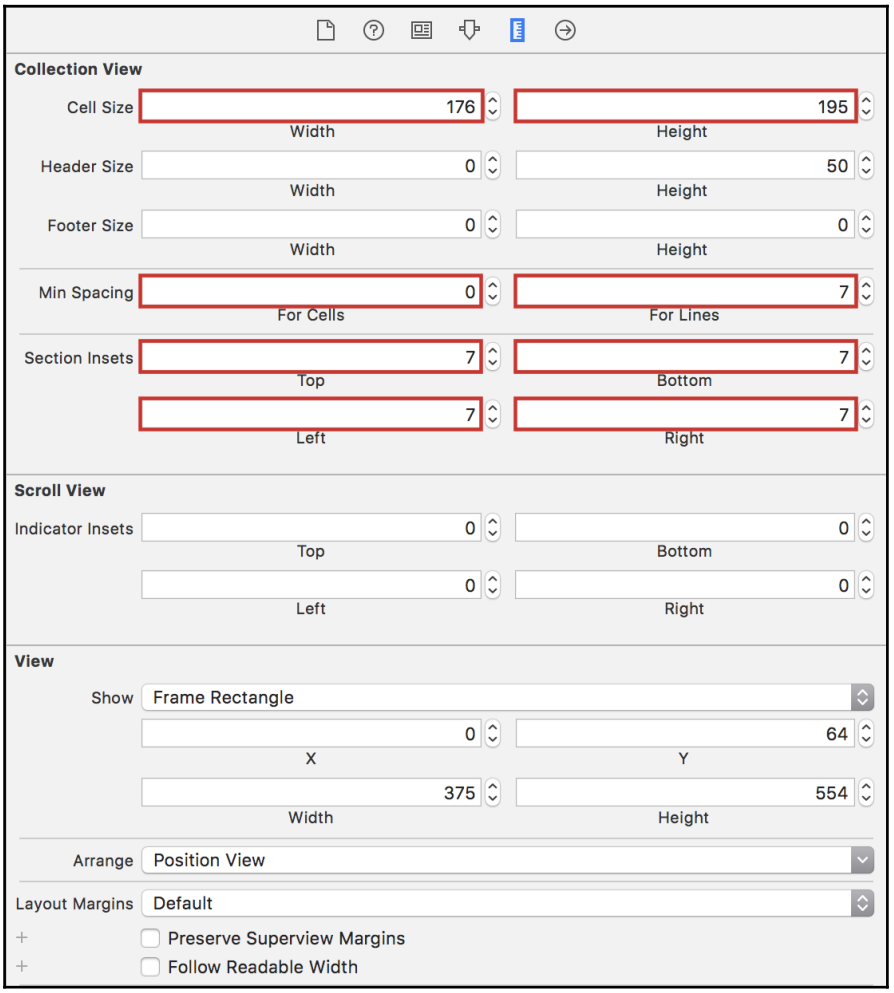




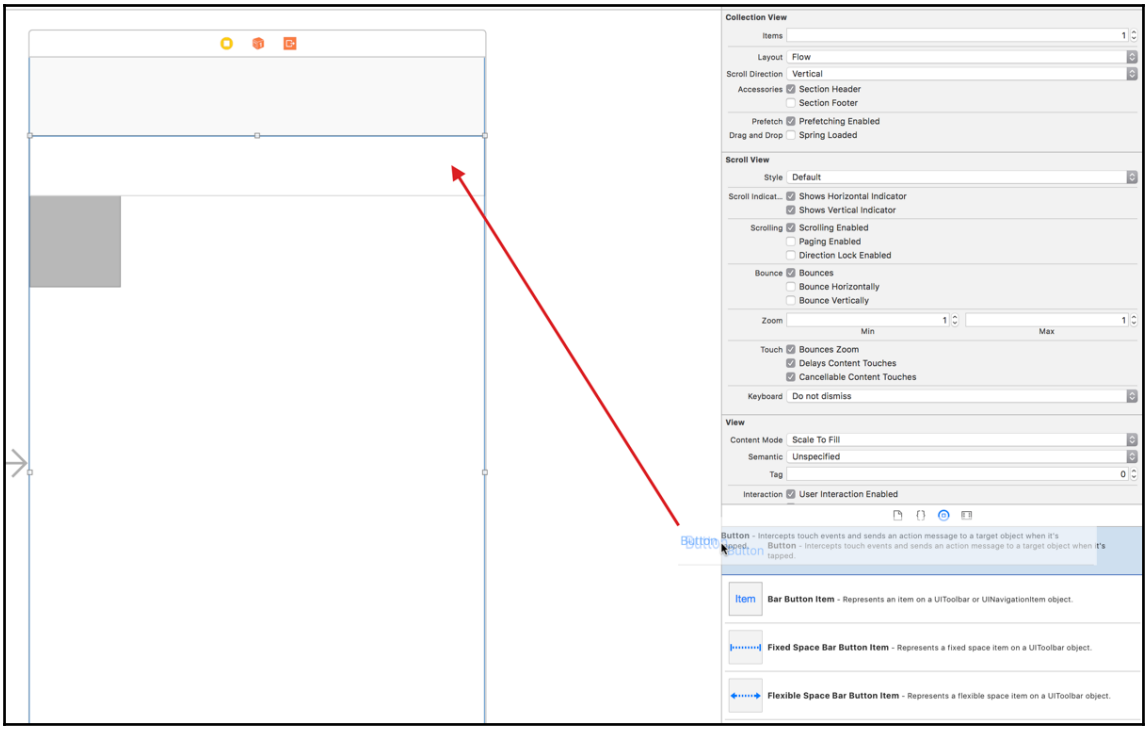
Fields	Values			
Cell Size	Width: 176	Height: 195		
Min Spacing	For Cells: 0	For Lines: 7		
Section Insets	Top: 7	Bottom: 7	Left: 7	Right: 7

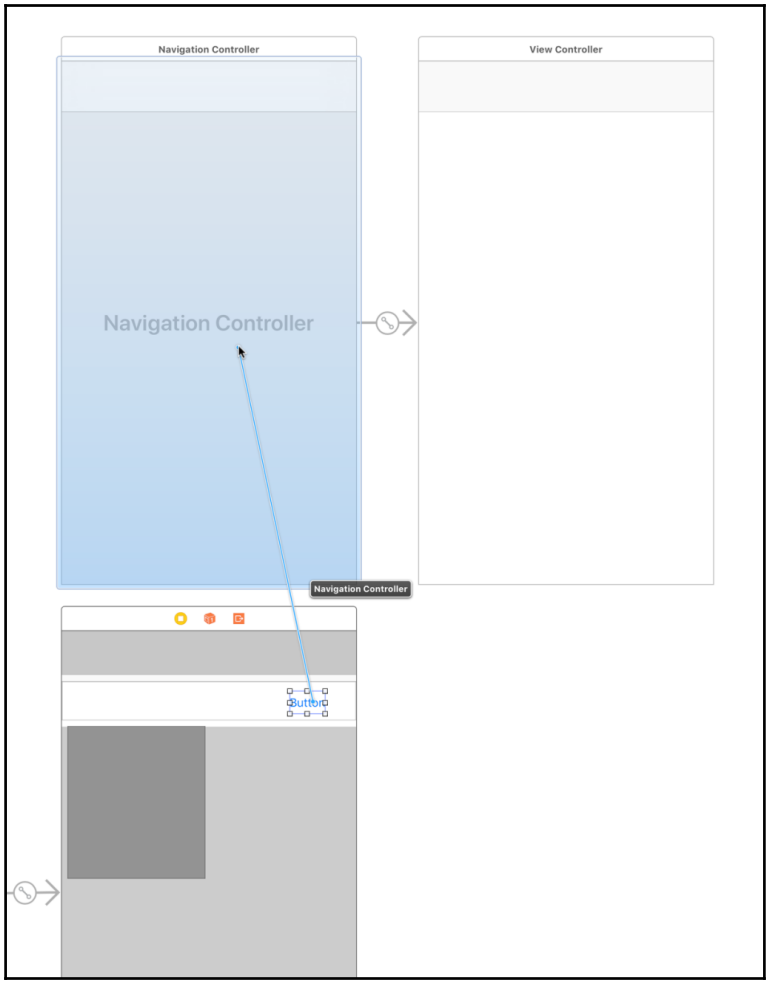
Fields	Values			
Cell Size	Width: 196	Height: 154		
Min Spacing	For Cells: 0	For Lines: 7		
Section Insets	Top: 7	Bottom: 7	Left: 7	Right: 7

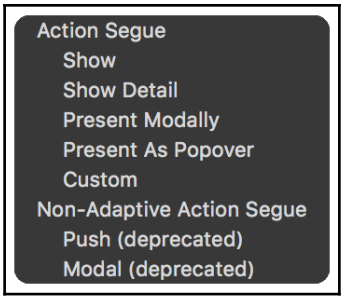
Fields	Values			
Cell Size	Width: 150	Height: 154		
Min Spacing	For Cells: 0	For Lines: 7		
Section Insets	Top: 7	Bottom: 7	Left: 7	Right: 7

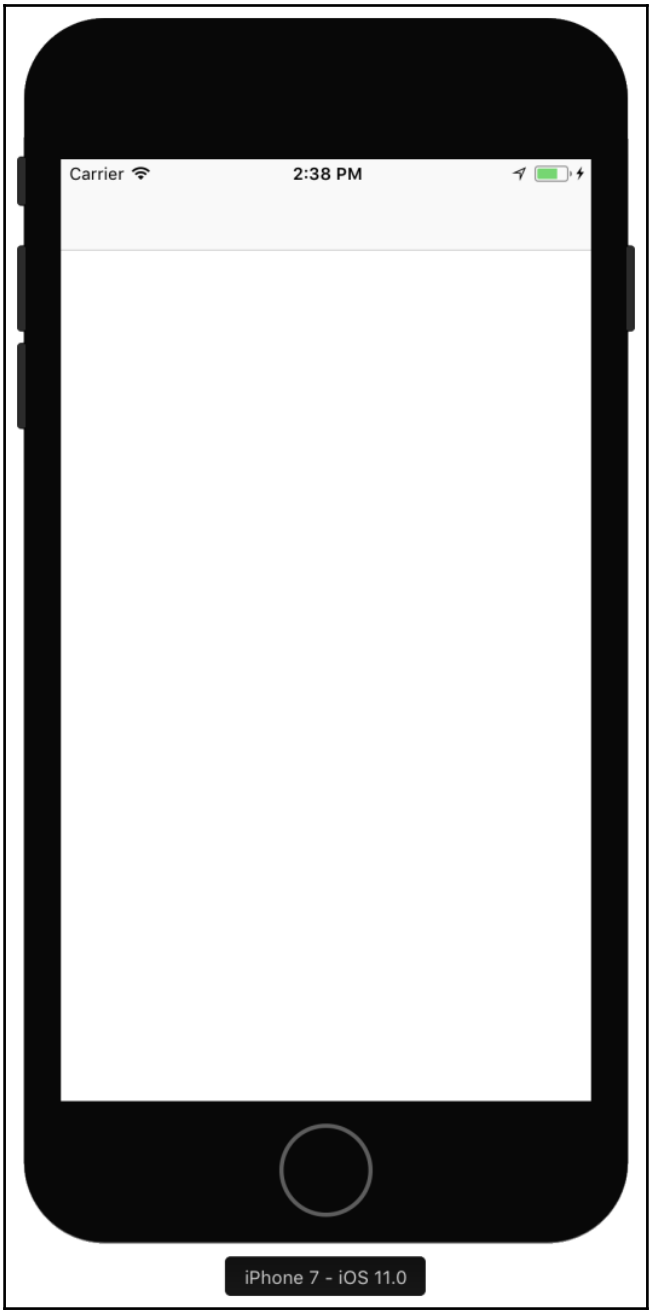


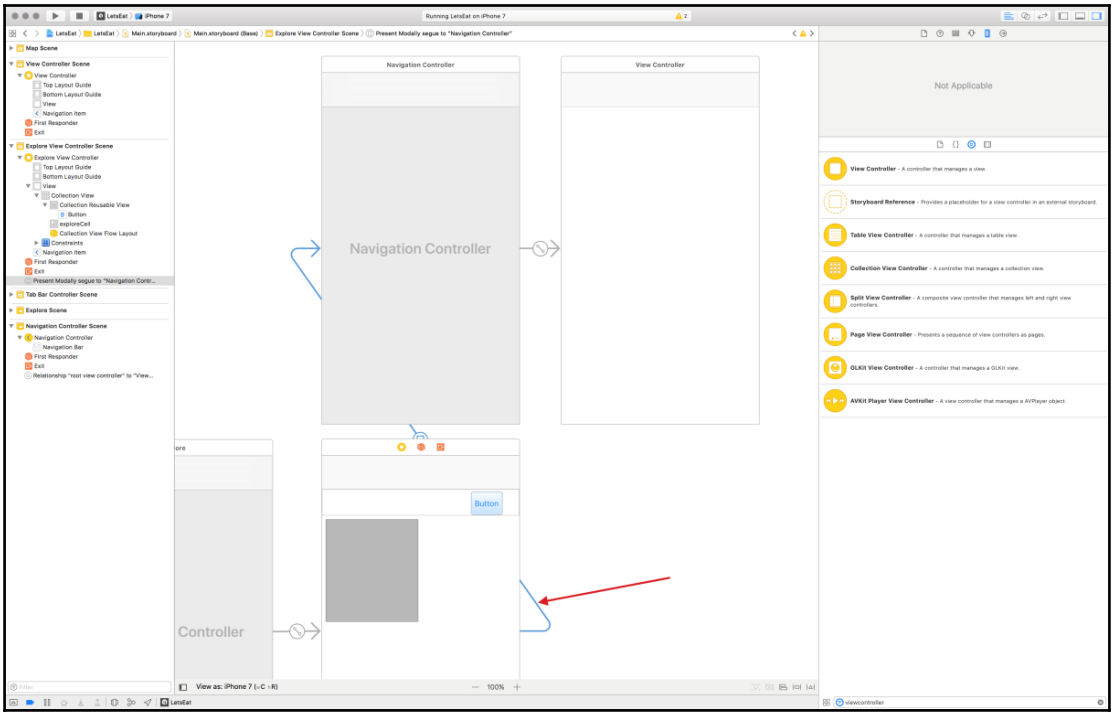

4:30 AM
LOS ANGELES, CA
Explore
+ LOCATION
Pizza
Steak
Seafood
American
Explore
Map

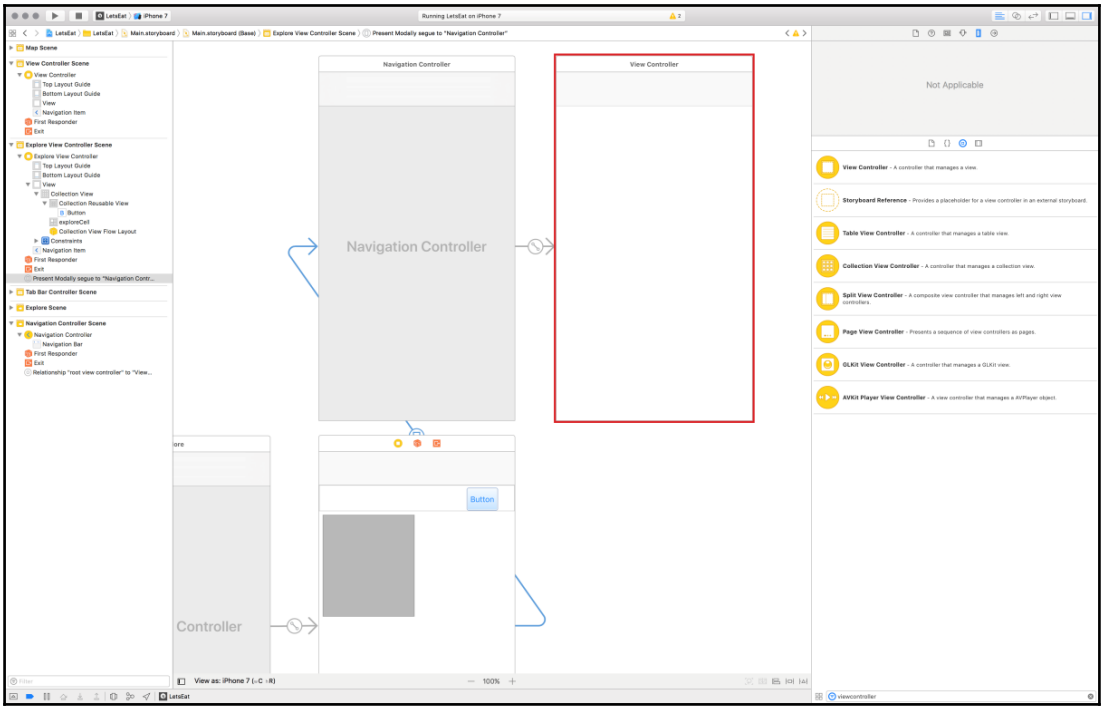

Collection View
Items
1
Layout
Flow
Scroll Direction
Vertical
Accessories
Section Header
Section Footer
Prefetch
Prefetching Enabled
Drag and Drop
Spring Loaded
Scroll View
Style
Default
Scroll Indicat...
Shows Horizontal Indicator
Shows Vertical Indicator
Scrolling
Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Bounce
Bounces
Bounce Horizontally
Bounce Vertically
Zoom
1
Min
1
Max
Touch
Bounces Zoom
Delays Content Touches
Cancellable Content Touches
Keyboard
Do not dismiss
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
0
Interaction
User Interaction Enabled
Button - Intercepts touch events and sends an action message to a target object when it's tapped.
Button - Intercepts touch events and sends an action message to a target object when it's tapped.
Item
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.
Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.
Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.

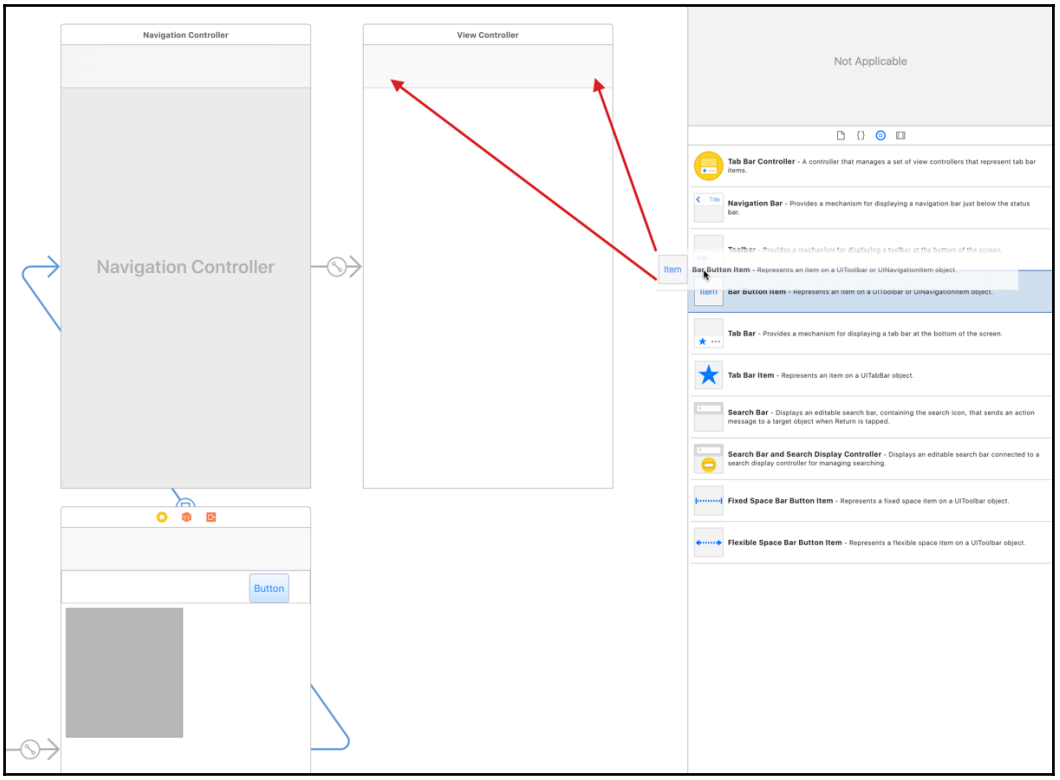

Navigation Controller
View Controller
Navigation Controller
Navigation Controller
Button

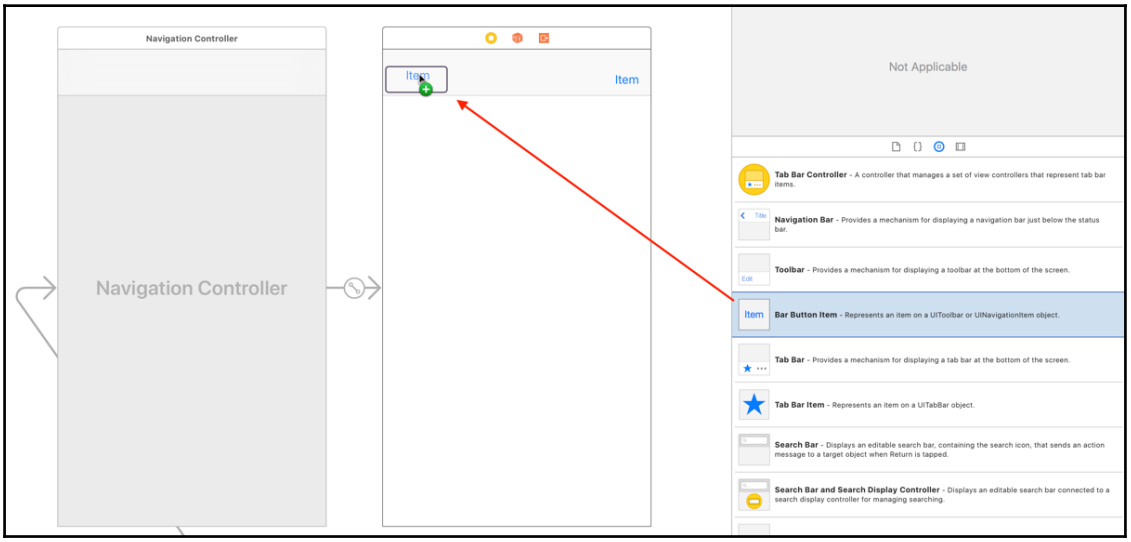

Action Segue
Show
Show Detail
Present Modally
Present As Popover
Custom
Non-Adaptive Action Segue
Push (deprecated)
Modal (deprecated)

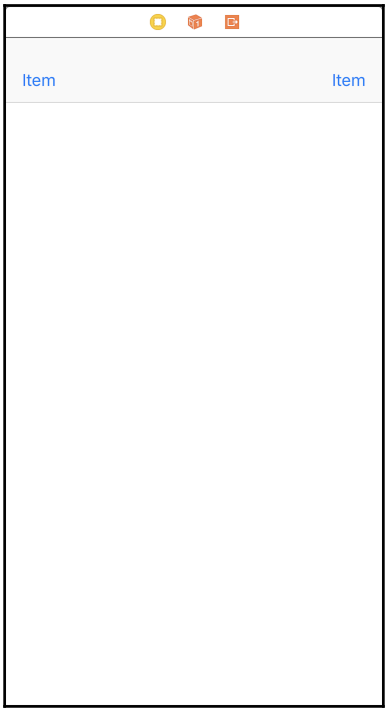

Carrier
2:38 PM
iPhone 7 - iOS 11.0

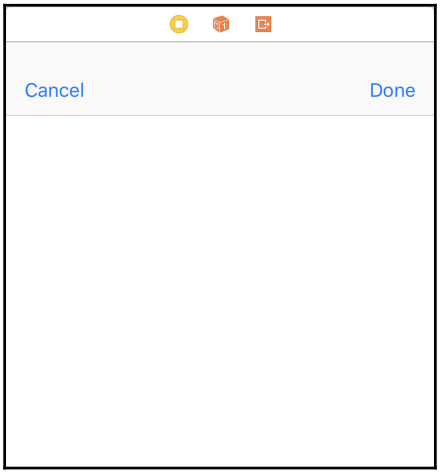

Navigation Controller
View Controller
Navigation Controller
Button
Controller
Not Applicable

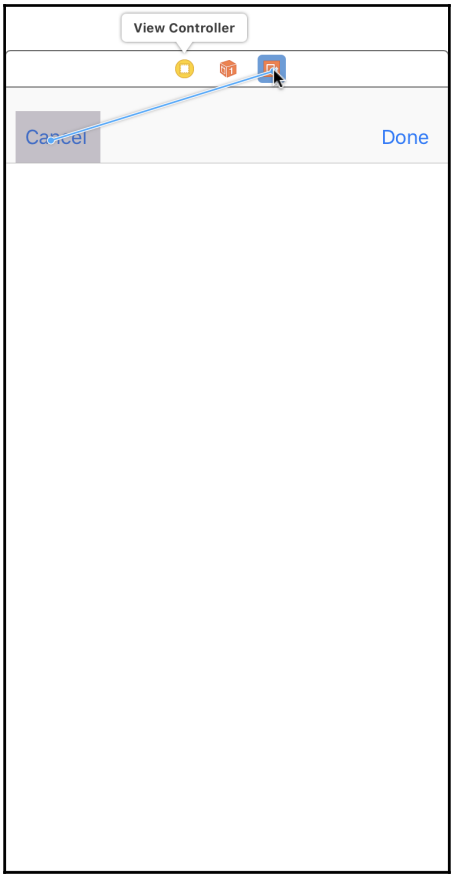

Navigation Controller
View Controller
Navigation Controller
Button
Controller
Not Applicable


Navigation Controller
View Controller
Navigation Controller
Button
Not Applicable
Item
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.
Navigation Bar - Provides a mechanism for displaying a navigation bar just below the status bar.
Tab Bar - Provides a mechanism for displaying a tab bar at the bottom of the screen.
Tab Bar Item - Represents an item on a UITabBar object.
Search Bar - Displays an editable search bar, containing the search icon, that sends an action message to a target object when Return is tapped.
Search Bar and Search Display Controller - Displays an editable search bar connected to a search display controller for managing searching.
Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.
Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.

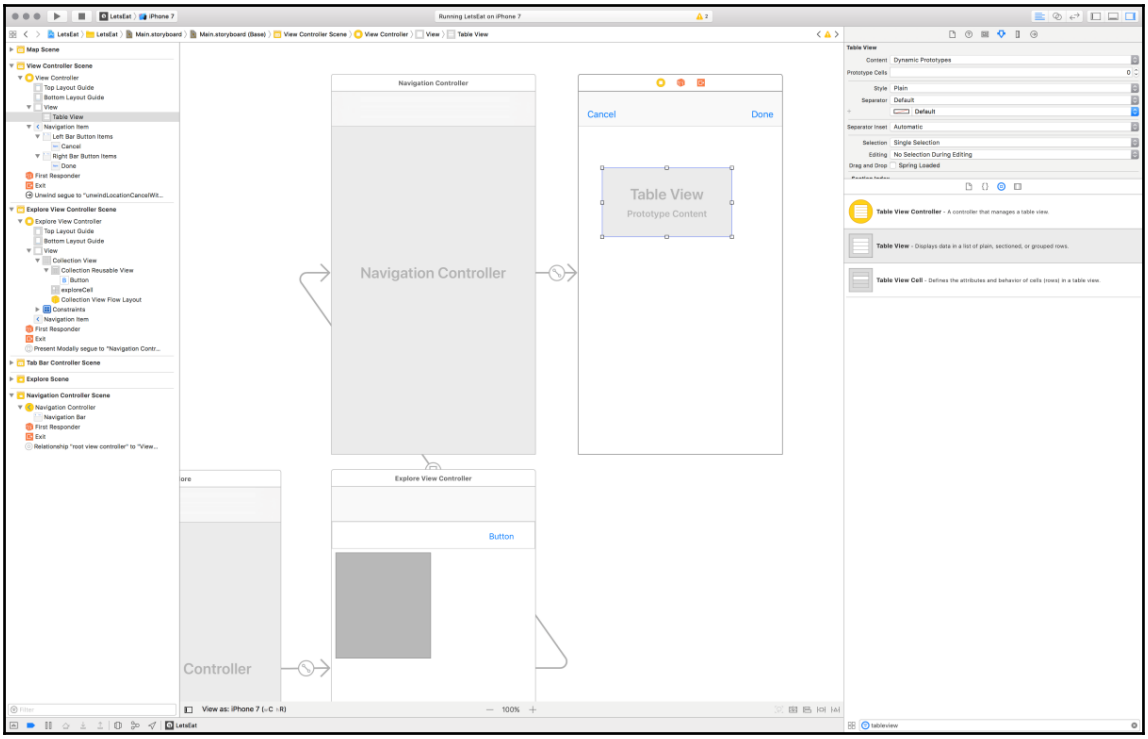

Navigation Controller
Item
Item
Navigation Controller
Not Applicable
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.
Navigation Bar - Provides a mechanism for displaying a navigation bar just below the status bar.
Toolbar - Provides a mechanism for displaying a toolbar at the bottom of the screen.
Item
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.
Tab Bar - Provides a mechanism for displaying a tab bar at the bottom of the screen.
Tab Bar Item - Represents an item on a UITabBar object.
Search Bar - Displays an editable search bar, containing the search icon, that sends an action message to a target object when Return is tapped.
Search Bar and Search Display Controller - Displays an editable search bar connected to a search display controller for managing searching.


Item
Item

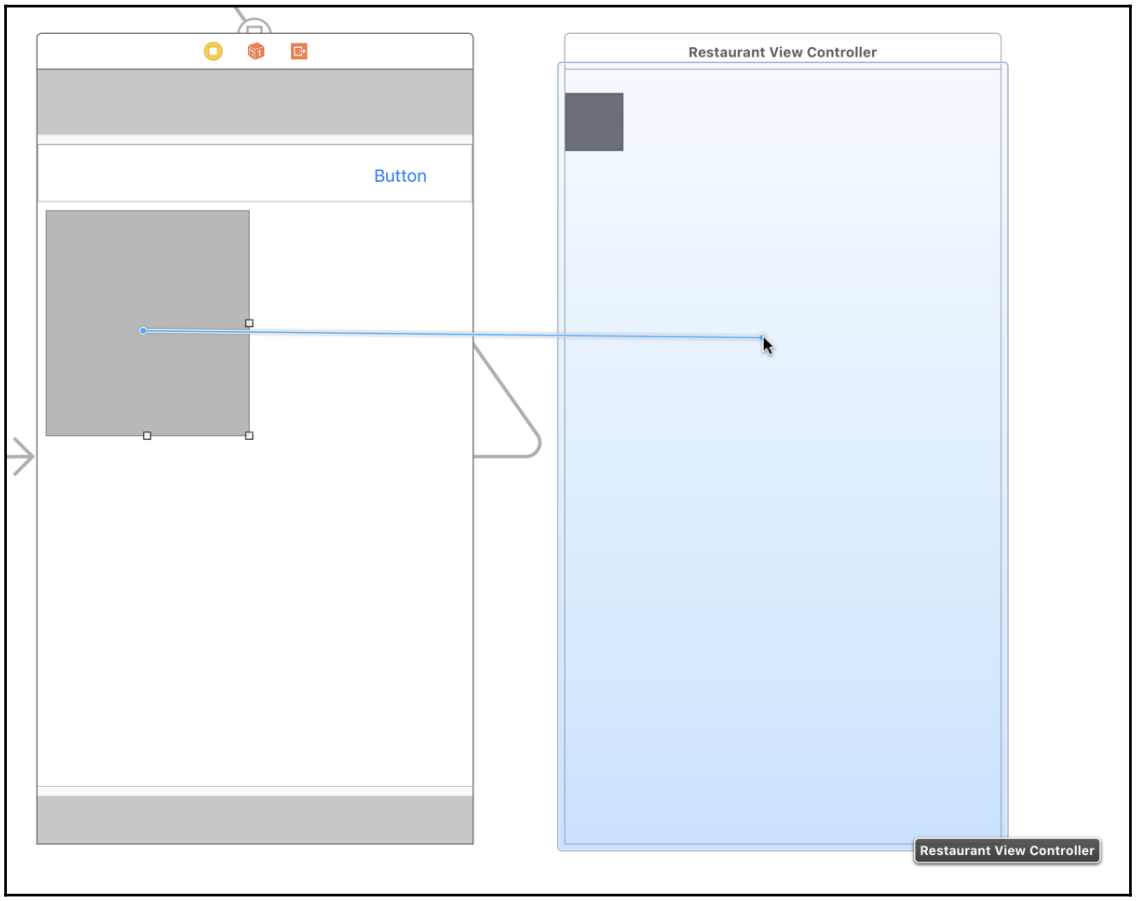

Cancel
Done

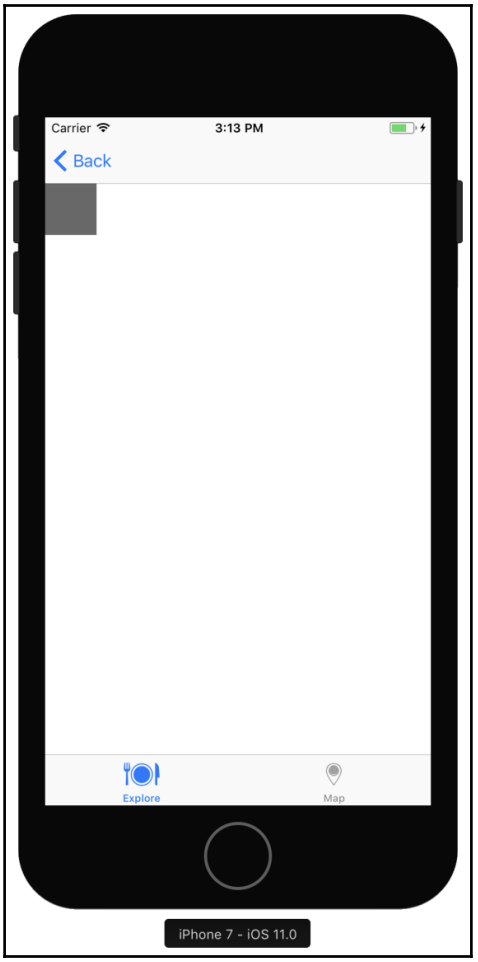

View Controller
Cancel
Done

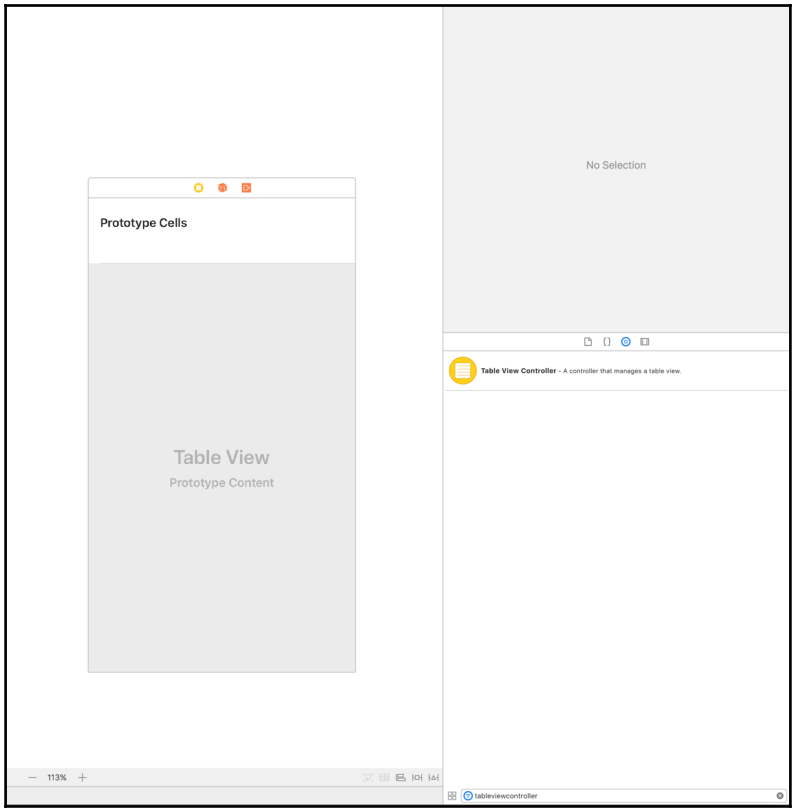

Action Segue
unwindLocationCancelWithSegue:

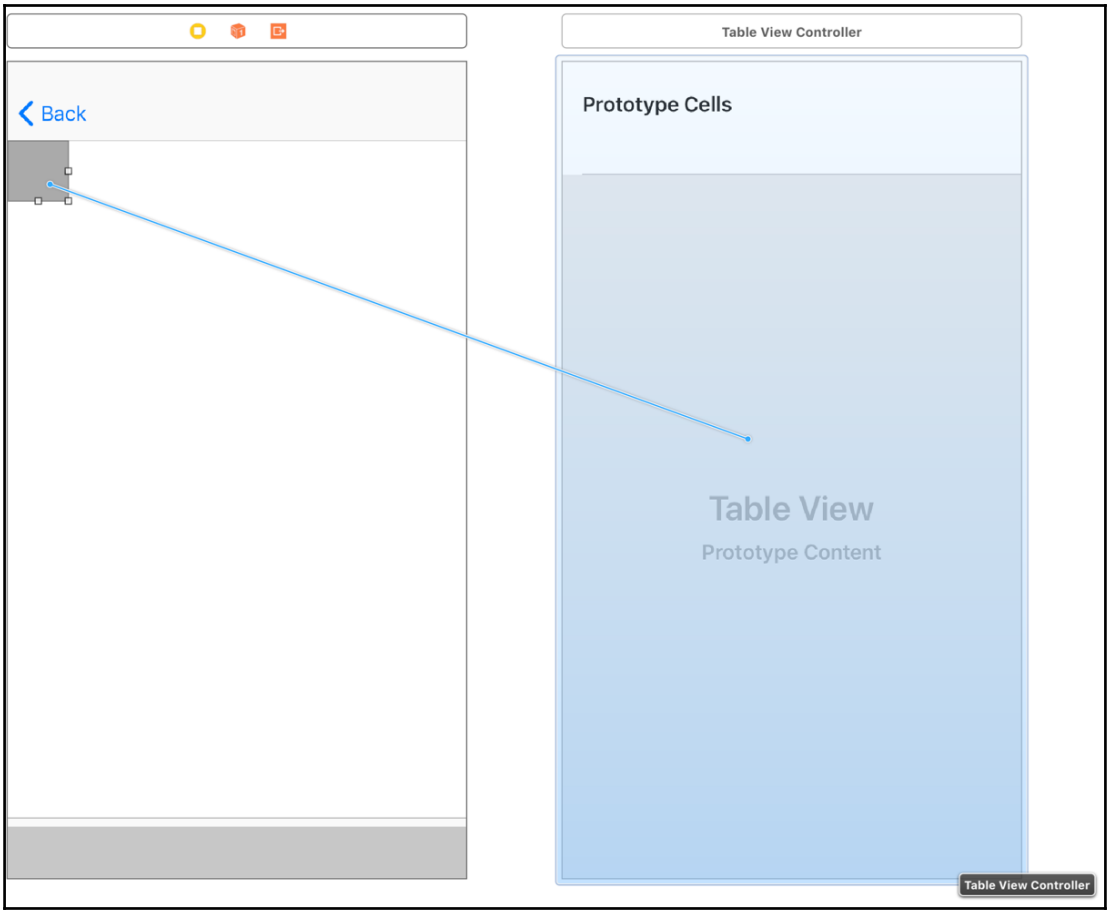

Navigation Controller
Cancel
Done
Table View
Prototype Content
Navigation Controller
Explore View Controller
Button
Controller

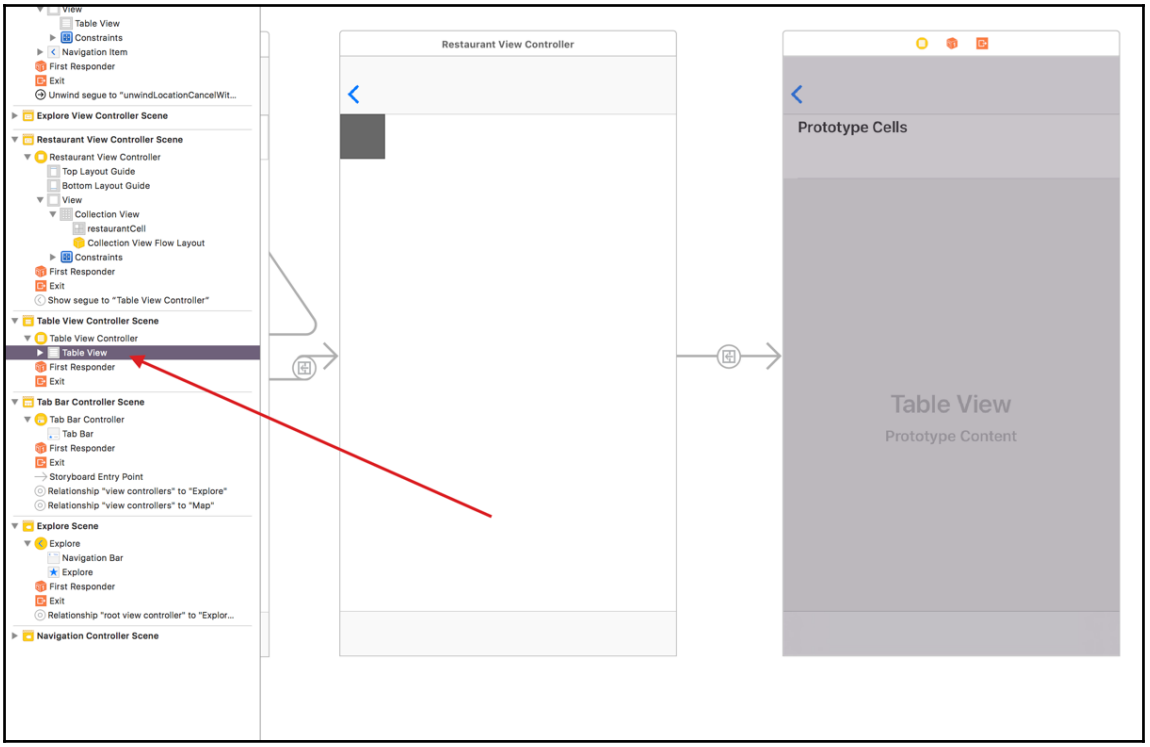
Chapter 9: Finishing Up Our App Structure in Storyboard




Carrier
3:13 PM
Back
Explore
Map
iPhone 7 - iOS 11.0


Prototype Cells
Table View
Prototype Content
No Selection
Table View Controller - A controller that manages a table view.
113%
tableviewcontroller


Back
Table View Controller
Prototype Cells
Table View
Prototype Content
Table View Controller


View
Table View
Constraints
Navigation Item
First Responder
Exit
Unwind segue to "unwindLocationCancelWit...
Explore View Controller Scene
Restaurant View Controller Scene
Restaurant View Controller
Top Layout Guide
Bottom Layout Guide
View
Collection View
restaurantCell
Collection View Flow Layout
Constraints
First Responder
Exit
Show segue to "Table View Controller"
Table View Controller Scene
Table View Controller
Table View
First Responder
Exit
Tab Bar Controller Scene
Tab Bar Controller
Tab Bar
First Responder
Exit
Storyboard Entry Point
Relationship "view controllers" to "Explore"
Relationship "view controllers" to "Map"
Explore Scene
Explore
Navigation Bar
Explore
First Responder
Exit
Relationship "root view controller" to "Explor...
Navigation Controller Scene
Restaurant View Controller
Prototype Cells
Table View
Prototype Content

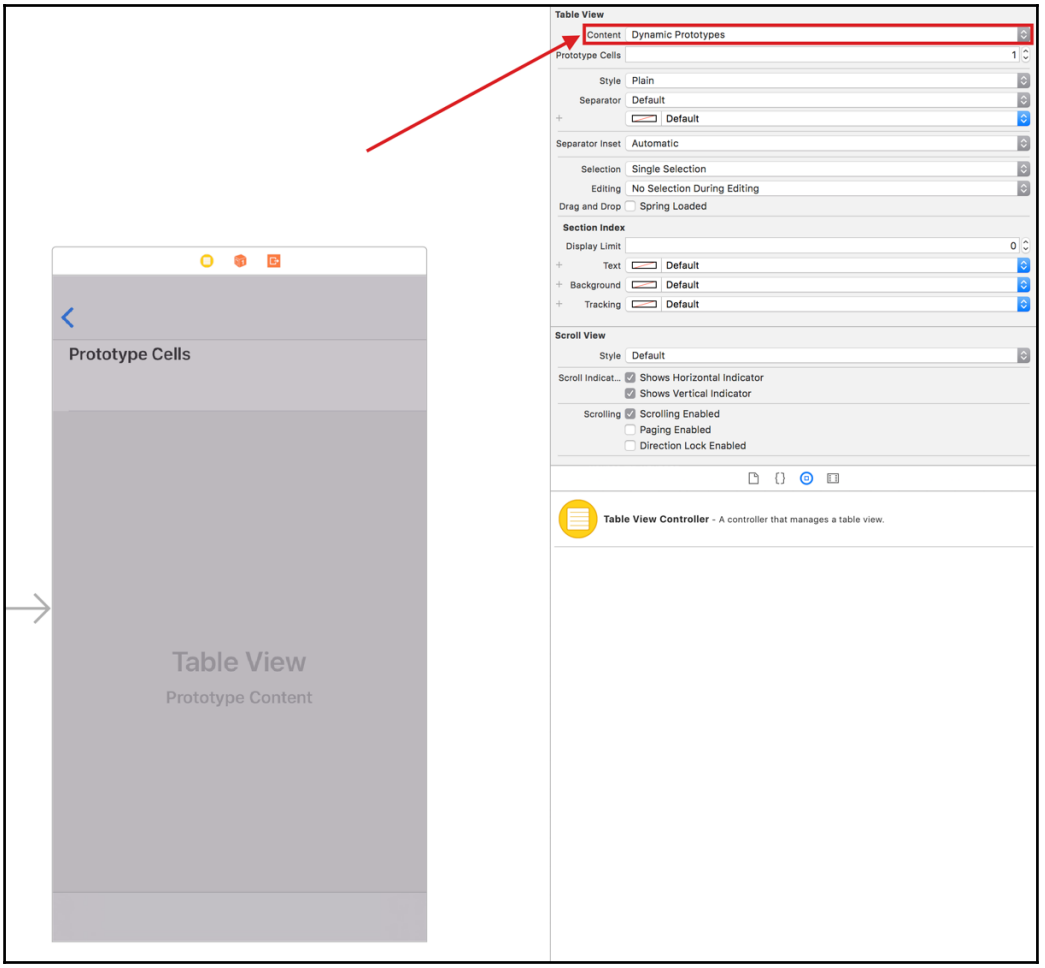
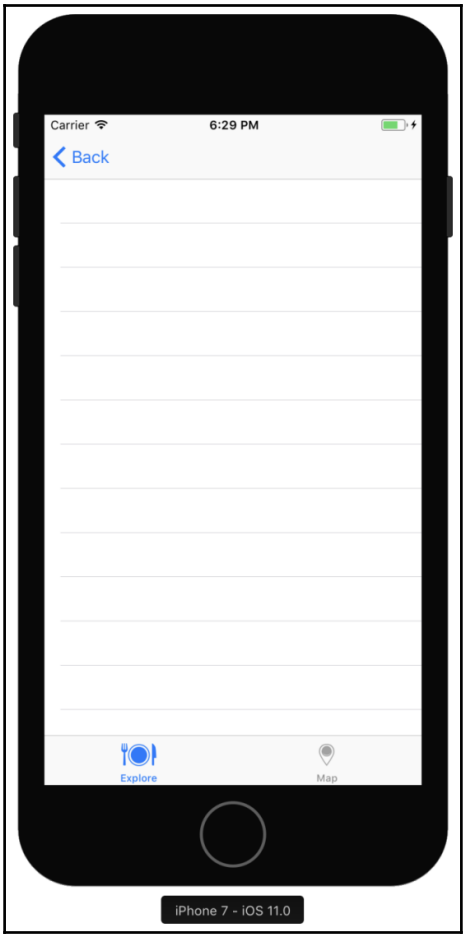


Table View
Content
Dynamic Prototypes
Prototype Cells
1
Style
Plain
Separator
Default
Default
Separator Inset
Automatic
Selection
Single Selection
Editing
No Selection During Editing
Drag and Drop
Spring Loaded
Section Index
Display Limit
0
Text
Default
Background
Default
Tracking
Default
Scroll View
Style
Default
Scroll Indicat...
Shows Horizontal Indicator
Shows Vertical Indicator
Scrolling
Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Table View Controller - A controller that manages a table view.
Prototype Cells
Table View
Prototype Content


Carrier
6:29 PM
Back
Explore
Map
iPhone 7 - iOS 11.0

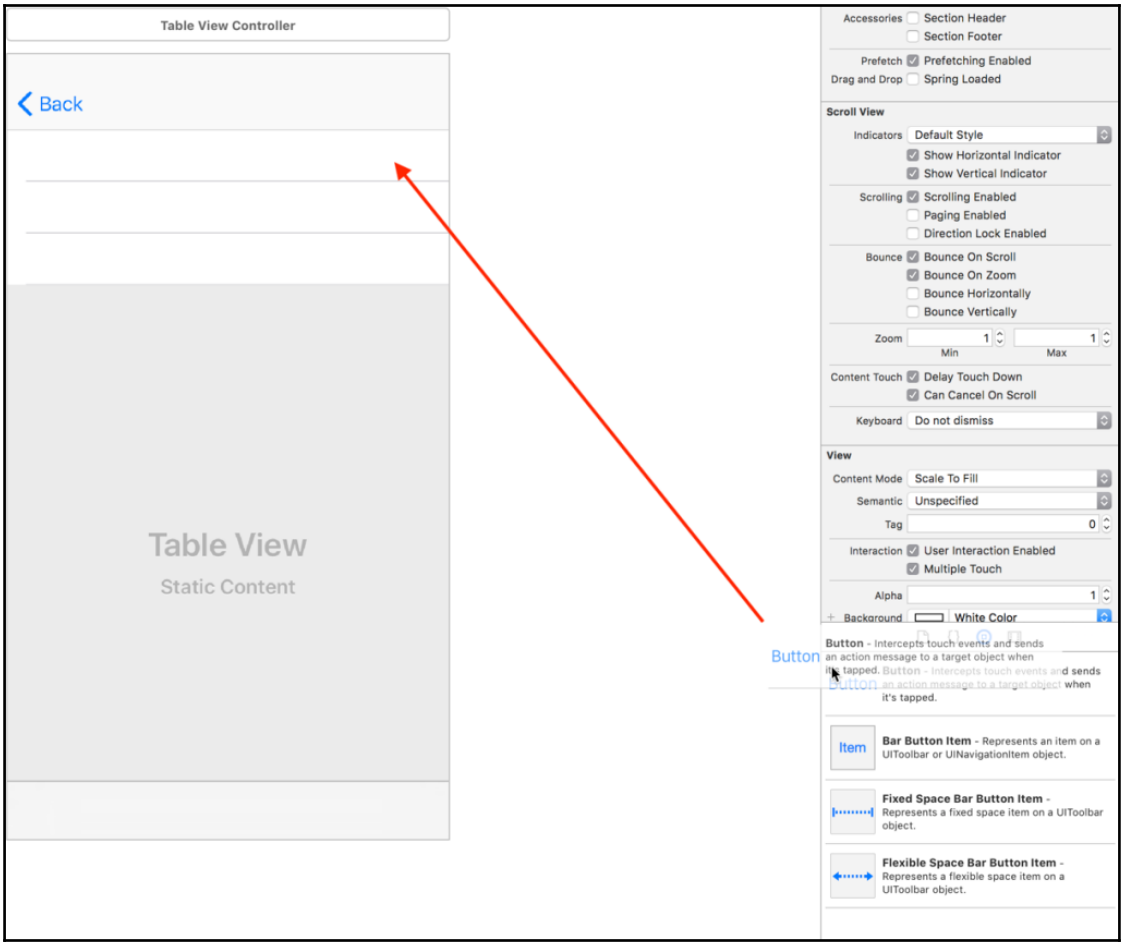


Table View Controller
Back
Table View
Static Content
Accessories
Section Header
Section Footer
Prefetch
Prefetching Enabled
Drag and Drop
Spring Loaded
Scroll View
Indicators
Default Style
Show Horizontal Indicator
Show Vertical Indicator
Scrolling
Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Bounce
Bounce On Scroll
Bounce On Zoom
Bounce Horizontally
Bounce Vertically
Zoom
Min
Max
Content Touch
Delay Touch Down
Can Cancel On Scroll
Keyboard
Do not dismiss
View
Content Mode
Scale To Fill
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
White Color
Button
Button - Intercepts touch events and sends an action message to a target object when it's tapped.
Item
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.
Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.
Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.

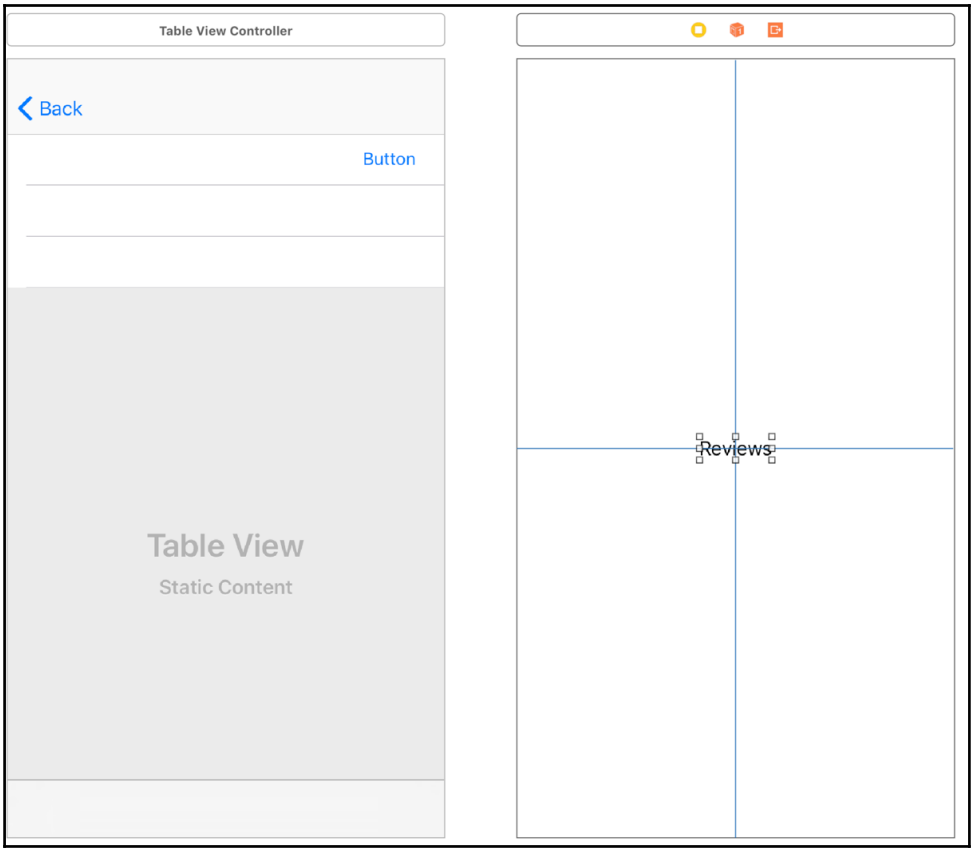
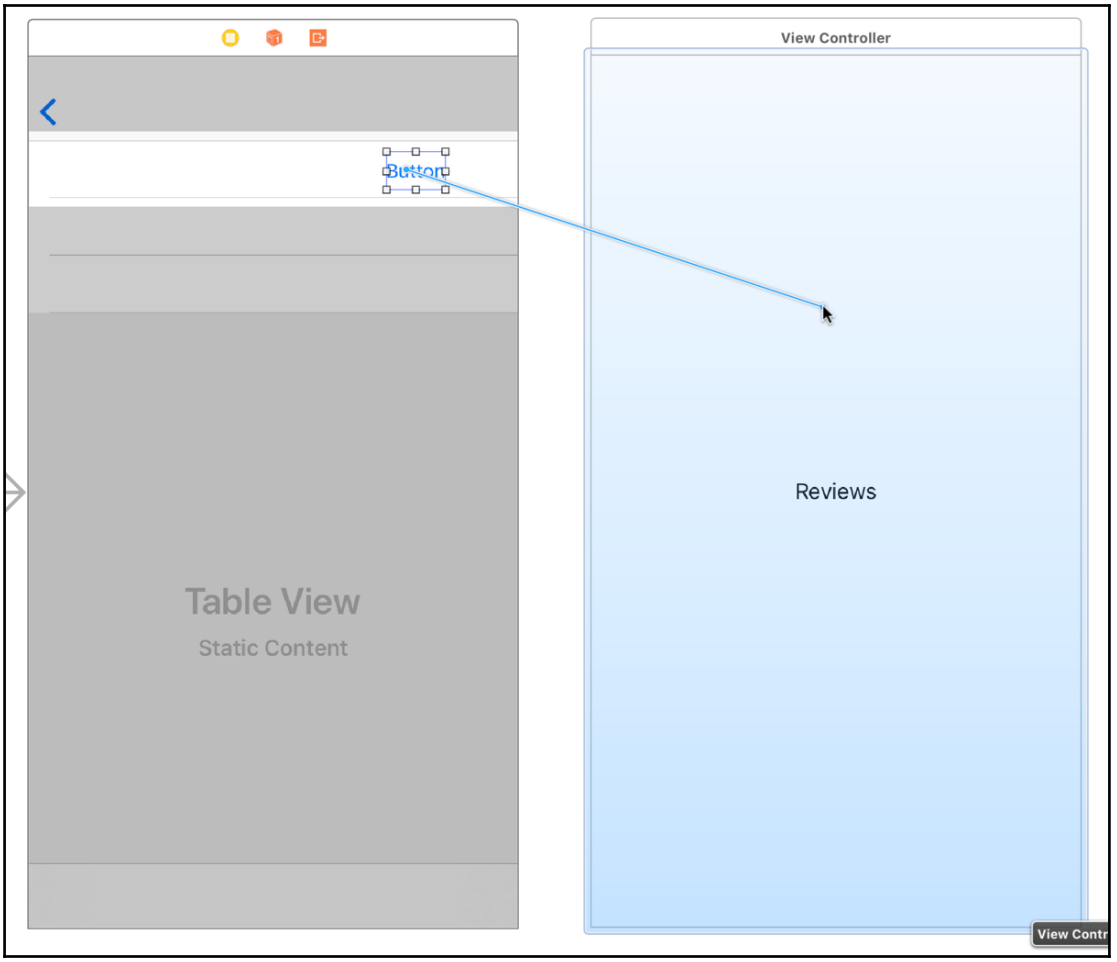
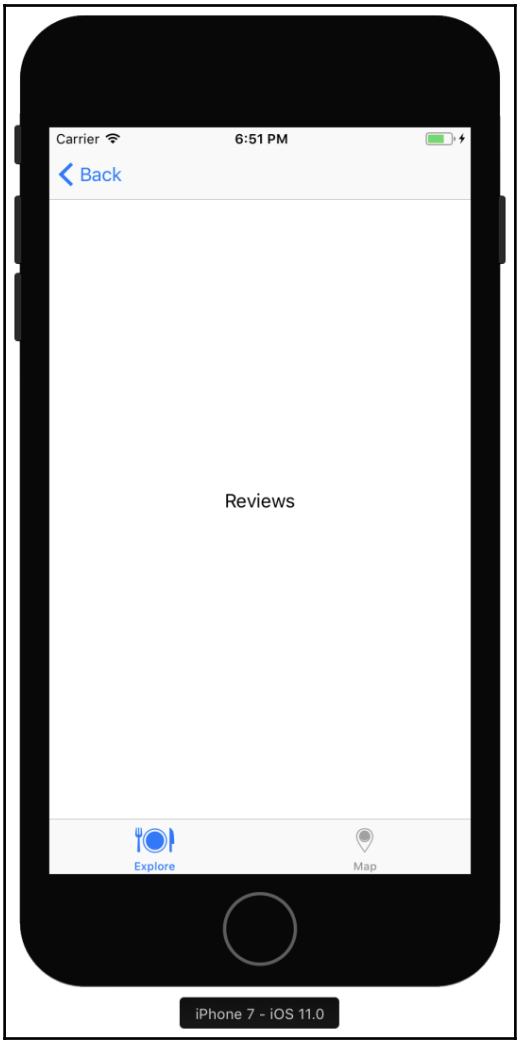
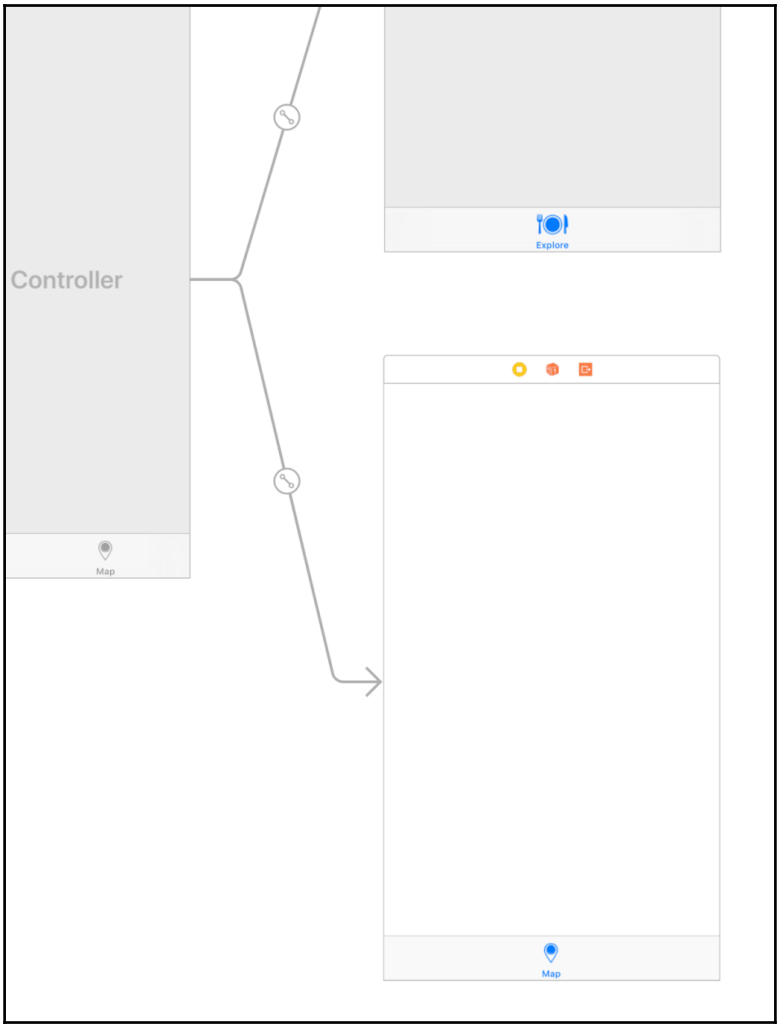
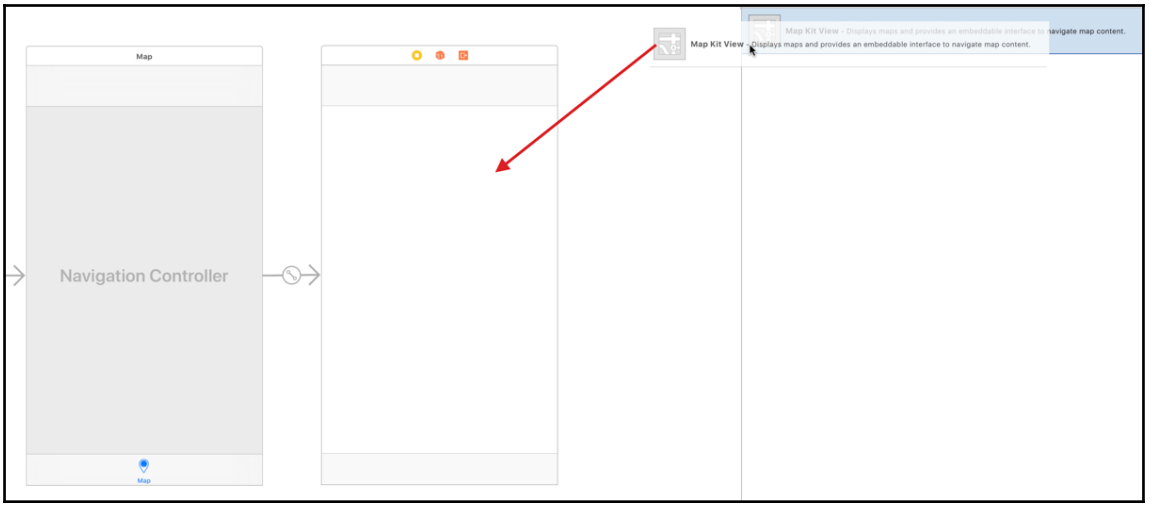


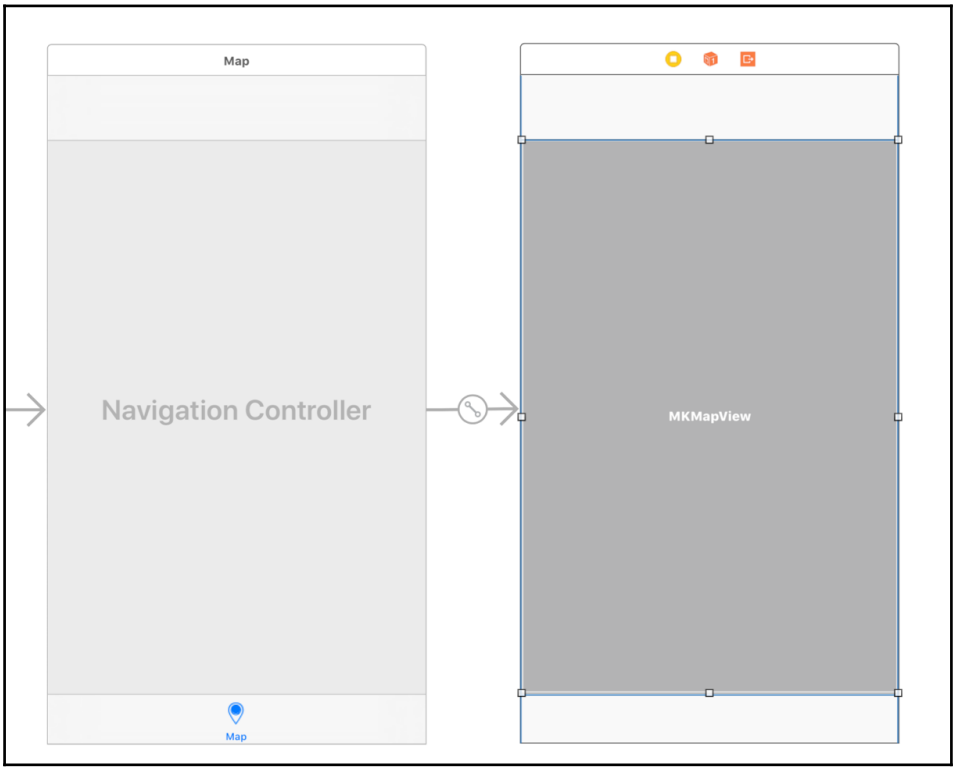
Table View Controller
Back
Button
Table View
Static Content
Reviews

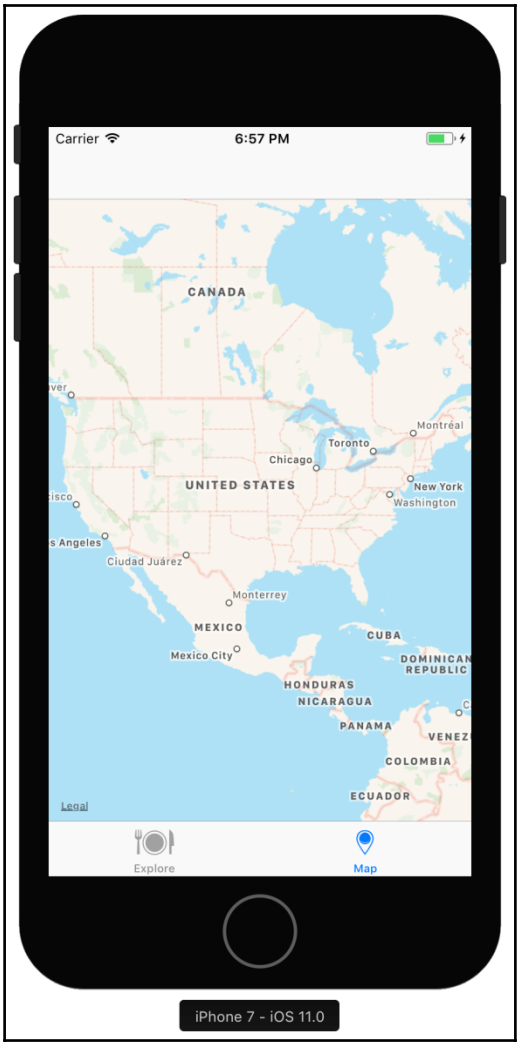

View Controller
Button
Reviews
Table View
Static Content
View Contr

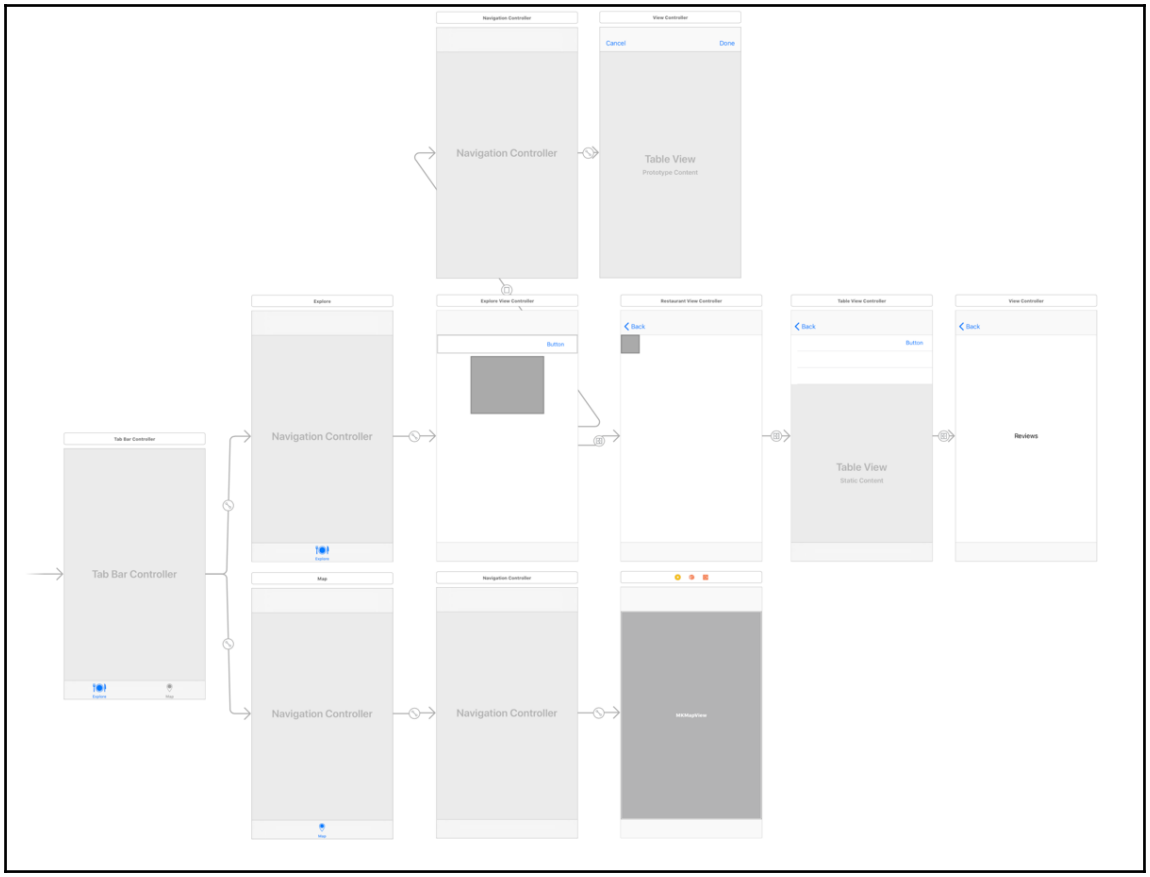

Carrier
6:51 PM
Back
Reviews
Explore
Map
iPhone 7 - iOS 11.0


Controller
Map
Explore
Map

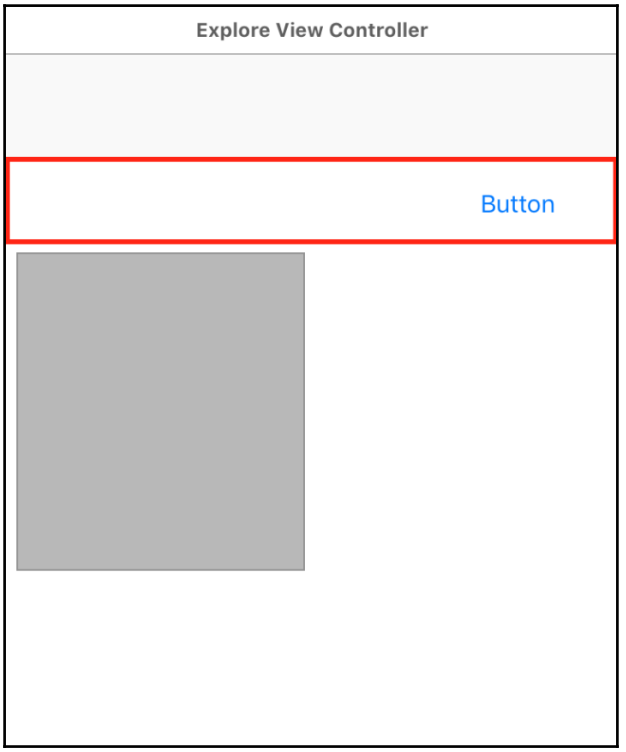

Map
Navigation Controller
Map
Map Kit View - Displays maps and provides an embeddable interface to navigate map content.

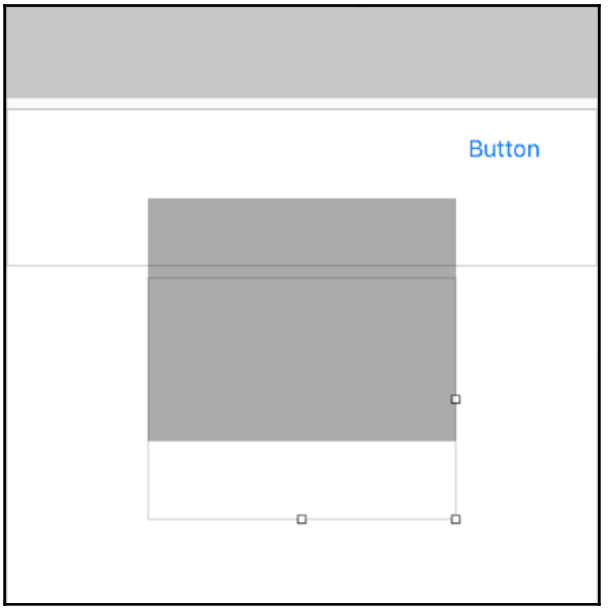

Map
Navigation Controller
Map
MKMapView

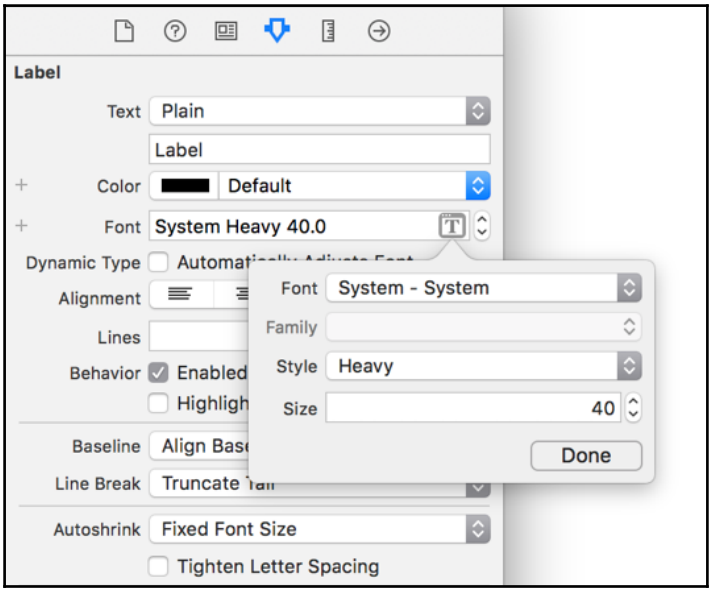

Carrier
6:57 PM
CANADA
Montréal
Toronto
Chicago
UNITED STATES
New York
Washington
Ciudad Juárez
Monterrey
MEXICO
CUBA
Mexico City
HONDURAS
NICARAGUA
PANAMA
COLOMBIA
ECUADOR
Legal
Explore
Map
iPhone 7 - iOS 11.0

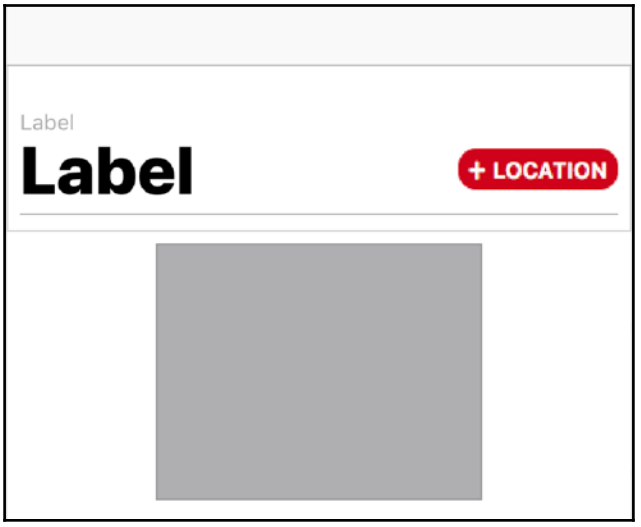

Navigation Controller
Navigation Controller
Cancel
Done
Table View
Prototype Content
Explore
Explore View Controller
Restaurant View Controller
Table View Controller
View Controller
Navigation Controller
Button
Back
Button
Back
Table View
Static Content
Back
Reviews
Tab Bar Controller
Tab Bar Controller
Map
Navigation Controller
Navigation Controller
MKMapView

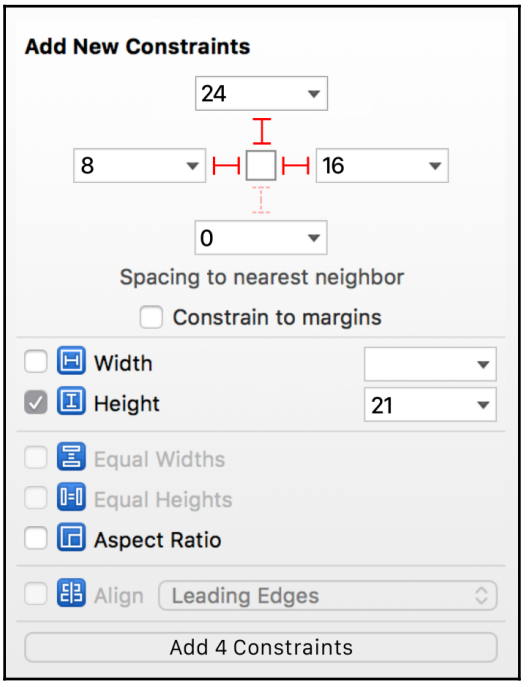
Chapter 10: Designing Cells



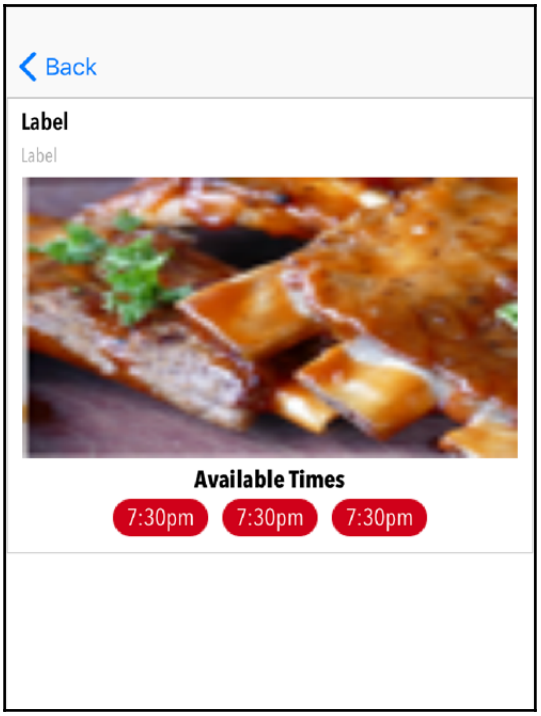

Button

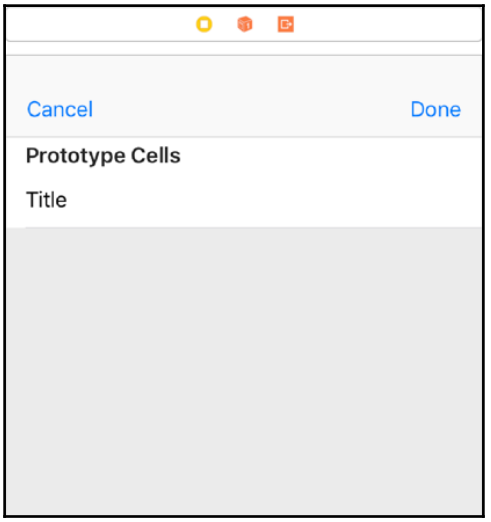

Label
Text
Plain
Label
Color
Default
Font
System Heavy 40.0
Dynamic Type
Alignment
Lines
Behavior
Enabled
Highligh
Baseline
Line Break
Autoshrink
Fixed Font Size
Tighten Letter Spacing
Font
System - System
Family
Style
Heavy
Size
40
Done


Label
Label
+ LOCATION

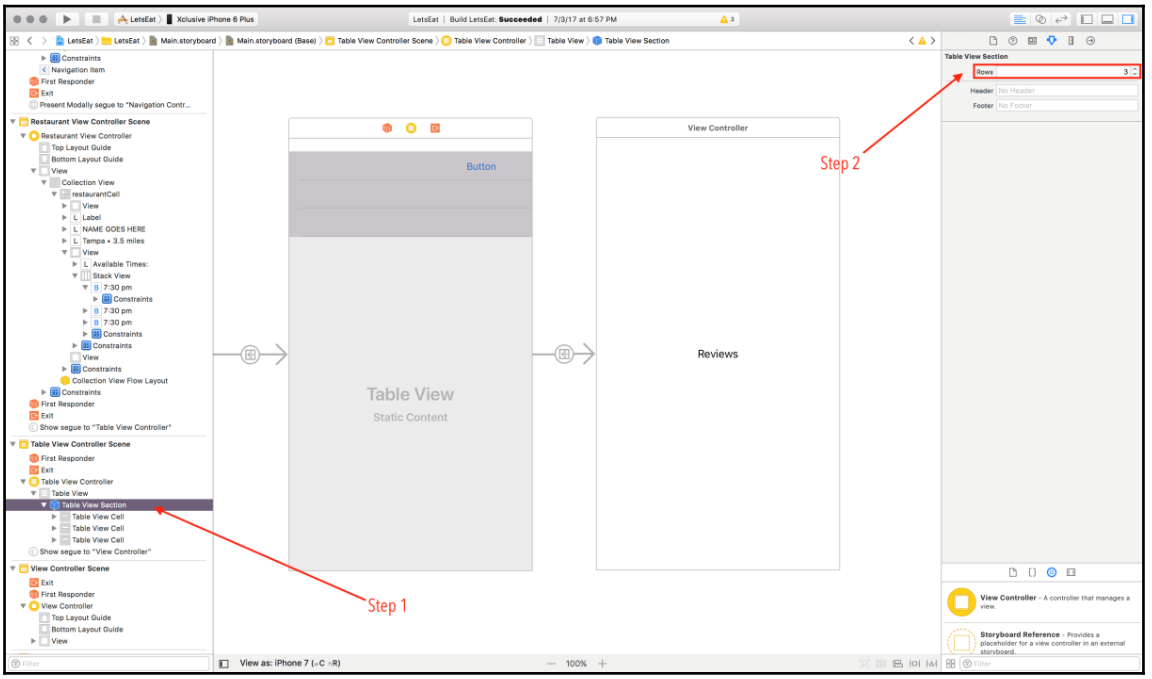

Add New Constraints
24
8
16
0
Spacing to nearest neighbor
Constrain to margins
Width
Height
21
Equal Widths
Equal Heights
Aspect Ratio
Align
Leading Edges
Add 4 Constraints

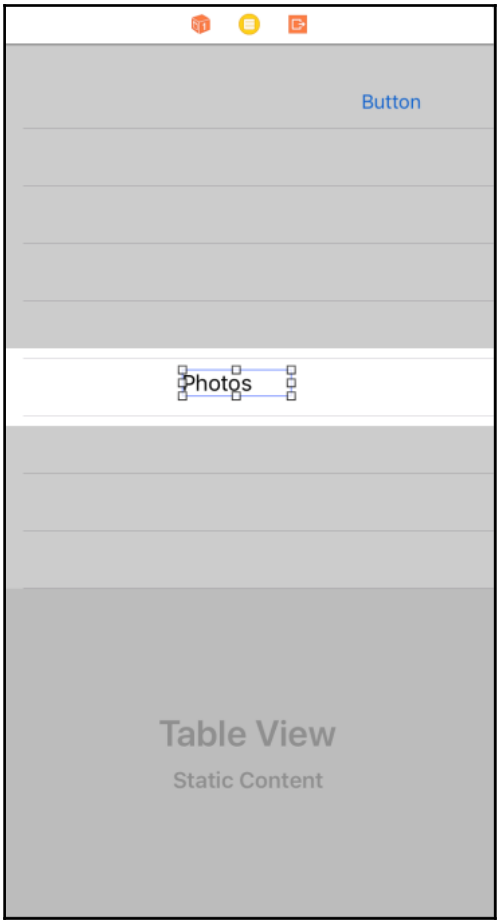

UIImageView
Label

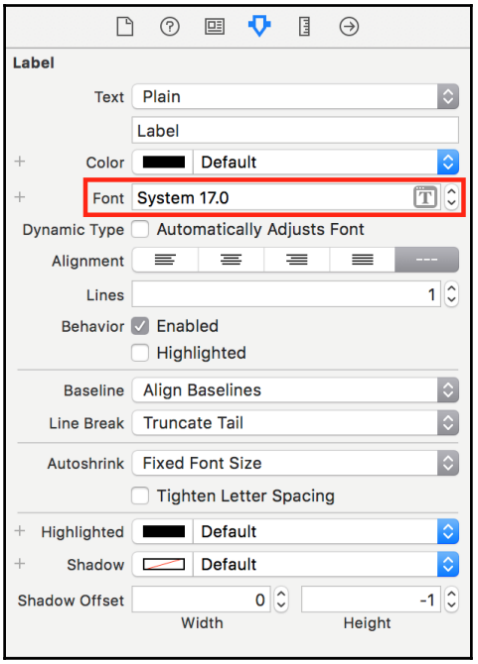

Back
Label
Label
Available Times
7:30pm
7:30pm
7:30pm

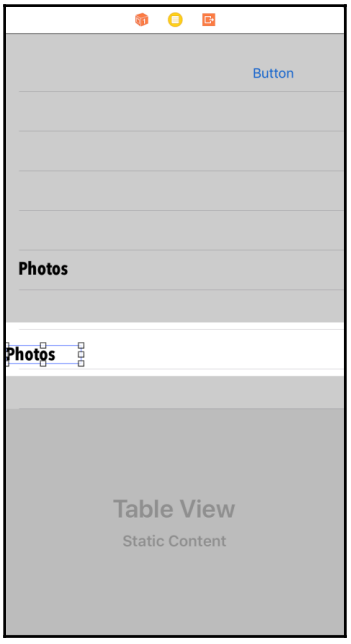

Cancel
Done
Prototype Cells
Title

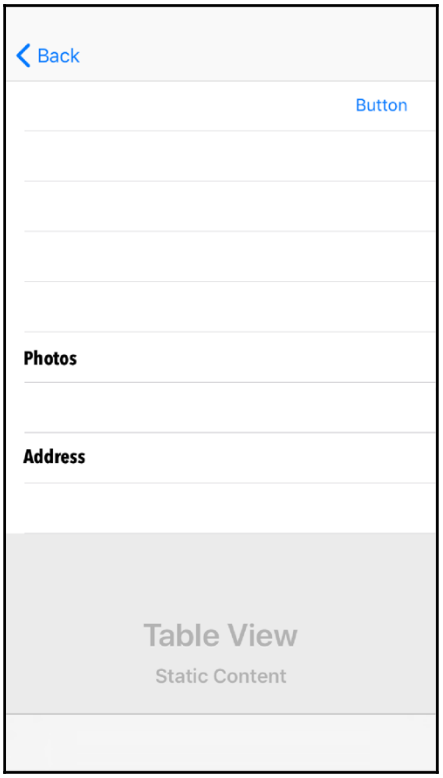
Chapter 11: Designing Static Tables



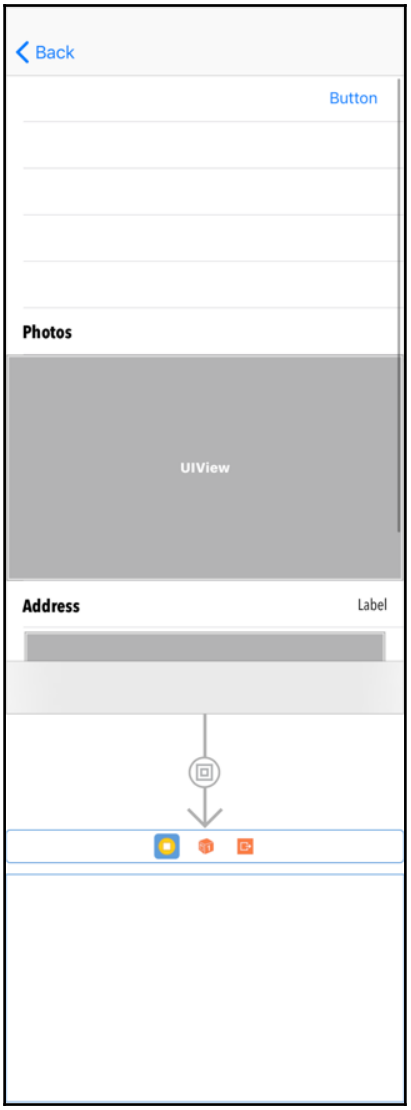

Button
Photos
Table View
Static Content

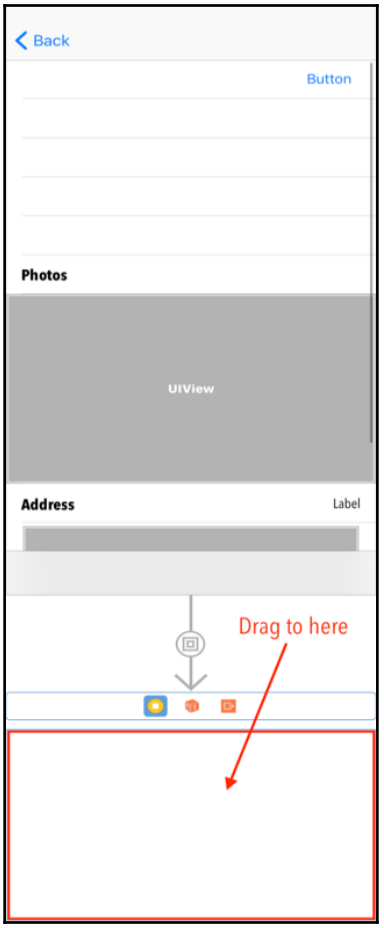

Label
Text
Plain
Label
Color
Default
Font
System 17.0
Dynamic Type
Automatically Adjusts Font
Alignment
Lines
1
Behavior
Enabled
Highlighted
Baseline
Align Baselines
Line Break
Truncate Tail
Autoshrink
Fixed Font Size
Tighten Letter Spacing
Highlighted
Default
Shadow
Default
Shadow Offset
0
-1
Width
Height

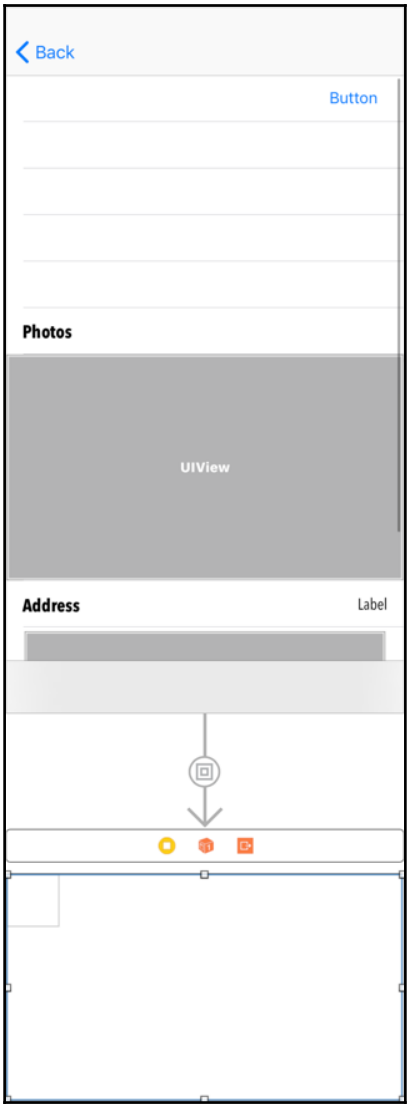

Button
Photos
Photos
Table View
Static Content

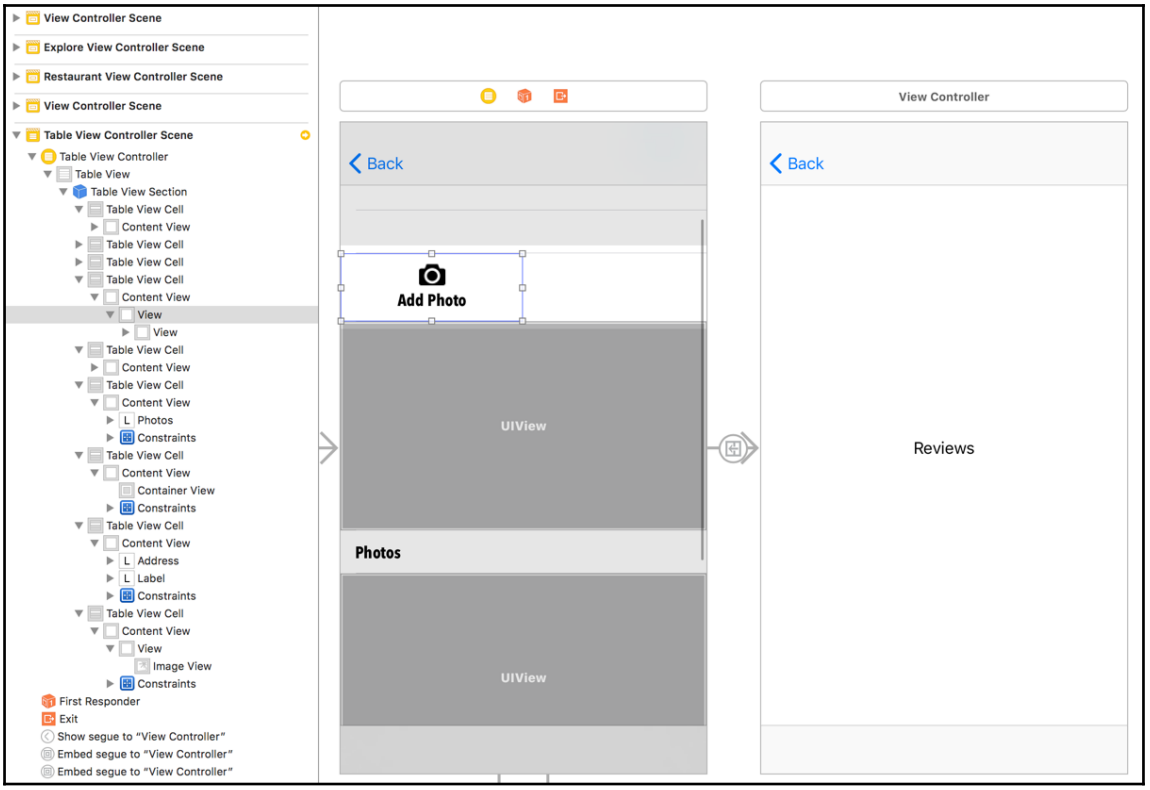

Back
Button
Photos
Address
Table View
Static Content

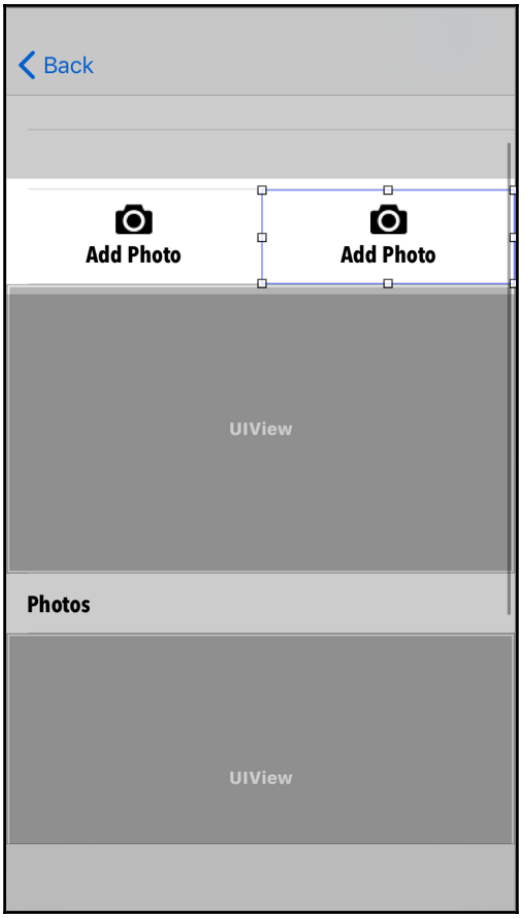

Back
Button
Photos
Address
Label
UIImageView

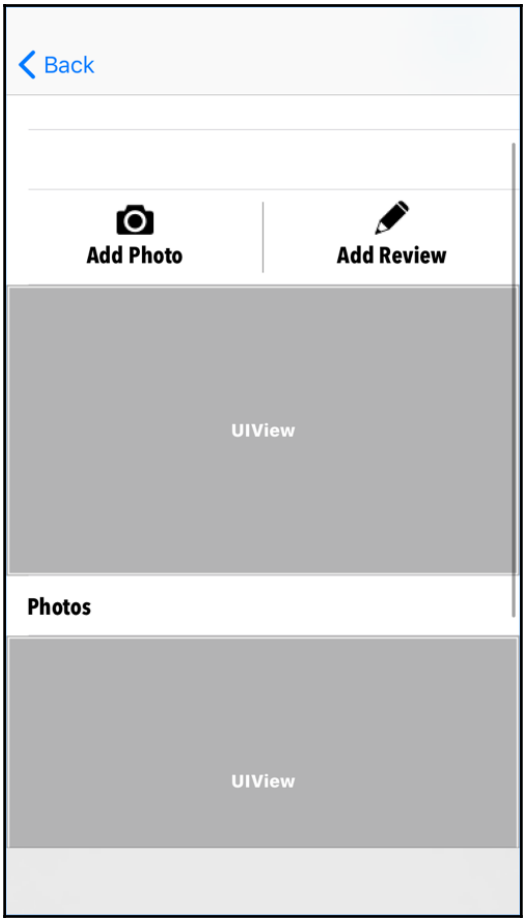

Back
Button
Photos
UIView
Address
Label

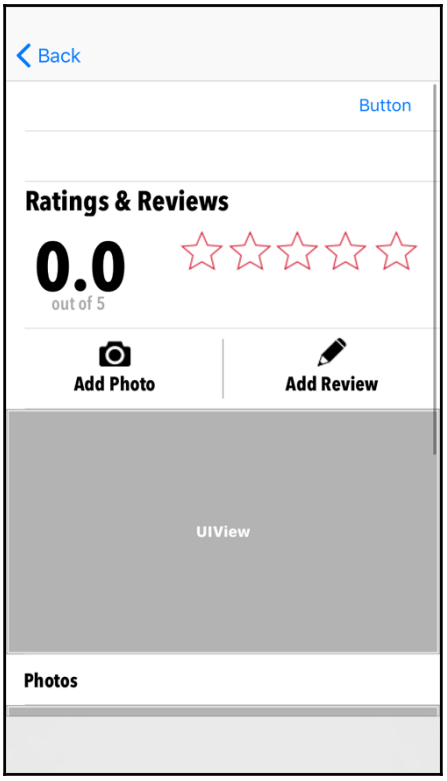

Back
Button
Photos
UIView
Address
Label
Drag to here

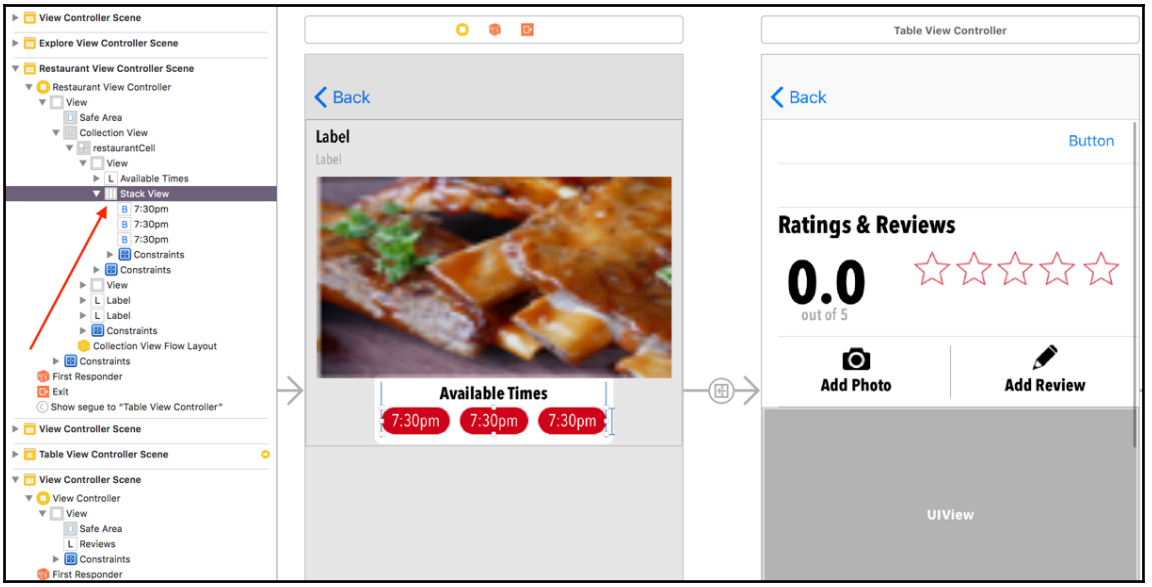

Back
Button
Photos
UIView
Address
Label

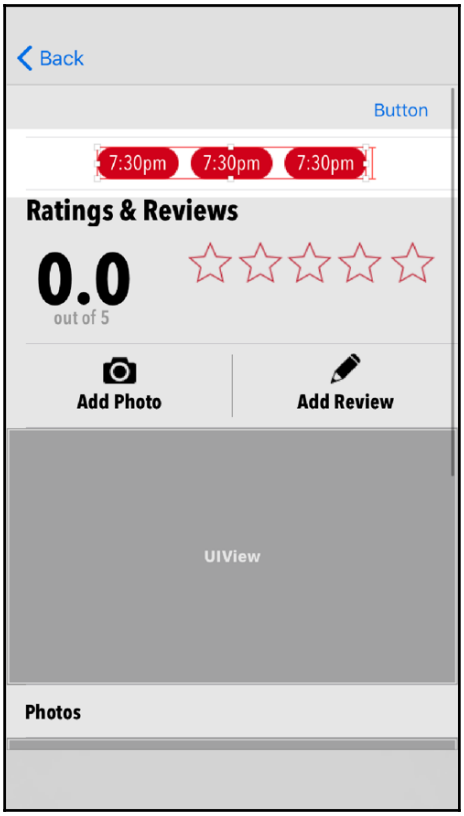

View Controller Scene
Explore View Controller Scene
Restaurant View Controller Scene
View Controller Scene
Table View Controller Scene
Table View Controller
Table View
Table View Section
Table View Cell
Content View
Table View Cell
Table View Cell
Table View Cell
Content View
View
View
Table View Cell
Content View
Table View Cell
Content View
Photos
Constraints
Table View Cell
Content View
Container View
Constraints
Table View Cell
Content View
Address
Label
Constraints
Table View Cell
Content View
View
Image View
Constraints
First Responder
Exit
Show segue to "View Controller"
Embed segue to "View Controller"
Embed segue to "View Controller"
Back
Add Photo
UIView
Photos
UIView
View Controller
Back
Reviews

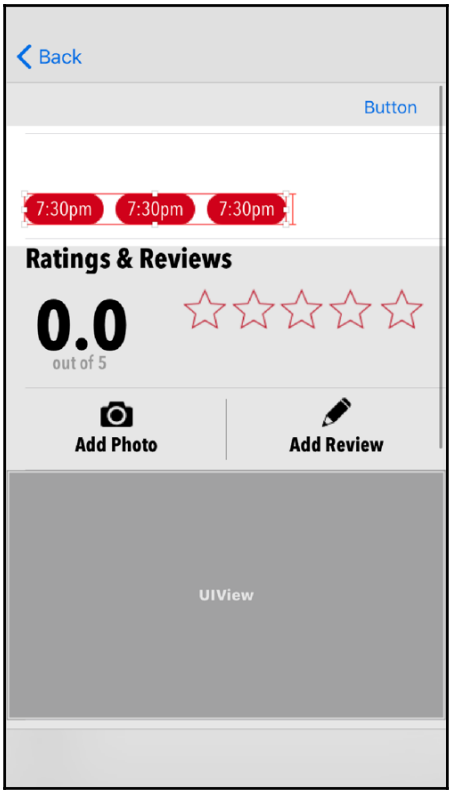

Back
Add Photo
Add Photo
UIView
Photos
UIView

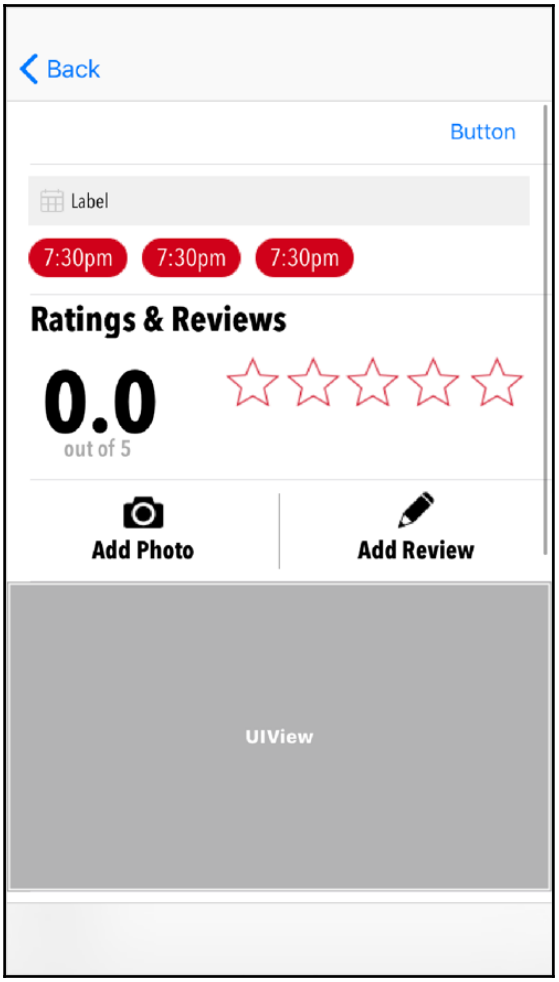

Back
Add Photo
Add Review
UIView
Photos
UIView

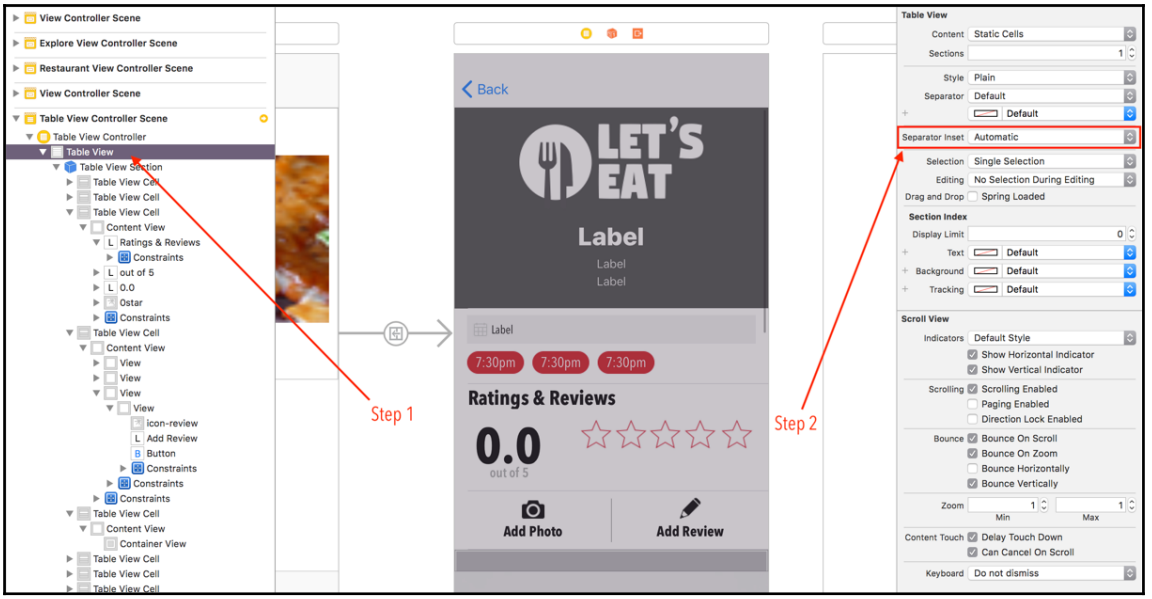

Back
Button
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
Photos


View Controller Scene
Explore View Controller Scene
Restaurant View Controller Scene
Restaurant View Controller
View
Safe Area
Collection View
restaurantCell
View
Available Times
Stack View
7:30pm
7:30pm
7:30pm
Constraints
Constraints
View
Label
Label
Constraints
Collection View Flow Layout
Constraints
First Responder
Exit
Show segue to "Table View Controller"
View Controller Scene
Table View Controller Scene
View Controller Scene
View Controller
View
Safe Area
Reviews
Constraints
First Responder
Back
Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Table View Controller
Back
Button
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView


Back
Button
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
Photos


Back
Button
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView


Back
Button
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView


View Controller Scene
Explore View Controller Scene
Restaurant View Controller Scene
View Controller Scene
Table View Controller Scene
Table View Controller
Table View
Table View Section
Table View Cell
Table View Cell
Table View Cell
Content View
Ratings & Reviews
Constraints
out of 5
0.0
Ostar
Constraints
Table View Cell
Content View
View
View
View
View
icon-review
Add Review
Button
Constraints
Constraints
Constraints
Table View Cell
Content View
Container View
Table View Cell
Table View Cell
Table View Cell
Step 1
Back
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
Step 2
Table View
Content Static Cells
Sections 1
Style Plain
Separator Default
Default
Separator Inset Automatic
Selection Single Selection
Editing No Selection During Editing
Drag and Drop Spring Loaded
Section Index
Display Limit 0
Text Default
Background Default
Tracking Default
Scroll View
Indicators Default Style
Show Horizontal Indicator
Show Vertical Indicator
Scrolling Scrolling Enabled
Paging Enabled
Direction Lock Enabled
Bounce Bounce On Scroll
Bounce On Zoom
Bounce Horizontally
Bounce Vertically
Zoom 1 1
Min Max
Content Touch Delay Touch Down
Can Cancel On Scroll
Keyboard Do not dismiss

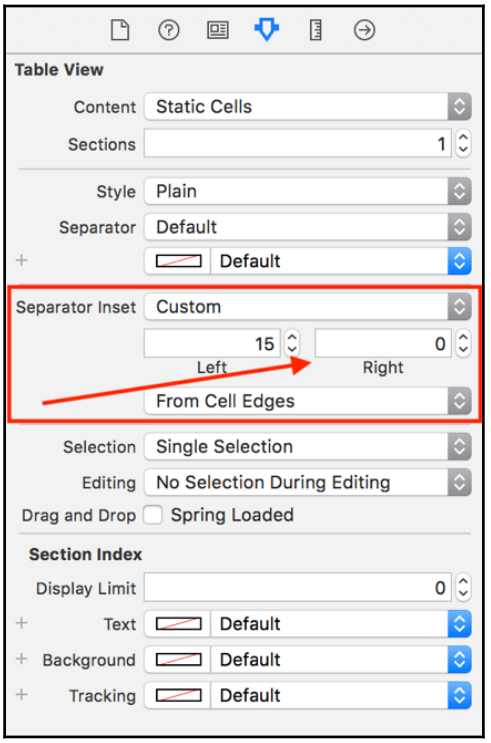
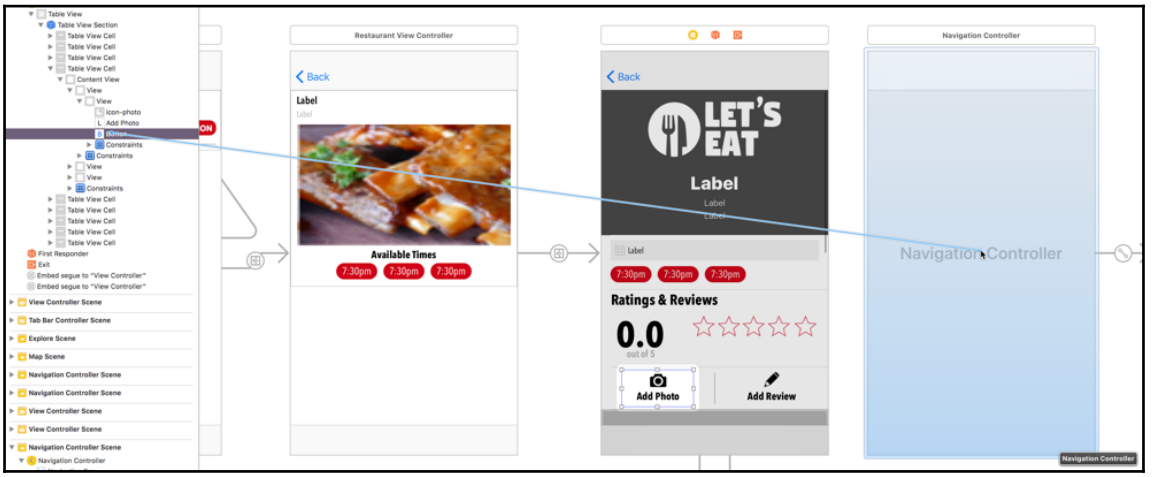
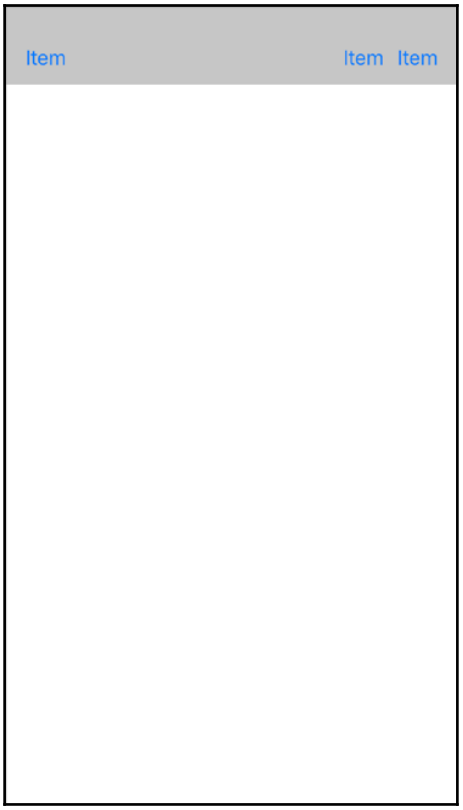
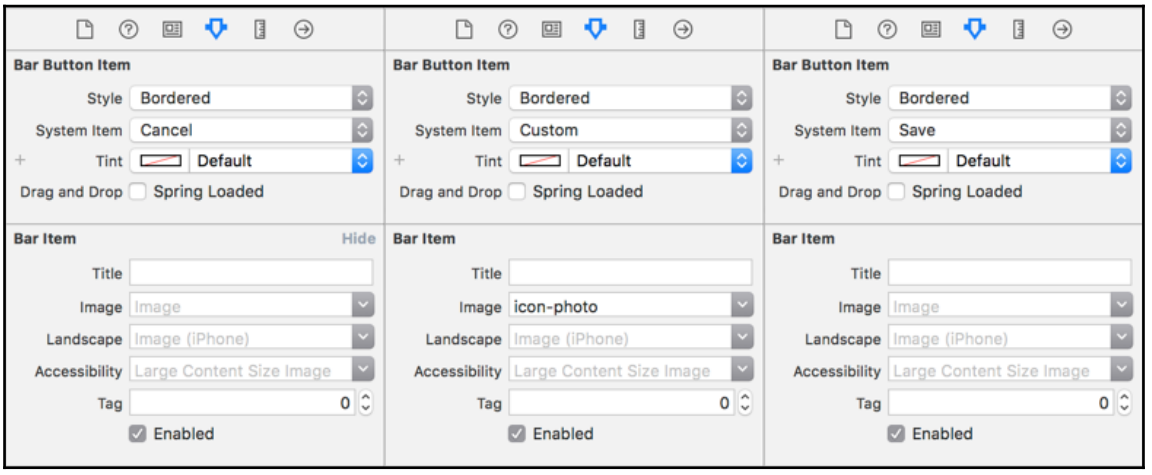


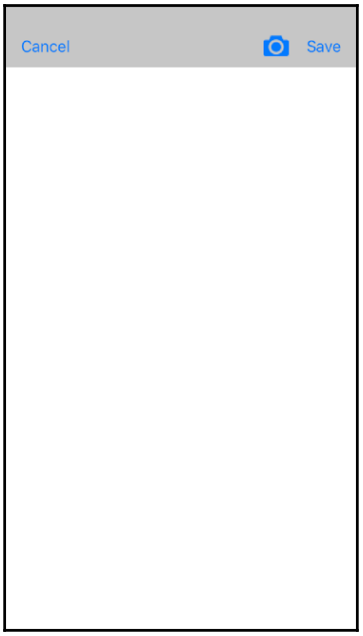
Table View
Content Static Cells
Sections 1
Style Plain
Separator Default
Default
Separator Inset Custom
15
0
Left
Right
From Cell Edges
Selection Single Selection
Editing No Selection During Editing
Drag and Drop Spring Loaded
Section Index
Display Limit 0
Text Default
Background Default
Tracking Default

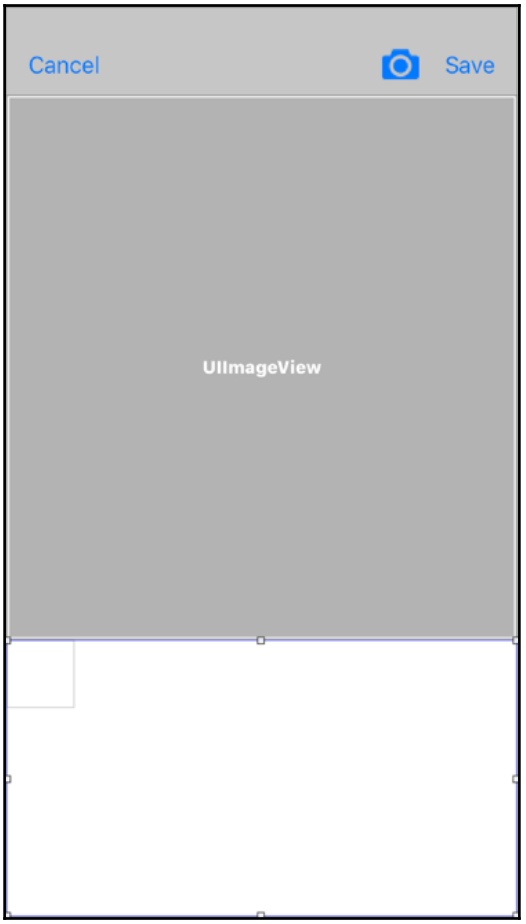
Chapter 12: Designing a Photo Filter and Review Form

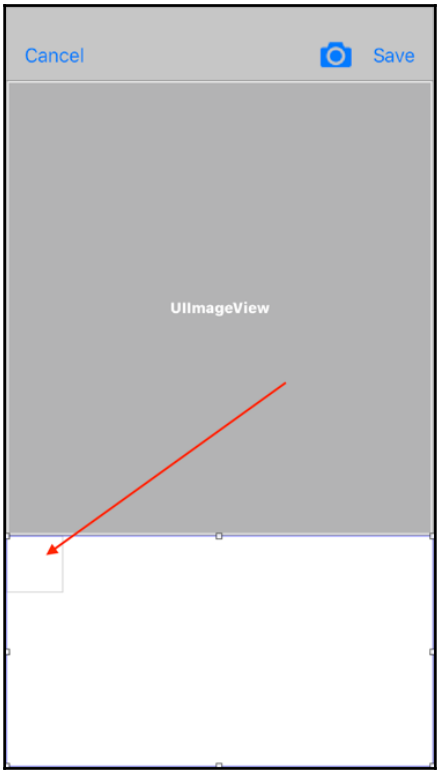


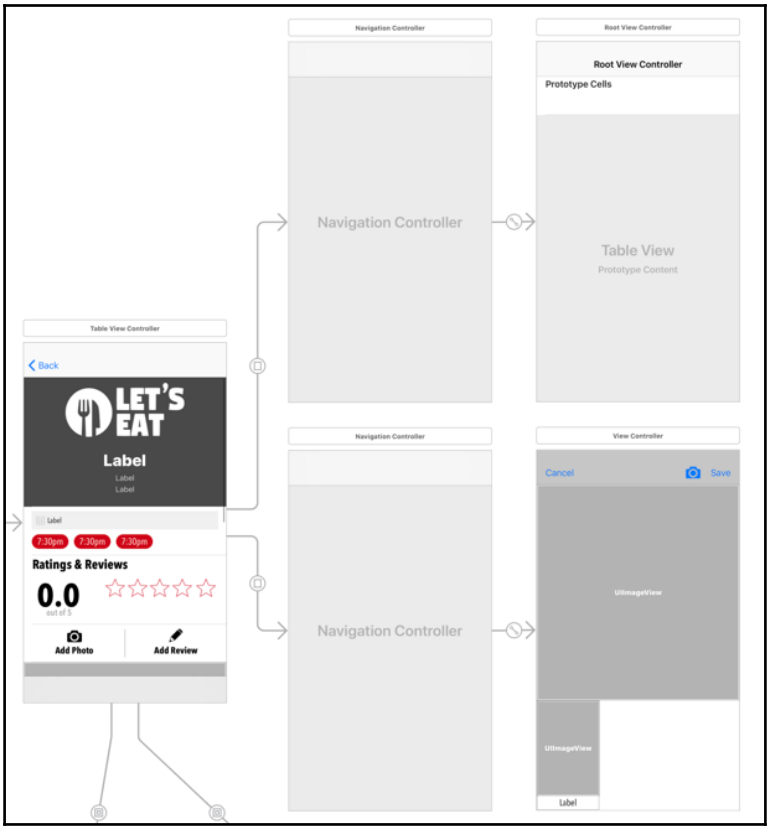

Item
Item
Item

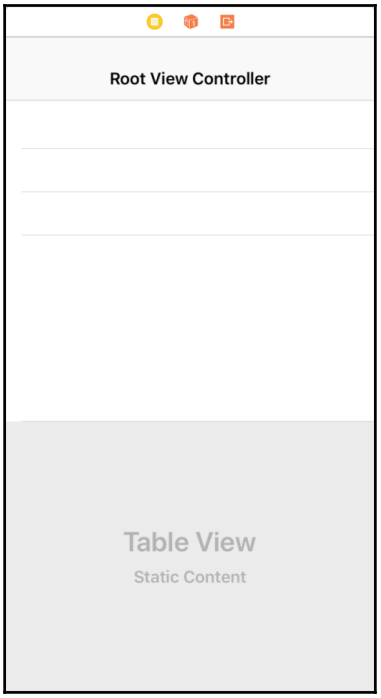

Bar Button Item
Style Bordered
System Item Cancel
Tint Default
Drag and Drop Spring Loaded
Bar Item
Hide
Title
Image Image
Landscape Image (iPhone)
Accessibility Large Content Size Image
Tag 0
Enabled
Bar Button Item
Style Bordered
System Item Custom
Tint Default
Drag and Drop Spring Loaded
Bar Item
Title
Image icon-photo
Landscape Image (iPhone)
Accessibility Large Content Size Image
Tag 0
Enabled
Bar Button Item
Style Bordered
System Item Save
Tint Default
Drag and Drop Spring Loaded
Bar Item
Title
Image Image
Landscape Image (iPhone)
Accessibility Large Content Size Image
Tag 0
Enabled

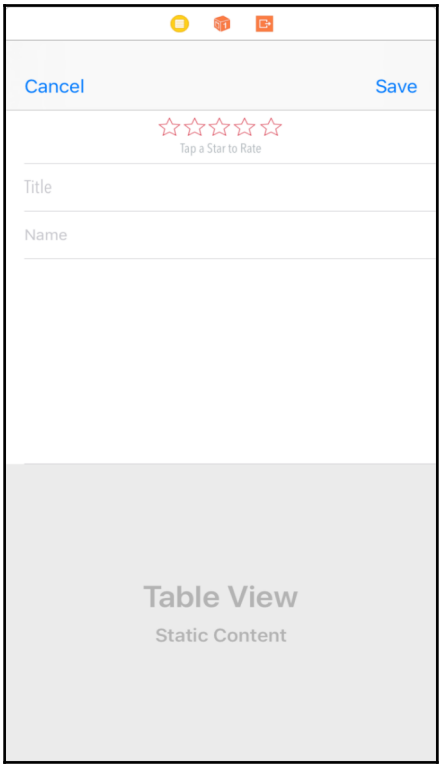

Cancel
Save

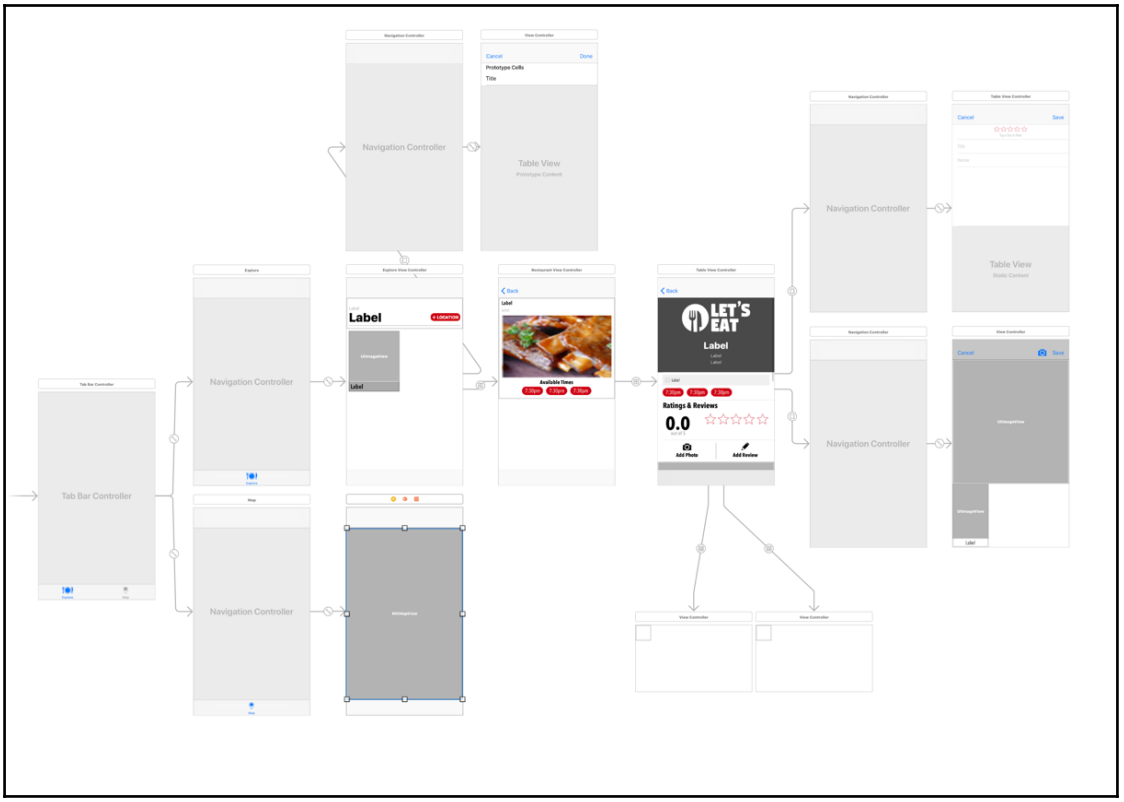

Cancel
Save
UIImageView

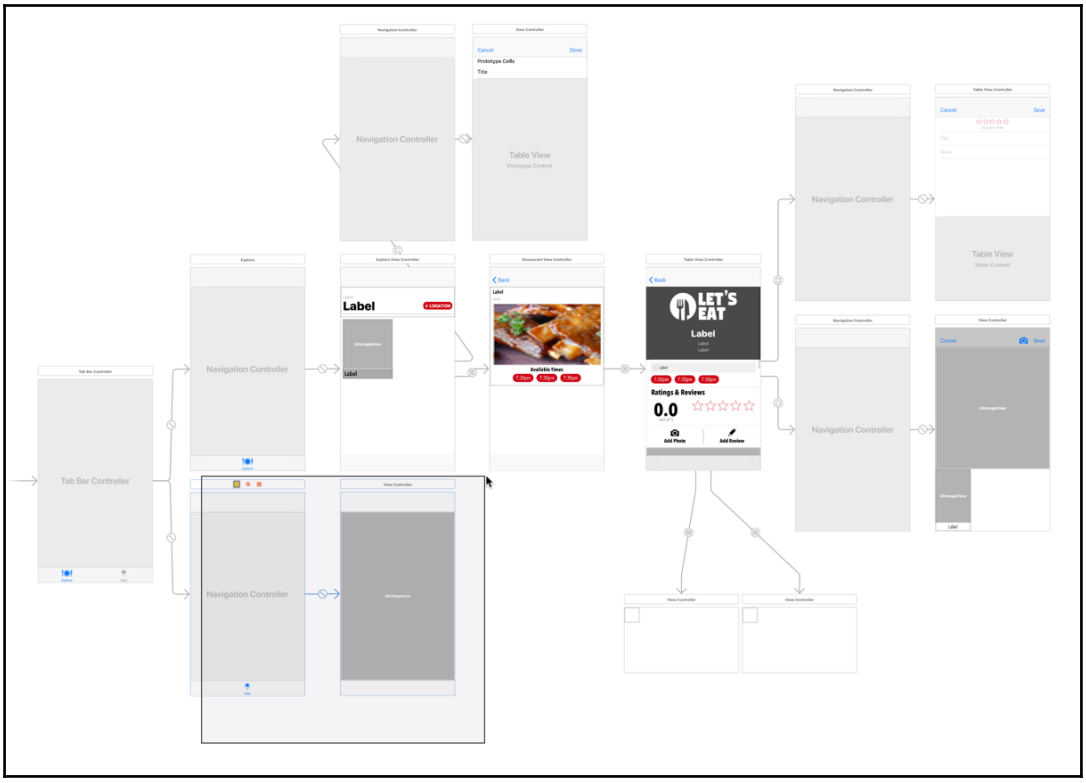

Cancel
Save
UIImageView

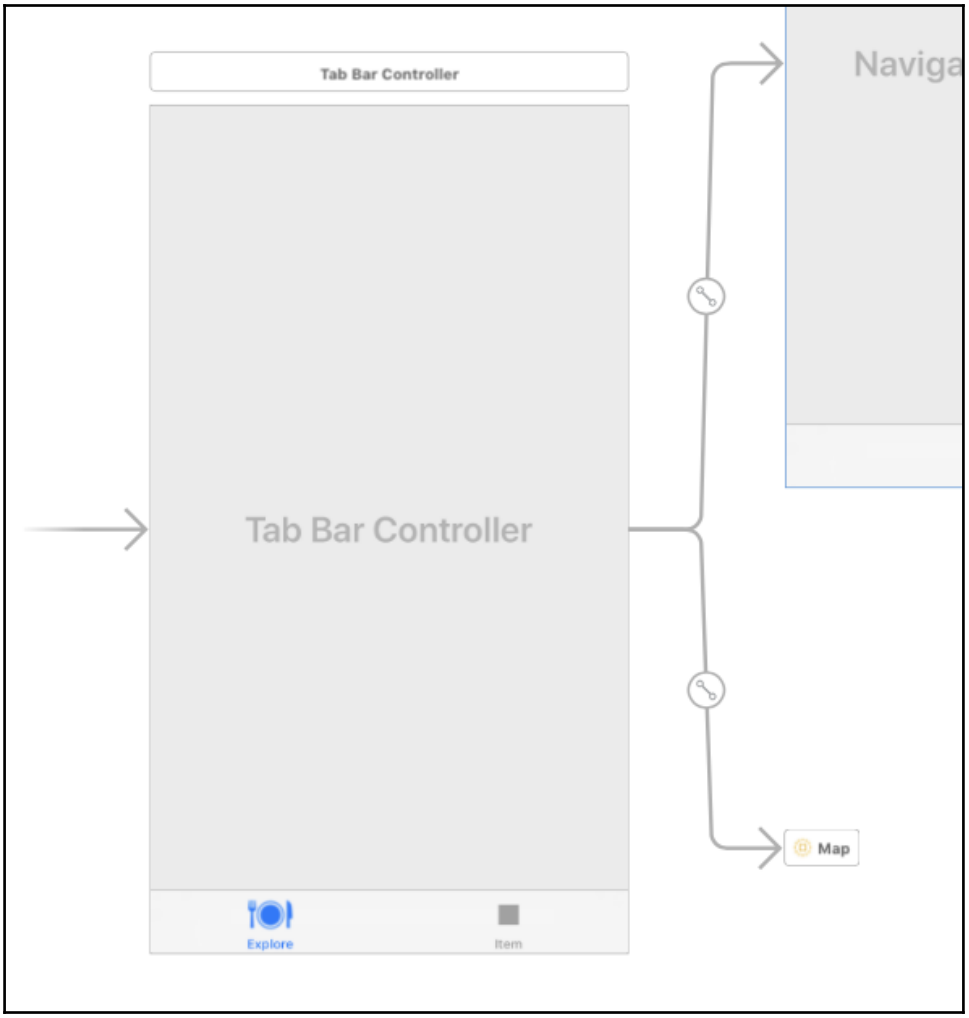

Navigation Controller
Root View Controller
Root View Controller
Prototype Cells
Navigation Controller
Table View
Prototype Content
Table View Controller
Back
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
Navigation Controller
Navigation Controller
View Controller
Cancel
Save
UIImageView
UIImageView
Label

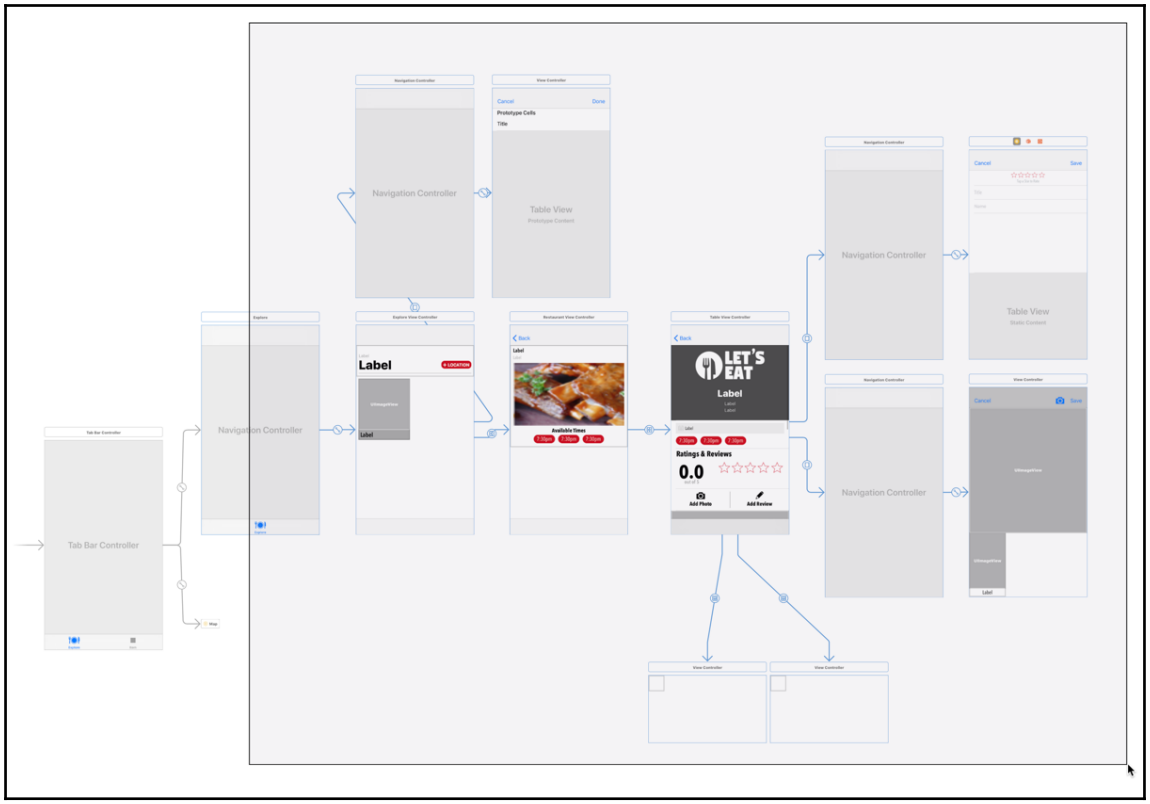

Root View Controller
Table View
Static Content

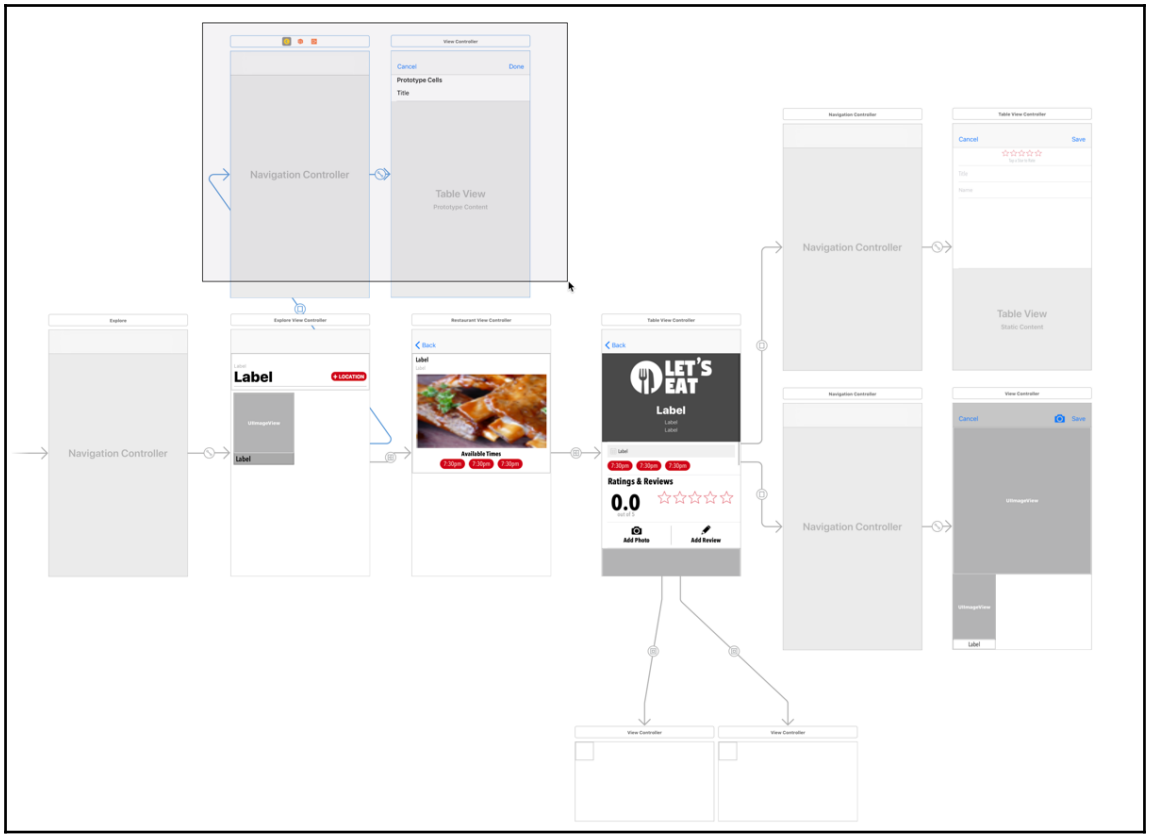

Cancel
Save
Tap a Star to Rate
Title
Name
Table View
Static Content

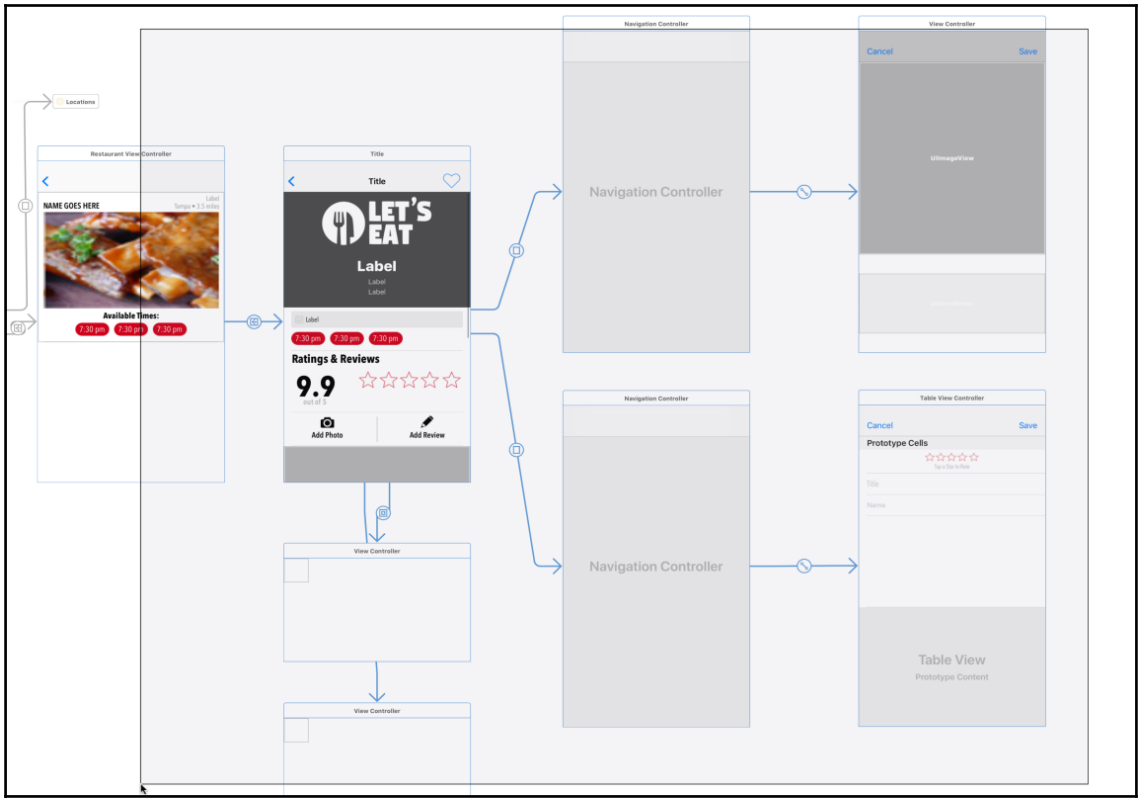

Navigation Controller
Table View
Cancel
Done
Prototype Cells
Title
Navigation Controller
Table View
Cancel
Save
Explore
Navigation Controller
Label
Label
Back
Label
Available Times
LET'S EAT
Label
Ratings & Reviews
0.0
Add Photo
Add Review
Navigation Controller
Cancel
Save
Tab Bar Controller
Navigation Controller
Navigation Controller

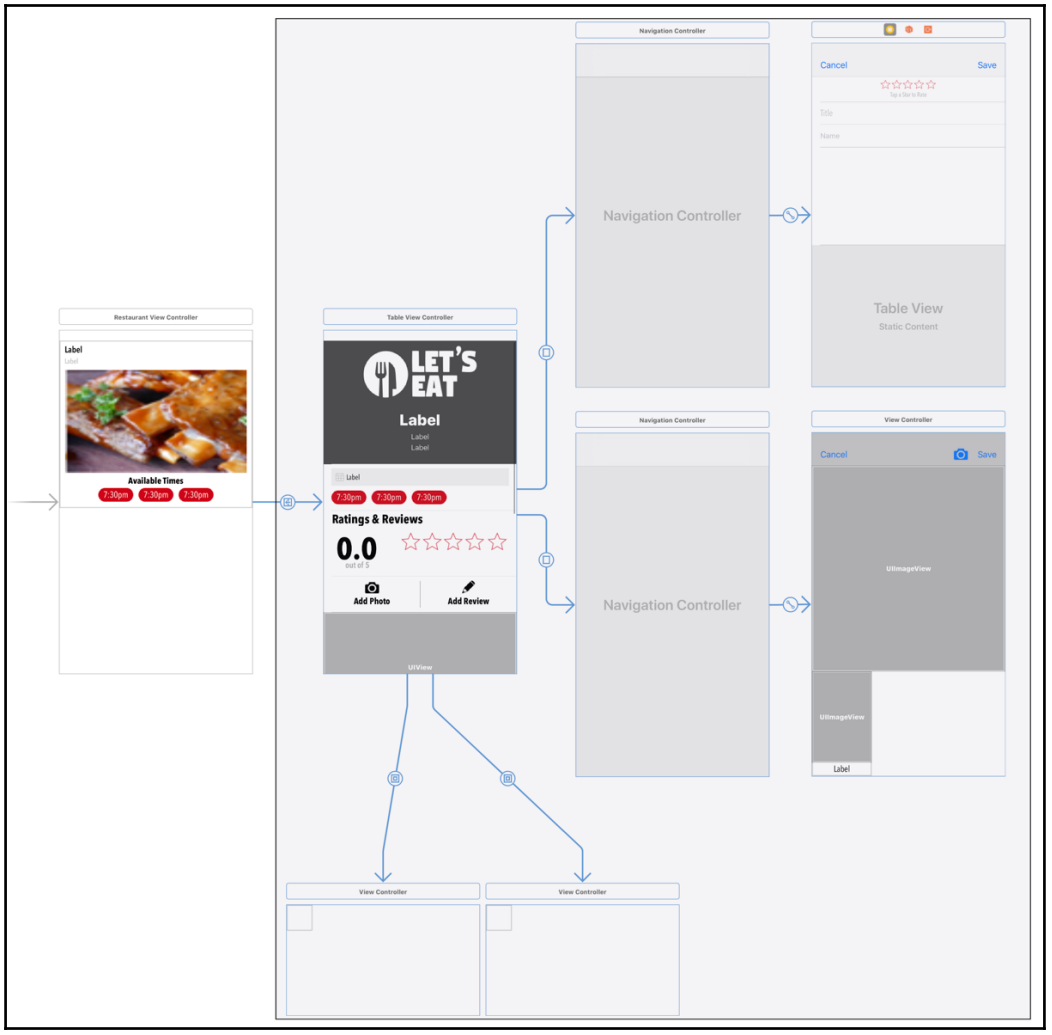

Navigation Controller
Table View
Prototype Content
Cancel
Done
Prototype Cells
Title
Navigation Controller
Table View
Static Content
Cancel
Save
Tab Bar Controller
Navigation Controller
Label
+ LOCATION
Label
Available Times
Back
LET'S EAT
Label
Ratings & Reviews
0.0
Add Photo
Add Review
Navigation Controller
Navigation Controller
Cancel
Save
Label
Navigation Controller

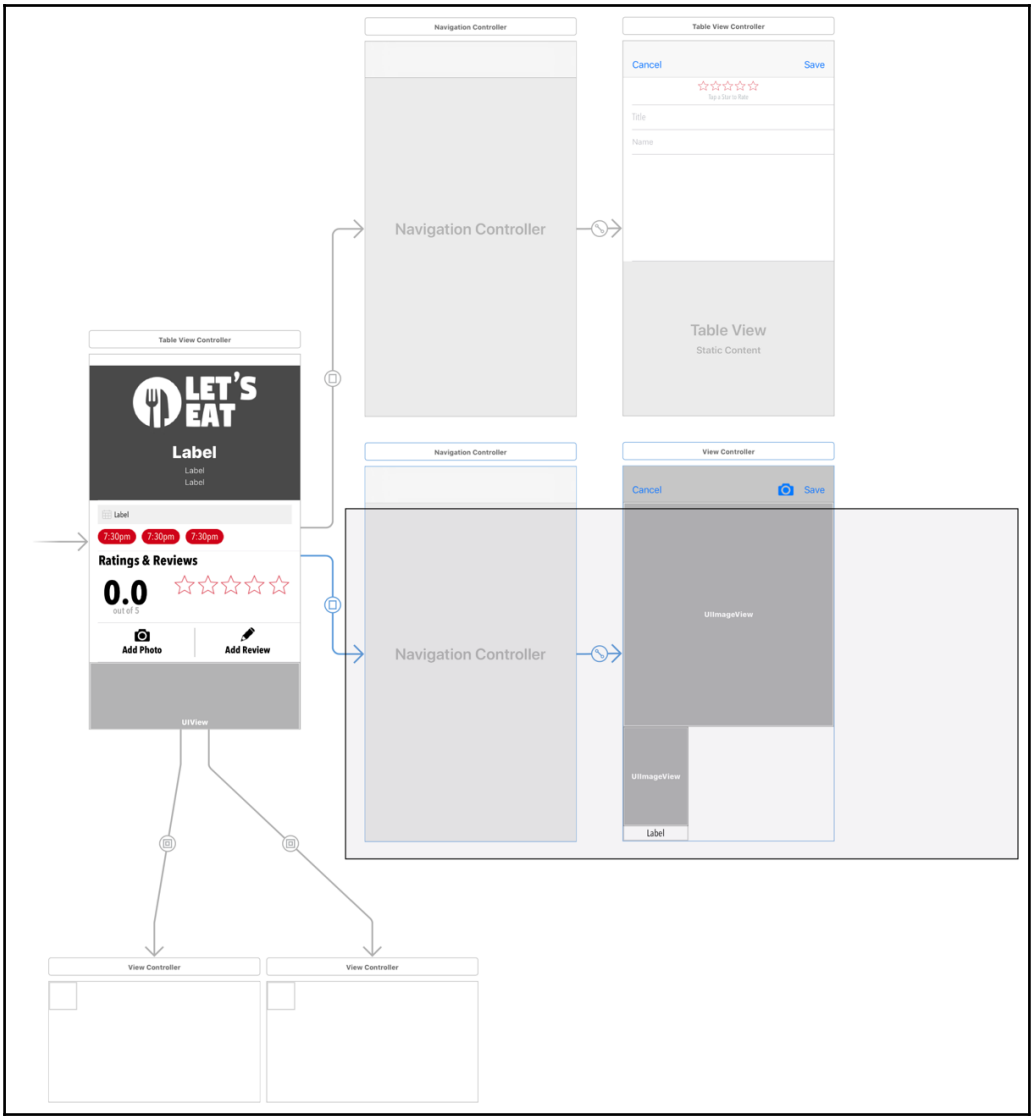

Tab Bar Controller
Tab Bar Controller
Explore
Item
Naviga
Map

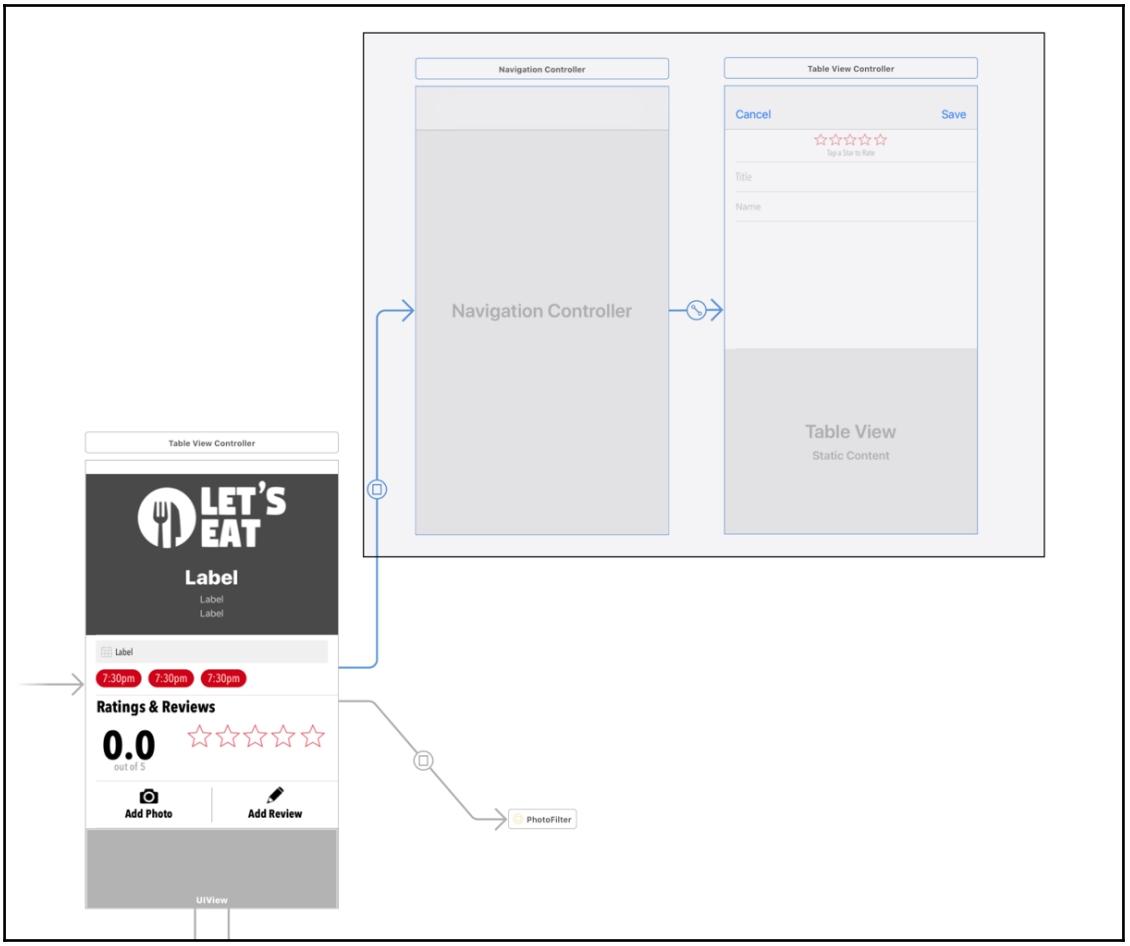

Navigation Controller
Table View
Prototype Content
Cancel
Done
Prototype Cells
Title
Navigation Controller
Table View
Static Content
Cancel
Save
Tab Bar Controller
Navigation Controller
Label
Label
Back
LET'S EAT
Label
Ratings & Reviews
0.0
Add Photo
Add Review
Navigation Controller
Navigation Controller
Cancel
Save
Label

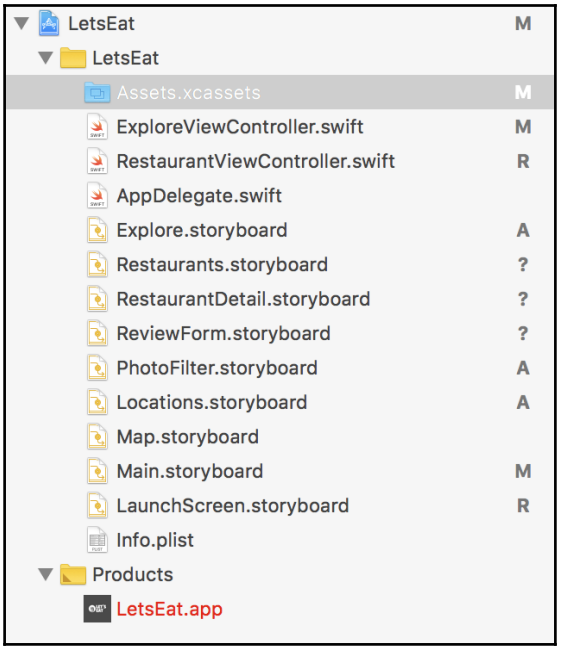

View Controller
Cancel
Done
Prototype Cells
Title
Navigation Controller
Table View
Prototype Content
Navigation Controller
Table View Controller
Cancel
Save
Table View
Static Content
Explore
Navigation Controller
Explore View Controller
Label
LOCATION
Label
Restaurant View Controller
Back
Label
Available Times
7:30pm
7:30pm
7:30pm
Table View Controller
Back
LET'S EAT
Label
Label
Label
Ratings & Reviews
0.0
Add Photo
Add Review
Navigation Controller
Navigation Controller
View Controller
Cancel
Save
Label
View Controller
View Controller


Locations
Restaurant View Controller
NAME GOES HERE
Available Times:
7:30 pm
7:30 pm
7:30 pm
Title
Title
LET'S EAT
Label
Label
Label
Label
7:30 pm
7:30 pm
7:30 pm
Ratings & Reviews
9.9
Add Photo
Add Review
View Controller
View Controller
Navigation Controller
Navigation Controller
Navigation Controller
Navigation Controller
View Controller
Cancel
Save
UIImageView
Table View Controller
Cancel
Save
Prototype Cells
Title
Name
Table View
Prototype Content

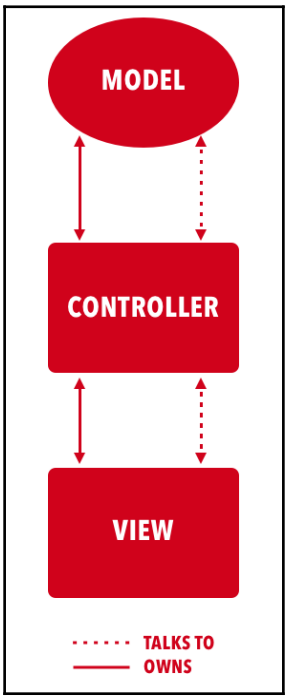

Navigation Controller
Cancel
Save
Navigation Controller
Table View
Static Content
Restaurant View Controller
Label
Available Times
7:30pm
7:30pm
7:30pm
Table View Controller
LET'S EAT
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
Add Photo
Add Review
UIView
Navigation Controller
Navigation Controller
View Controller
Cancel
Save
UIImageView
UIImageView
Label
View Controller
View Controller

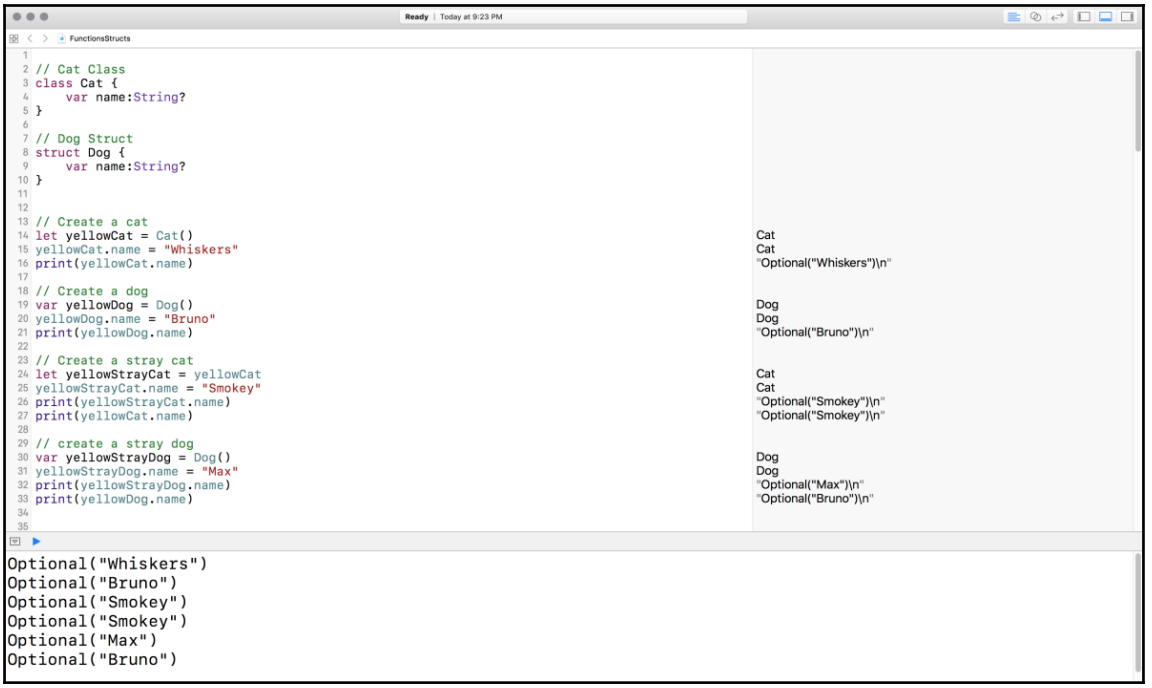

Navigation Controller
Table View Controller
Cancel
Save
Title
Name
Navigation Controller
Table View
Static Content
Table View Controller
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
Navigation Controller
View Controller
Cancel
Save
Navigation Controller
UIImageView
UIImageView
Label
View Controller
View Controller

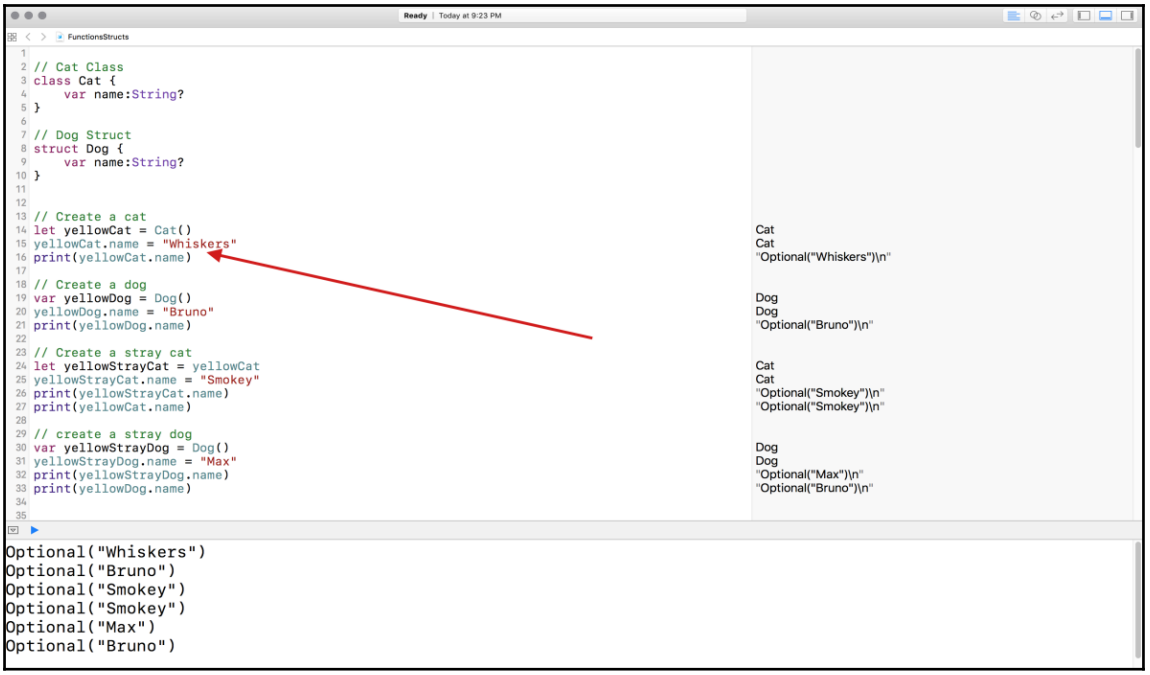

Navigation Controller
Table View Controller
Cancel
Save
Tap a Star to Rate
Title
Name
Navigation Controller
Table View
Static Content
Table View Controller
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
PhotoFilter

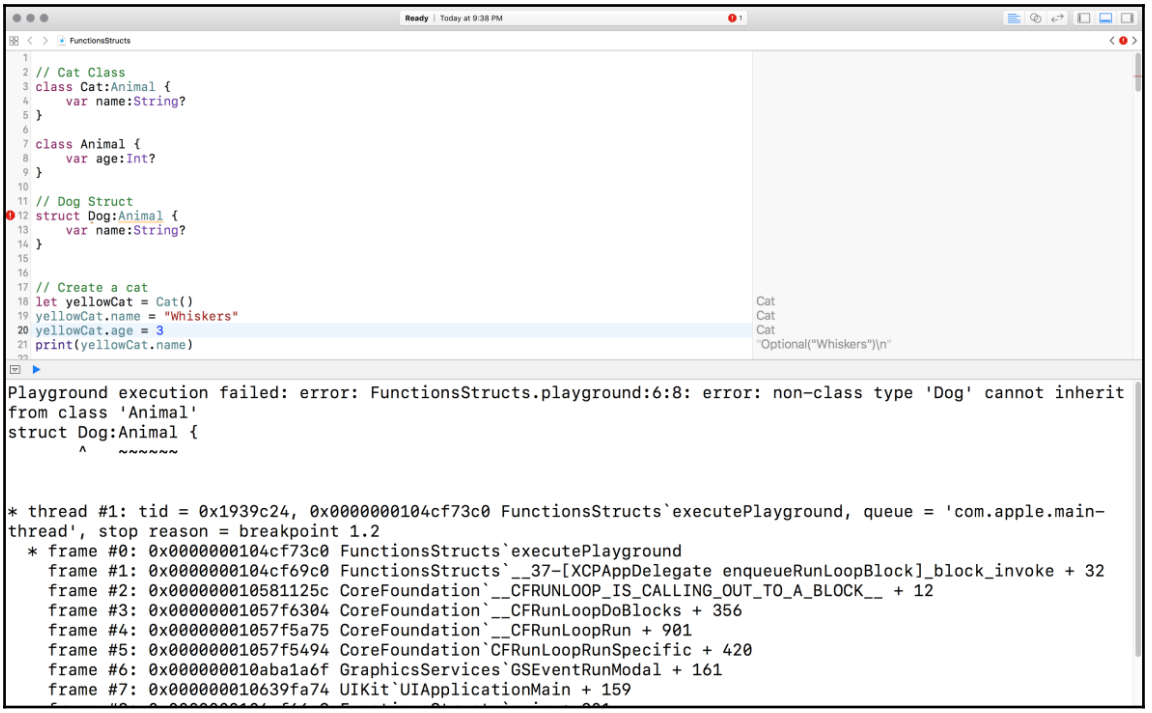

LetsEat M
LetsEat
Assets.xcassets M
ExploreViewController.swift M
RestaurantViewController.swift R
AppDelegate.swift
Explore.storyboard A
Restaurants.storyboard ?
RestaurantDetail.storyboard ?
ReviewForm.storyboard ?
PhotoFilter.storyboard A
Locations.storyboard A
Map.storyboard
Main.storyboard M
LaunchScreen.storyboard R
Info.plist
Products
LetsEat.app

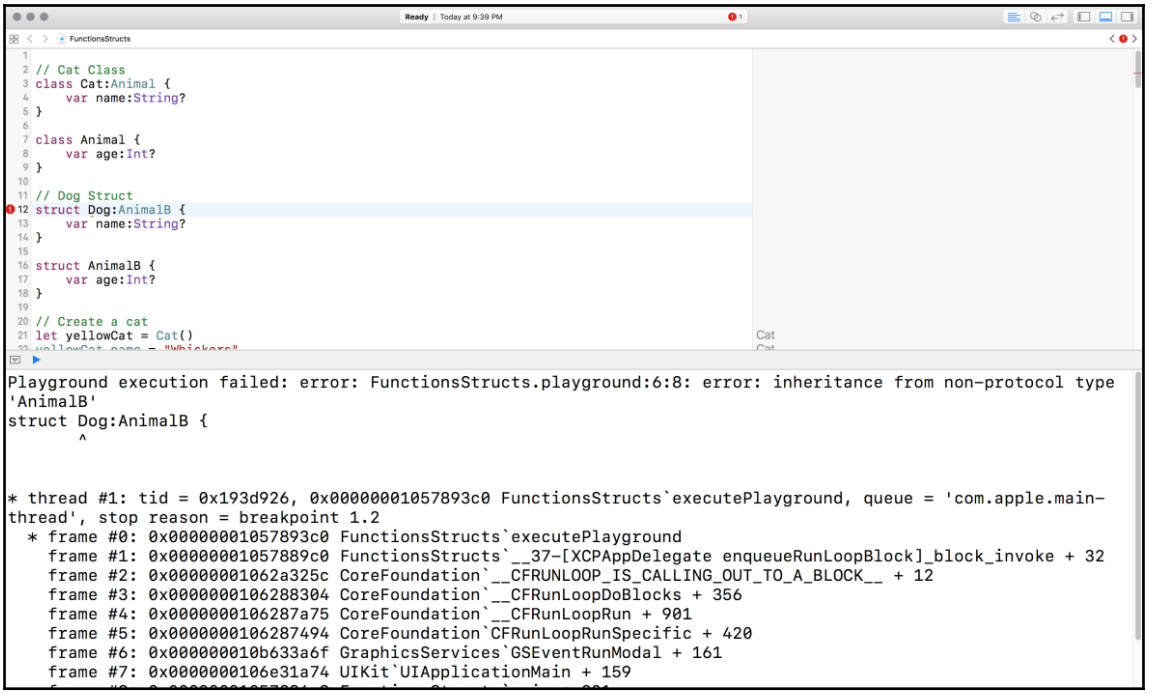
Chapter 13: Getting Started with the Grid

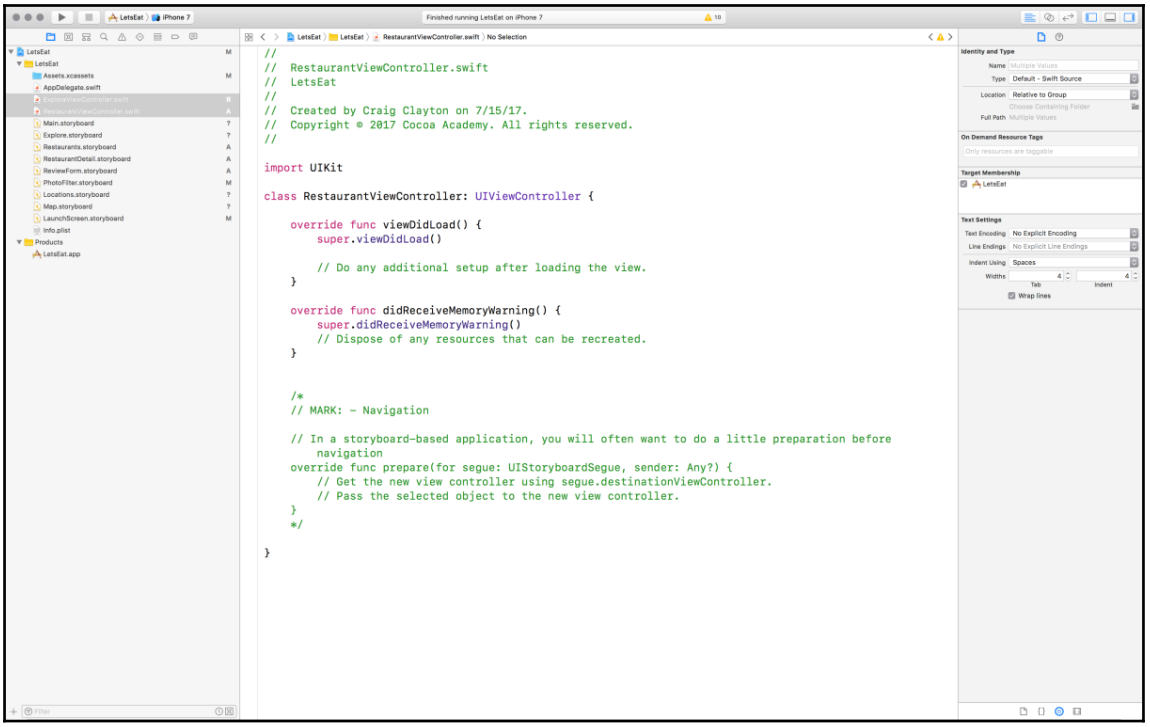



Ready | Today at 9:23 PM
FunctionsStructs
// Cat Class
class Cat {
var name:String?
}
// Dog Struct
struct Dog {
var name:String?
}
// Create a cat
let yellowCat = Cat()
yellowCat.name = "Whiskers"
print(yellowCat.name)
// Create a dog
var yellowDog = Dog()
yellowDog.name = "Bruno"
print(yellowDog.name)
// Create a stray cat
let yellowStrayCat = yellowCat
yellowStrayCat.name = "Smokey"
print(yellowStrayCat.name)
print(yellowCat.name)
// create a stray dog
var yellowStrayDog = Dog()
yellowStrayDog.name = "Max"
print(yellowStrayDog.name)
print(yellowDog.name)
Cat
Cat
"Optional("Whiskers")\n"
Dog
Dog
"Optional("Bruno")\n"
Cat
Cat
"Optional("Smokey")\n"
"Optional("Smokey")\n"
Dog
Dog
"Optional("Max")\n"
"Optional("Bruno")\n"
Optional("Whiskers")
Optional("Bruno")
Optional("Smokey")
Optional("Smokey")
Optional("Max")
Optional("Bruno")

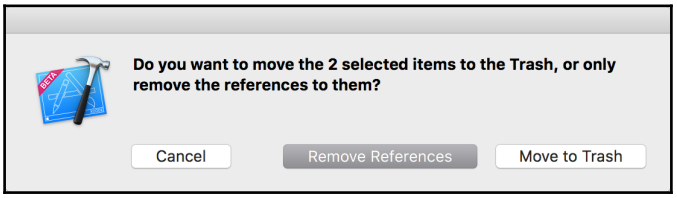

Ready | Today at 9:23 PM
FunctionsStructs
// Cat Class
class Cat {
var name:String?
}
// Dog Struct
struct Dog {
var name:String?
}
// Create a cat
let yellowCat = Cat()
yellowCat.name = "Whiskers"
print(yellowCat.name)
// Create a dog
var yellowDog = Dog()
yellowDog.name = "Bruno"
print(yellowDog.name)
// Create a stray cat
let yellowStrayCat = yellowCat
yellowStrayCat.name = "Smokey"
print(yellowStrayCat.name)
print(yellowCat.name)
// create a stray dog
var yellowStrayDog = Dog()
yellowStrayDog.name = "Max"
print(yellowStrayDog.name)
print(yellowDog.name)
Cat
Cat
"Optional("Whiskers")\n"
Dog
Dog
"Optional("Bruno")\n"
Cat
Cat
"Optional("Smokey")\n"
"Optional("Smokey")\n"
Dog
Dog
"Optional("Max")\n"
"Optional("Bruno")\n"
Optional("Whiskers")
Optional("Bruno")
Optional("Smokey")
Optional("Smokey")
Optional("Max")
Optional("Bruno")

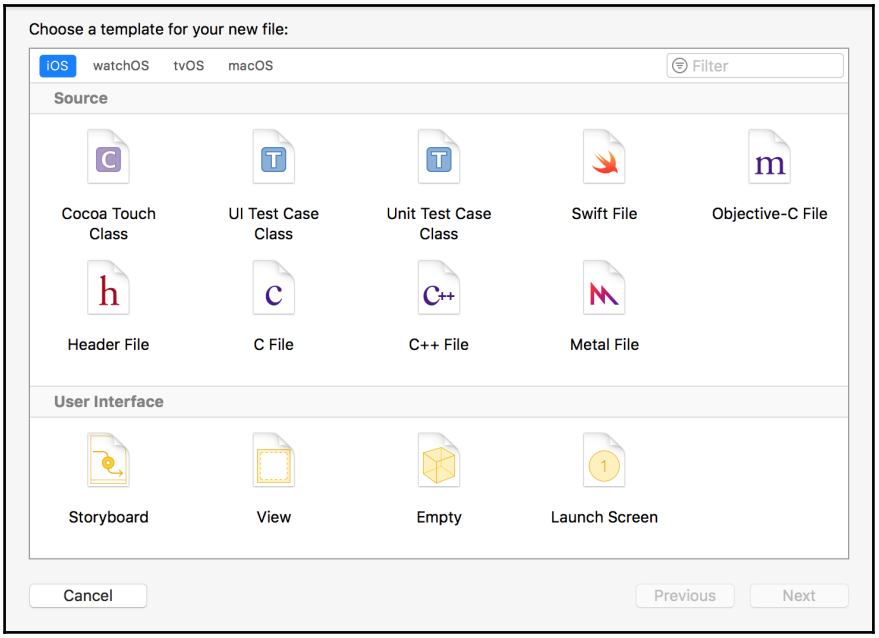

Ready | Today at 9:38 PM
FunctionsStructs
// Cat Class
class Cat:Animal {
var name:String?
}
class Animal {
var age:Int?
}
// Dog Struct
struct Dog:Animal {
var name:String?
}
// Create a cat
let yellowCat = Cat()
yellowCat.name = "Whiskers"
yellowCat.age = 3
print(yellowCat.name)
Cat
Cat
Cat
"Optional("Whiskers")\n"
Playground execution failed: error: FunctionsStructs.playground:6:8: error: non-class type 'Dog' cannot inherit from class 'Animal'
struct Dog:Animal {
^ ~~~~~~
* thread #1: tid = 0x1939c24, 0x0000000104cf73c0 FunctionsStructs`executePlayground, queue = 'com.apple.main-thread', stop reason = breakpoint 1.2
* frame #0: 0x0000000104cf73c0 FunctionsStructs`executePlayground
frame #1: 0x0000000104cf69c0 FunctionsStructs`__37-[XCPAppDelegate enqueueRunLoopBlock]_block_invoke + 32
frame #2: 0x000000010581125c CoreFoundation`__CFRUNLOOP_IS_CALLING_OUT_TO_A_BLOCK__ + 12
frame #3: 0x00000001057f6304 CoreFoundation`__CFRunLoopDoBlocks + 356
frame #4: 0x00000001057f5a75 CoreFoundation`__CFRunLoopRun + 901
frame #5: 0x00000001057f5494 CoreFoundation`CFRunLoopRunSpecific + 420
frame #6: 0x000000010aba1a6f GraphicsServices`GSEventRunModal + 161
frame #7: 0x000000010639fa74 UIKit`UIApplicationMain + 159


Ready | Today at 9:39 PM
FunctionsStructs
// Cat Class
class Cat:Animal {
var name:String?
}
class Animal {
var age:Int?
}
// Dog Struct
struct Dog:AnimalB {
var name:String?
}
struct AnimalB {
var age:Int?
}
// Create a cat
let yellowCat = Cat()
Cat
Playground execution failed: error: FunctionsStructs.playground:6:8: error: inheritance from non-protocol type 'AnimalB'
struct Dog:AnimalB {
^
* thread #1: tid = 0x193d926, 0x00000001057893c0 FunctionsStructs`executePlayground, queue = 'com.apple.main-thread', stop reason = breakpoint 1.2
* frame #0: 0x00000001057893c0 FunctionsStructs`executePlayground
frame #1: 0x00000001057889c0 FunctionsStructs`__37-[XCPAppDelegate enqueueRunLoopBlock]_block_invoke + 32
frame #2: 0x00000001062a325c CoreFoundation`__CFRUNLOOP_IS_CALLING_OUT_TO_A_BLOCK__ + 12
frame #3: 0x0000000106288304 CoreFoundation`__CFRunLoopDoBlocks + 356
frame #4: 0x0000000106287a75 CoreFoundation`__CFRunLoopRun + 901
frame #5: 0x0000000106287494 CoreFoundation`CFRunLoopRunSpecific + 420
frame #6: 0x000000010b633a6f GraphicsServices`GSEventRunModal + 161
frame #7: 0x0000000106e31a74 UIKit`UIApplicationMain + 159



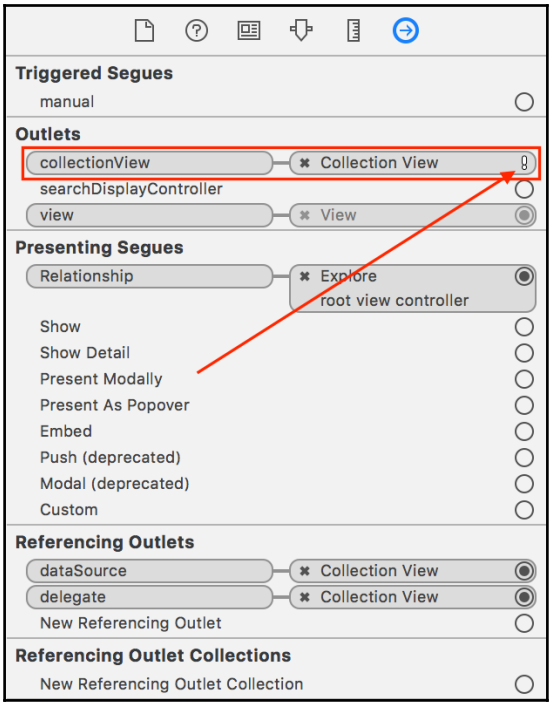

```
//
//  RestaurantViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 7/15/17.
//  Copyright © 2017 Cocoa Academy. All rights reserved.
//

import UIKit

class RestaurantViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before
        navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */

}
```

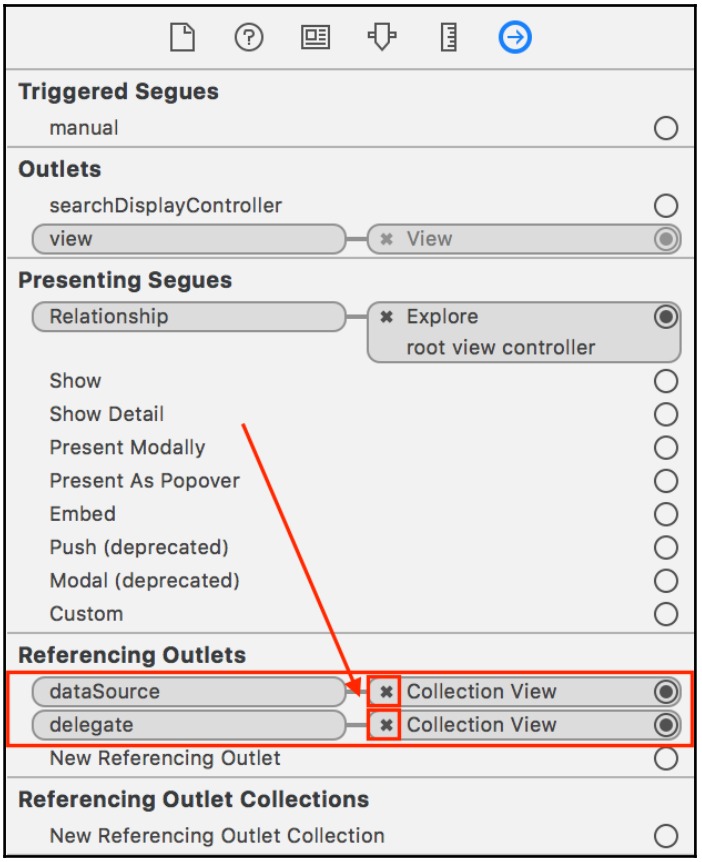


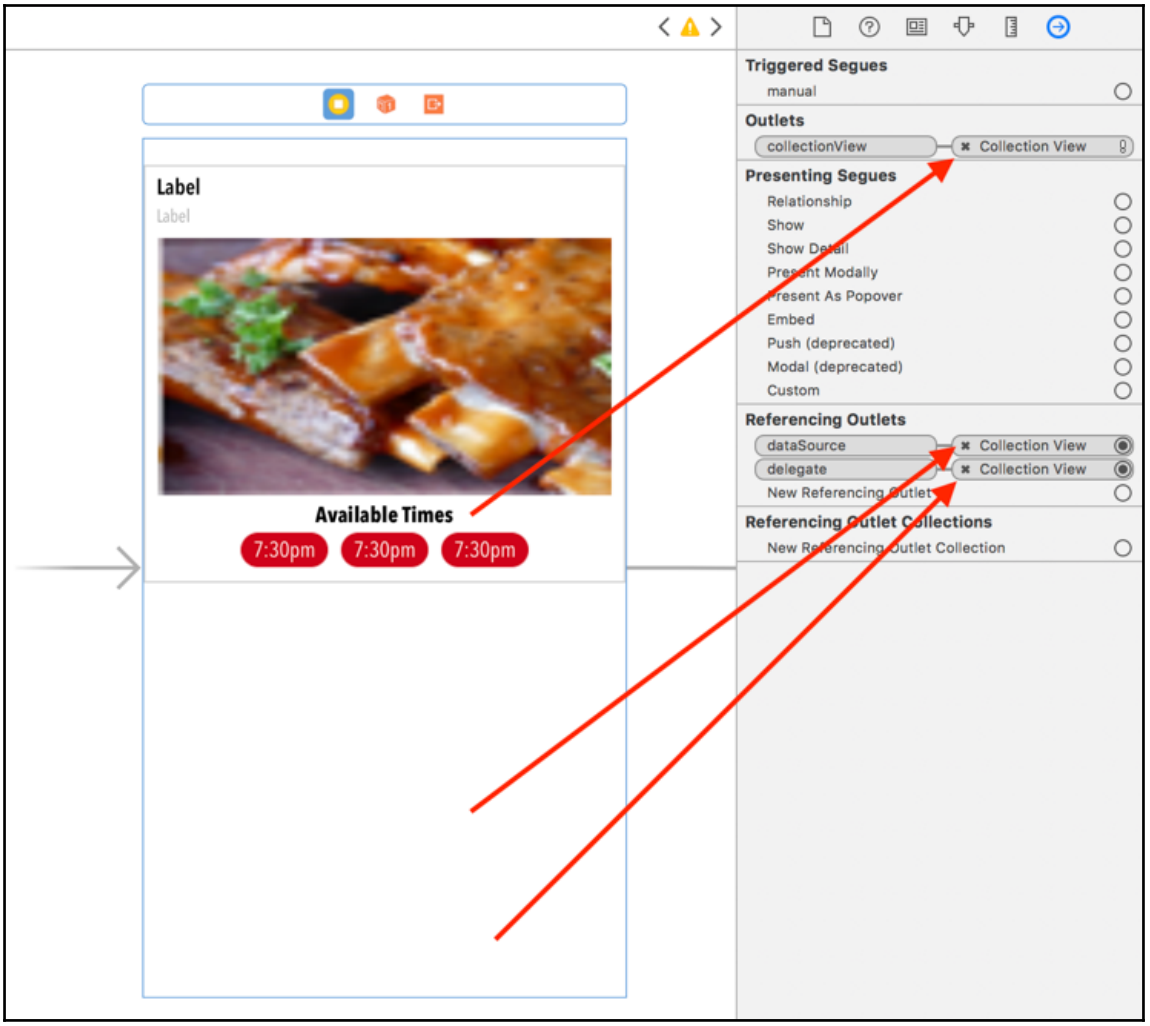

Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
User Interface
Storyboard
View
Empty
Launch Screen
Cancel
Previous
Next

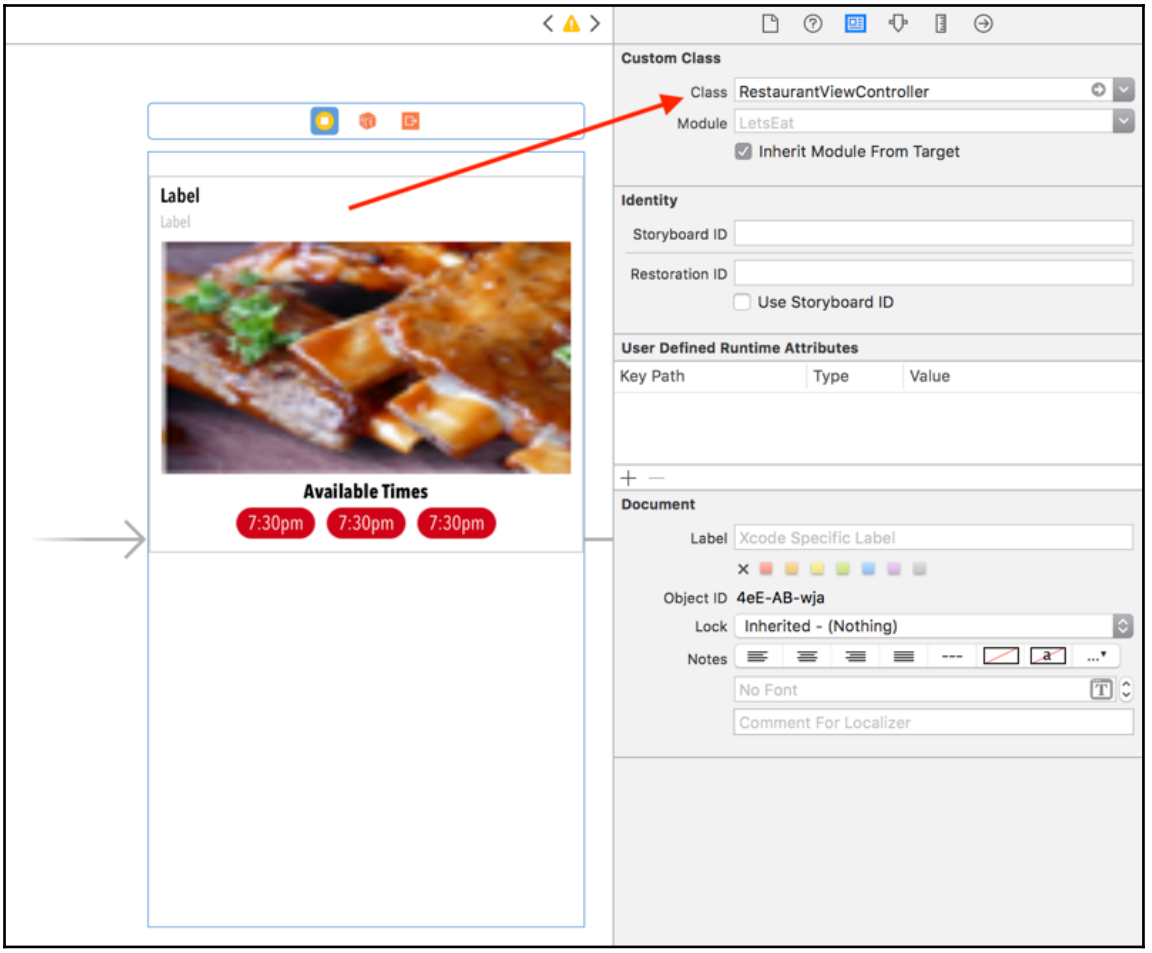

import UIKit
class ExploreViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
DELETE
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
/*
// MARK: - Navigation
// In a storyboard-based application, you will often want to do a little preparation before navigation
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
// Get the new view controller using segue.destinationViewController.
// Pass the selected object to the new view controller.
}
*/
}
DELETE
import UIKit
class ExploreViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
UPDATED

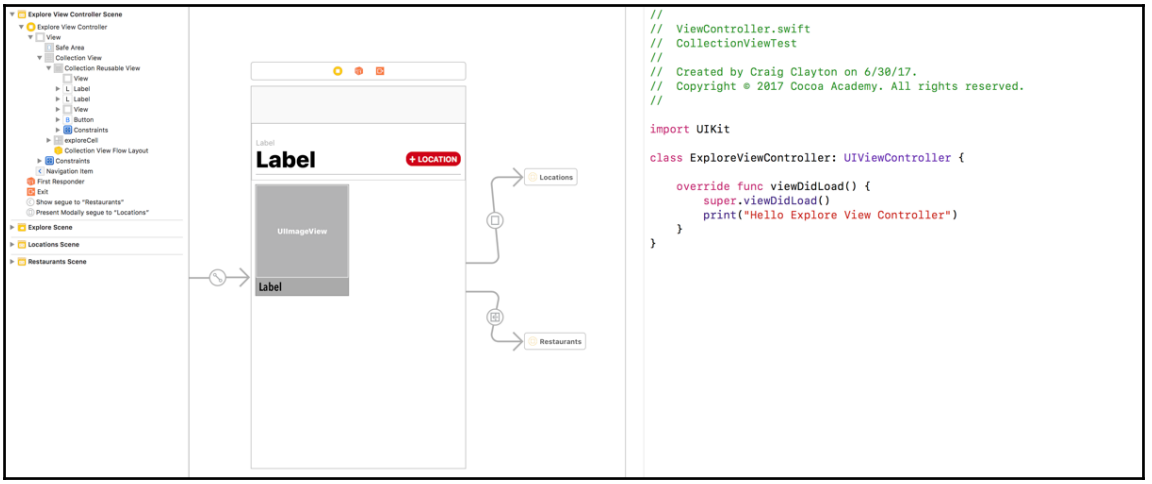

import UIKit
class ExploreViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
DELETE
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
/*
// MARK: - Navigation
// In a storyboard-based application, you will often want to do a little preparation before navigation
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
// Get the new view controller using segue.destinationViewController.
// Pass the selected object to the new view controller.
}
*/
}
DELETE
import UIKit
class ExploreViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
print("Hello Explore View Controller")
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
UPDATED


Triggered Segues
manual
Outlets
collectionView
Collection View
searchDisplayController
view
View
Presenting Segues
Relationship
Explore
root view controller
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Collection View
delegate
Collection View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


Triggered Segues
manual
Outlets
searchDisplayController
view
View
Presenting Segues
Relationship
Explore
root view controller
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Collection View
delegate
Collection View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

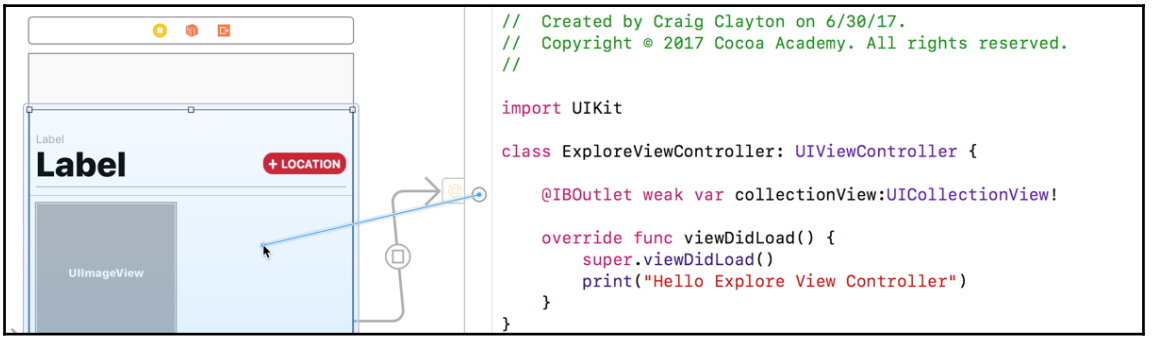

Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Triggered Segues
manual
Outlets
collectionView
Collection View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Collection View
delegate
Collection View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Custom Class
Class RestaurantViewController
Module LetsEat
Inherit Module From Target
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label Xcode Specific Label
Object ID 4eE-AB-wja
Lock Inherited - (Nothing)
Notes
No Font
Comment For Localizer

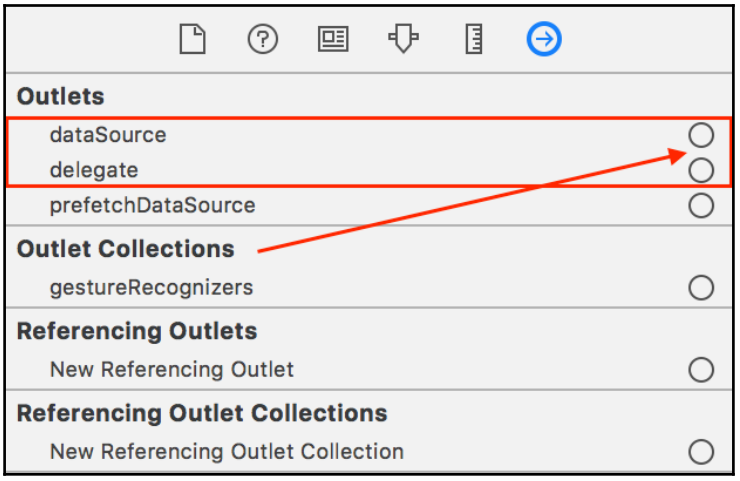


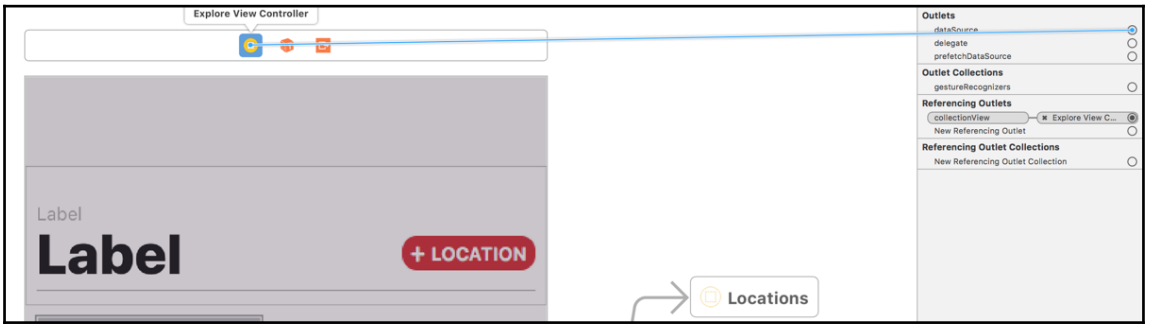

```
@IBOutlet weak var collectionView:UICollectionView!
```

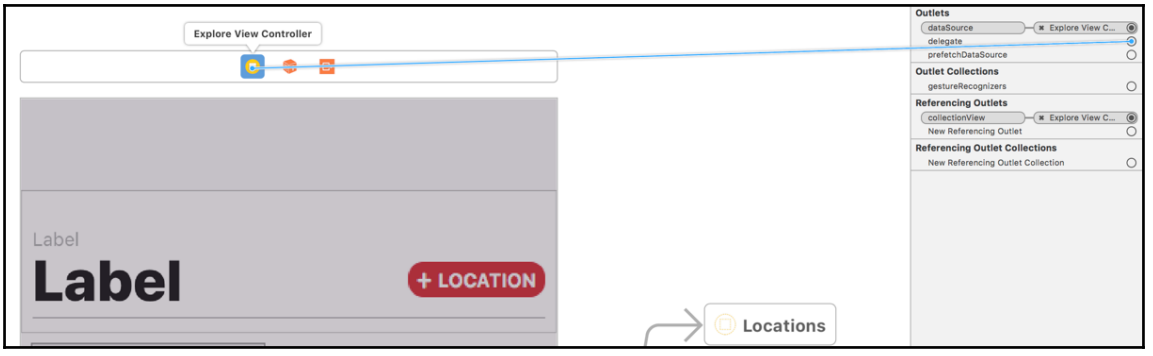
```
@IBOutlet weak var collectionView:UICollectionView!
```




```
@IBOutlet weak var collectionView:UICollectionView!
```


Outlets
dataSource
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


Explore View Controller
Label
Label
+ LOCATION
Locations
Outlets
dataSource
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
collectionView
Explore View C...
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


Explore View Controller
Label
Label
+ LOCATION
Locations
Outlets
dataSource
Explore View C...
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
collectionView
Explore View C...
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

```swift
import UIKit

class ExploreViewController: UIViewController, UICollectionViewDataSource, UICollectionViewDelegate {

    @IBOutlet weak var collectionView: UICollectionView!

    override func viewDidLoad() {
        super.viewDidLoad()
        print("Hello Explore View Controller")
    }

    func collectionView(_ collectionView: UICollectionView, viewForSupplementaryElementOfKind kind: String, at indexPath: IndexPath) -> A
        UICollectionReusableView {
        B let headerView = collectionView.dequeueReusableSupplementaryView(ofKind: kind, withReuseIdentifier: "header", for: indexPath)
        return headerView
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell { C
D       return collectionView.dequeueReusableCell(withReuseIdentifier: "exploreCell", for: indexPath)
    }

    func numberOfSections(in collectionView: UICollectionView) -> Int { E
F       return 1
    }

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int { G
H       return 20
    }

    // Add Unwind here
I   @IBAction func unwindLocationCancel(segue:UIStoryboardSegue) {}
}
```

Chapter 14: Getting Data into Our Grid

Root		Array	(31 items)
▼ Item 0		Dictionary	(2 items)
name		String	All
image		String	all.png
▼ Item 1		Dictionary	(2 items)
name		String	Bistro
image		String	bistro.png
▼ Item 2		Dictionary	(2 items)
name		String	Bar / Lounge / Bottle Service
image		String	bar.png
▼ Item 3		Dictionary	(2 items)
name		String	Brewery
image		String	brewery.png

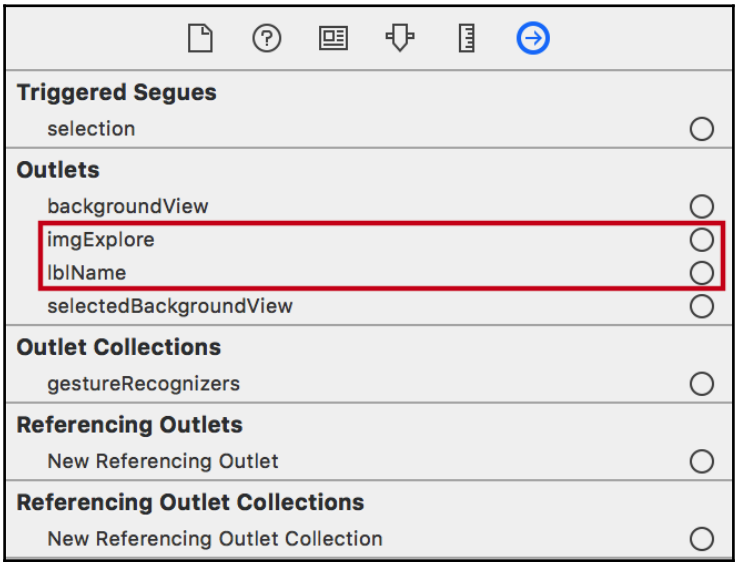
```
import Foundation

class ExploreDataManager {

    fileprivate func loadData() -> [[String: AnyObject]] {
        guard let path = Bundle.main.path(forResource: "ExploreData", ofType: "plist"),
            let items = NSArray(contentsOfFile: path) else {
              return [[:]]
          }

        return items as! [[String : AnyObject]]
    }
}
```

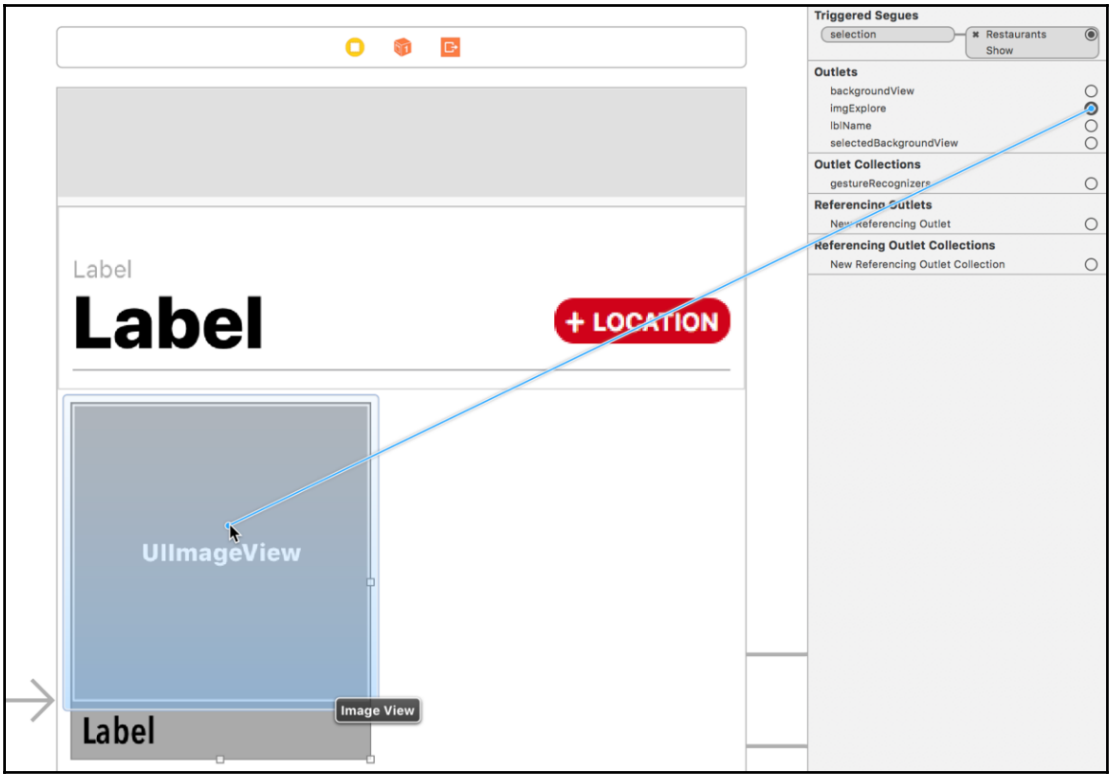

Root		Array	(31 items)
▼ Item 0		Dictionary	(2 items)
name		String	All
image		String	all.png
▼ Item 1		Dictionary	(2 items)
name		String	Bistro
image		String	bistro.png
▼ Item 2		Dictionary	(2 items)
name		String	Bar / Lounge / Bottle Service
image		String	bar.png
▼ Item 3		Dictionary	(2 items)
name		String	Brewery
image		String	brewery.png

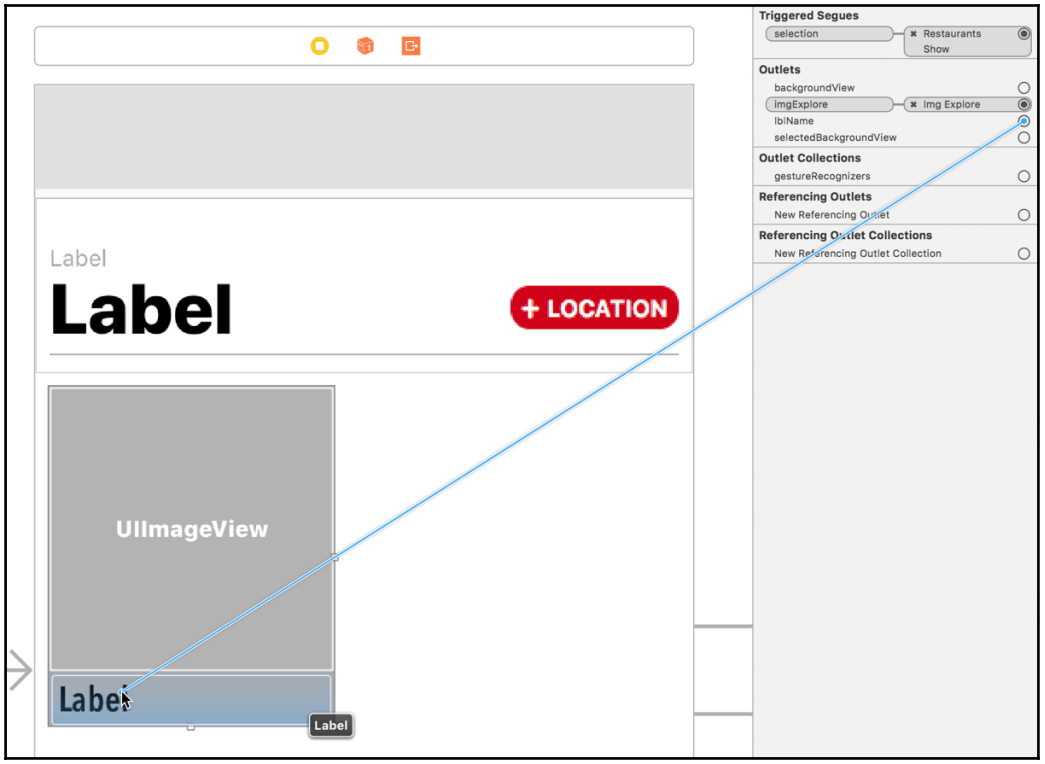
```
import Foundation

class ExploreDataManager {

    func fetch() {
        for data in loadData() {
            print(data)
        }
    }

    fileprivate func loadData() -> [[String: AnyObject]] {
      guard let path = Bundle.main.path(forResource: "ExploreData", ofType: "plist"),
          let items = NSArray(contentsOfFile: path) else {
            return [[:]]
        }

      return items as! [[String : AnyObject]]
    }
}
```

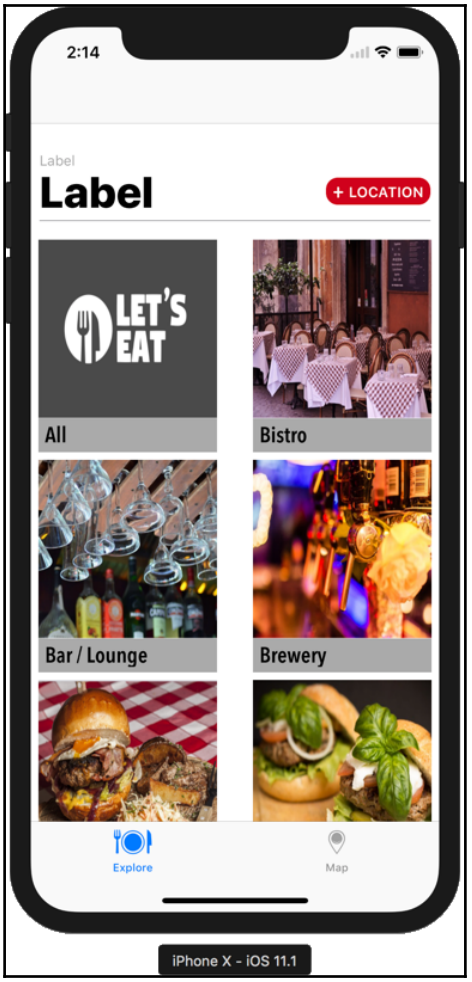
```
["image": all.png, "name": All]
["image": bistro.png, "name": Bistro]
["image": bar.png, "name": Bar / Lounge]
["image": brewery.png, "name": Brewery]
["image": burgers.png, "name": Burgers]
["image": californian.png, "name": Californian]
["image": caribbean.png, "name": Caribbean]
["image": comfort.png, "name": Comfort Food]
["image": cuban.png, "name": Cuban]
["image": continental.png, "name": Continental]
["image": french.png, "name": French]
["image": international.png, "name": International]
["image": italian.png, "name": Italian]
["image": japanese.png, "name": Japanese]
["image": latin.png, "name": Latin American]
["image": mediterranean.png, "name": Mediterranean]
["image": mexican.png, "name": Mexican]
["image": organic.png, "name": Organic]
["image": panasian.png, "name": Pan-Asian]
["image": peruvian.png, "name": Peruvian]
["image": pizza.png, "name": Pizzeria]
["image": primerib.png, "name": Prime Rib]
["image": seafood.png, "name": Seafood]
["image": southamerican.png, "name": South American]
["image": southern.png, "name": Southern]
["image": spanish.png, "name": Spanish]
["image": steak.png, "name": Steakhouse]
["image": sushi.png, "name": Sushi]
["image": tapas.png, "name": Tapas / Small Plates]
["image": vietnamese.png, "name": Vietnamese]
["image": wine.png, "name": Wine Bar]
```

```
import Foundation

class ExploreDataManager {

    fileprivate var items:[ExploreItem] = []

    func fetch() {
        for data in loadData() {
            items.append(ExploreItem(dict: data))
        }
    }

    fileprivate func loadData() -> [[String: AnyObject]] {
        guard let path = Bundle.main.path(forResource: "ExploreData", ofType: "plist"),
            let items = NSArray(contentsOfFile: path) else {
            return [[:]]
        }

        return items as! [[String : AnyObject]]
    }
}
```

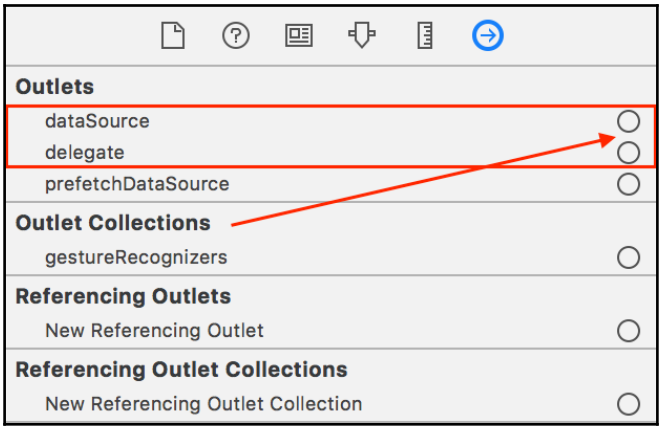


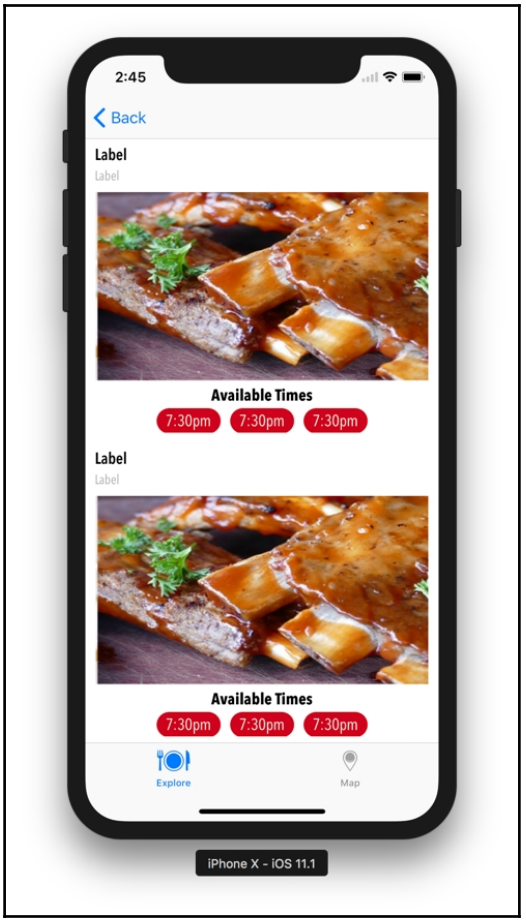

Triggered Segues
selection
Restaurants
Show
Outlets
backgroundView
imgExplore
lblName
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label
Label
+ LOCATION
UIImageView
Image View
Label


Triggered Segues
selection
Restaurants
Show
Outlets
backgroundView
imgExplore
Img Explore
lblName
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label
Label
+ LOCATION
UIImageView
Label
Label




```
@IBOutlet weak var collectionView:UICollectionView!
```


Outlets
dataSource
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

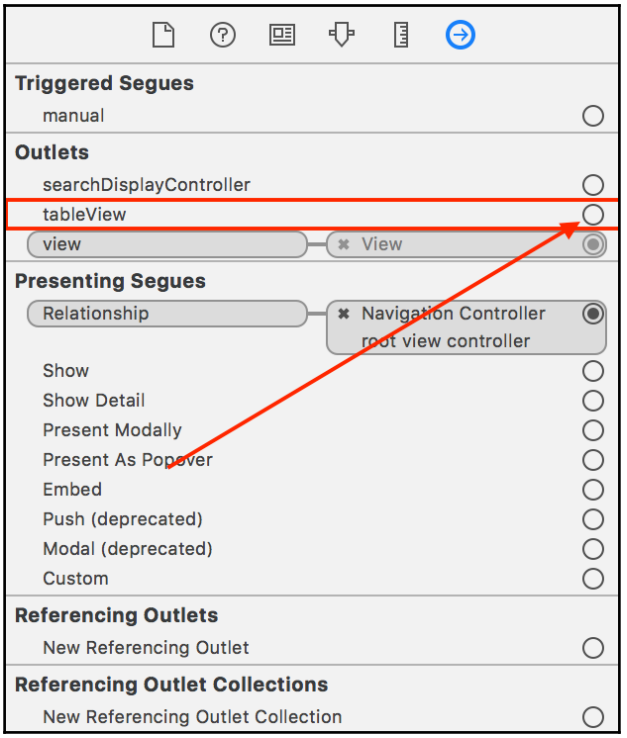

2:45
Back
Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Explore
Map
iPhone X - iOS 11.1

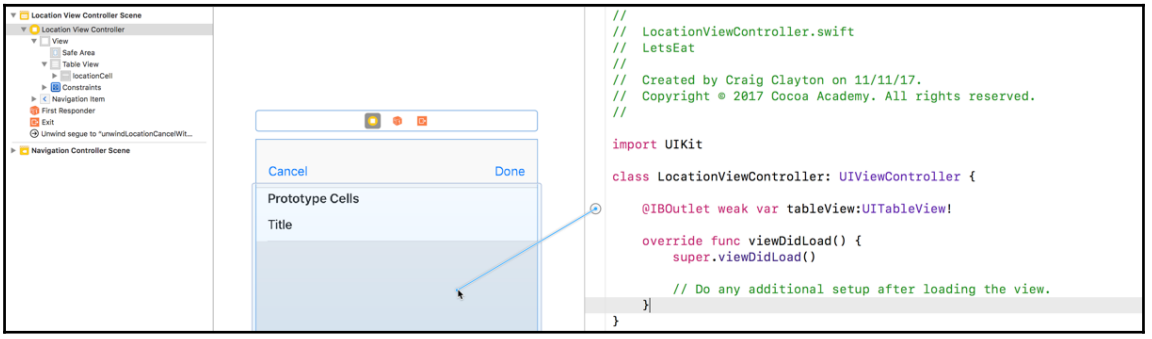
Chapter 15: Getting Started with the List

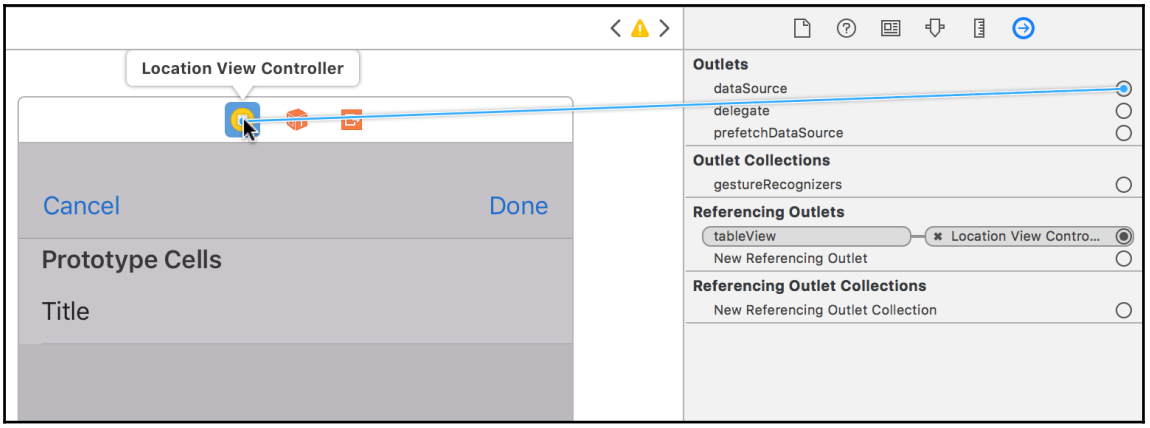
```
class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!
```

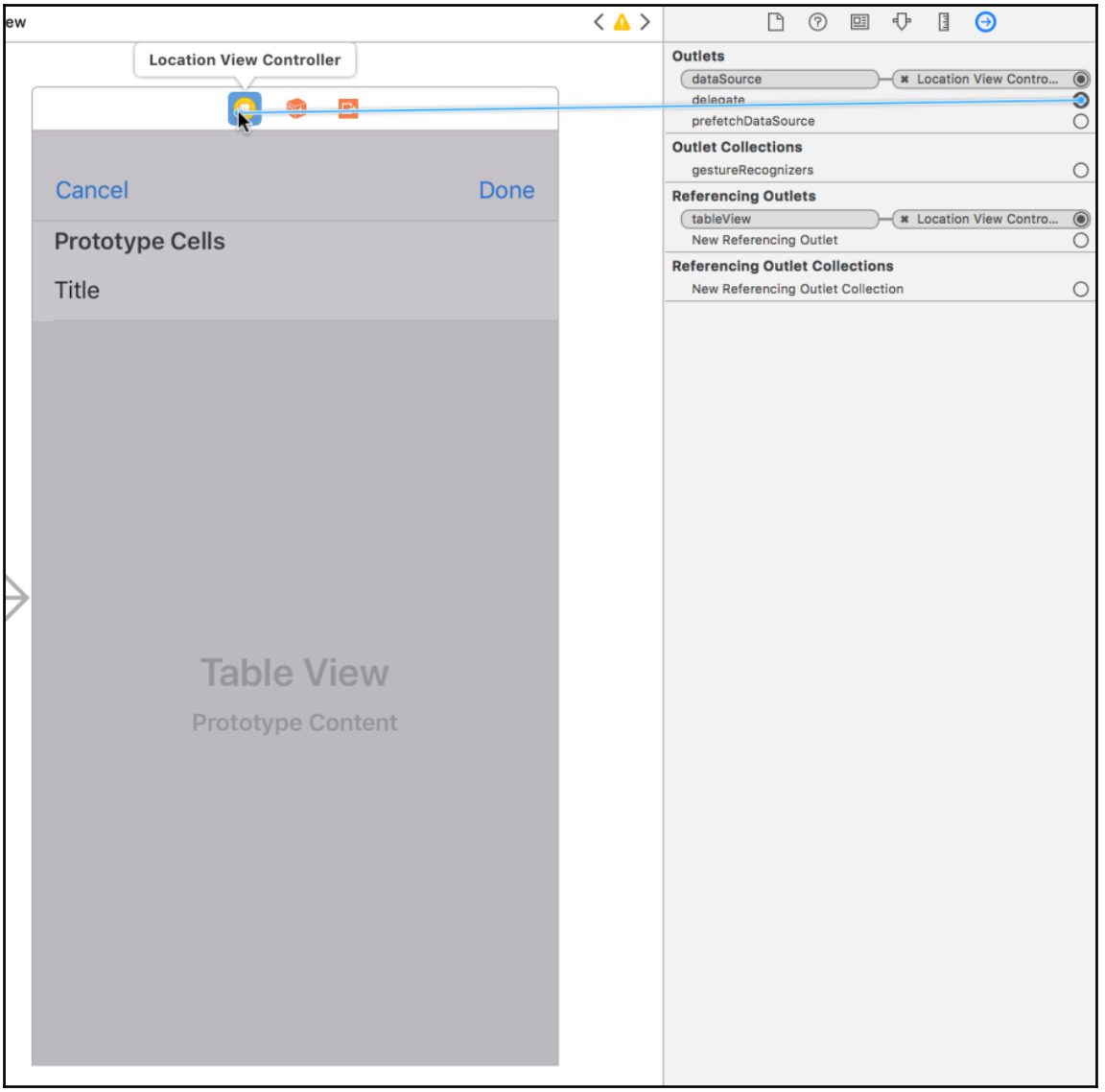
```
//
//  LocationViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 11/19/16.
//  Copyright © 2016 Craig Clayton. All rights reserved.
//

import UIKit

class LocationViewController: UIViewController {

    @IBOutlet var tableView:UITableView!

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */

}
```

Delete

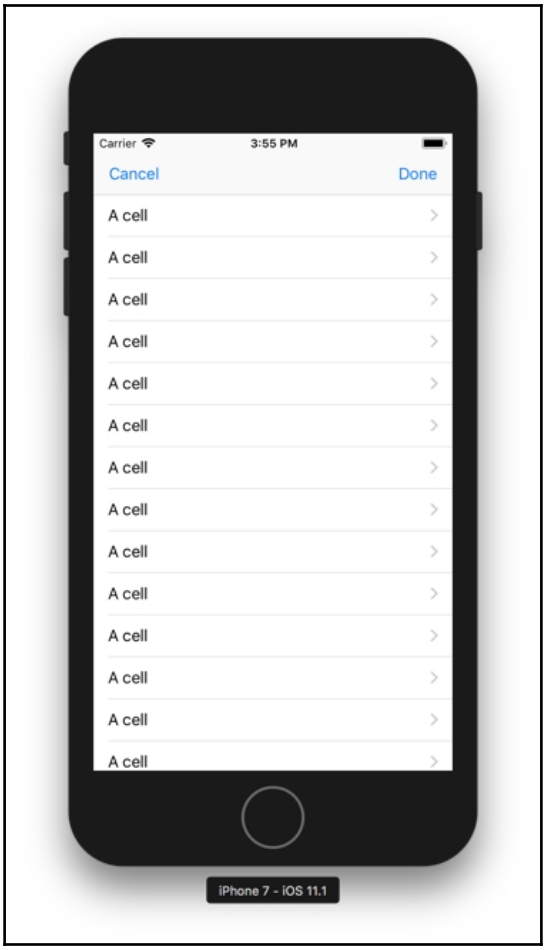

Triggered Segues
manual
Outlets
searchDisplayController
tableView
view
View
Presenting Segues
Relationship
Navigation Controller
root view controller
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


Cancel
Done
Prototype Cells
Title
//
// LocationViewController.swift
// LetsEat
//
// Created by Craig Clayton on 11/11/17.
// Copyright © 2017 Cocoa Academy. All rights reserved.
//
import UIKit
class LocationViewController: UIViewController {
@IBOutlet weak var tableView:UITableView!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
}
}

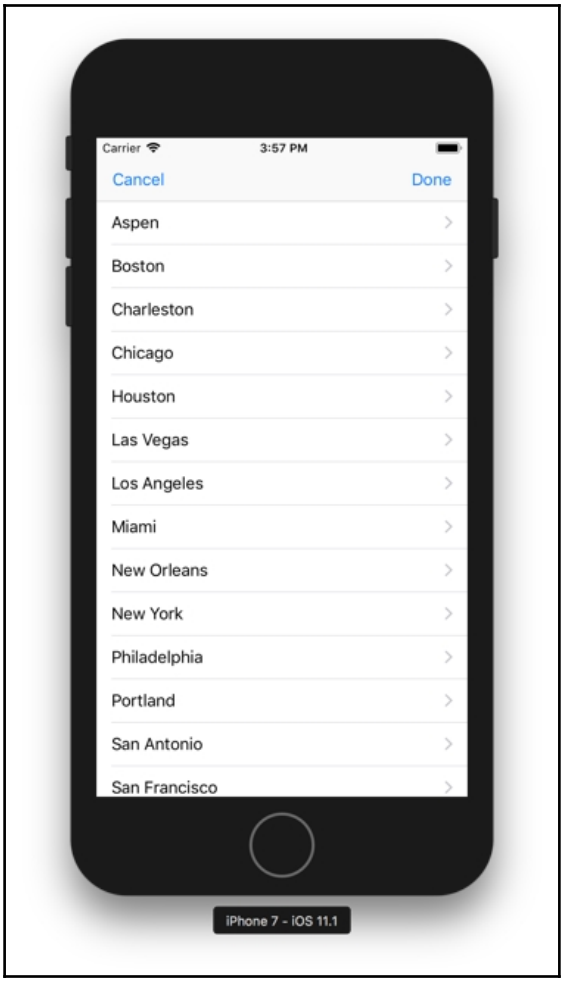

Location View Controller
Cancel
Done
Prototype Cells
Title
Outlets
dataSource
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
tableView
Location View Contro...
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

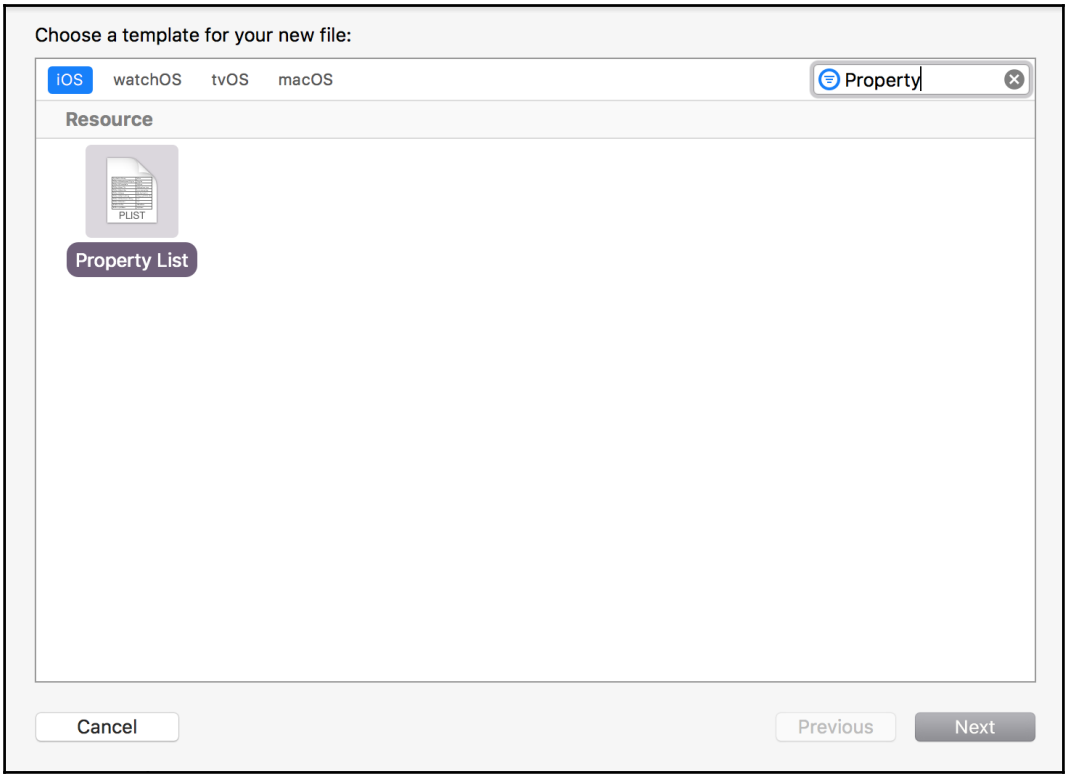

ew
Location View Controller
Cancel
Done
Prototype Cells
Title
Table View
Prototype Content
Outlets
dataSource
Location View Contro...
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
tableView
Location View Contro...
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

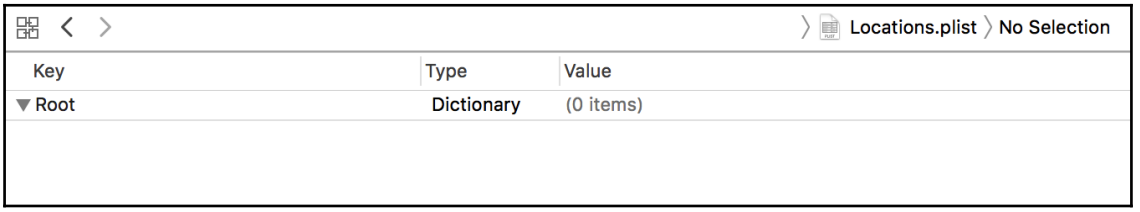

import UIKit
class LocationViewController: UIViewController, UITableViewDataSource {
@IBOutlet weak var tableView:UITableView!
A func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
return 15
B
}
C func numberOfSections(in tableView: UITableView) -> Int {
return 1
D
}
E func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
let cell = tableView.dequeueReusableCell(withIdentifier: "locationCell", for: indexPath) as UITableViewCell
F
cell.textLabel?.text = "A cell"
return cell
G
}
}

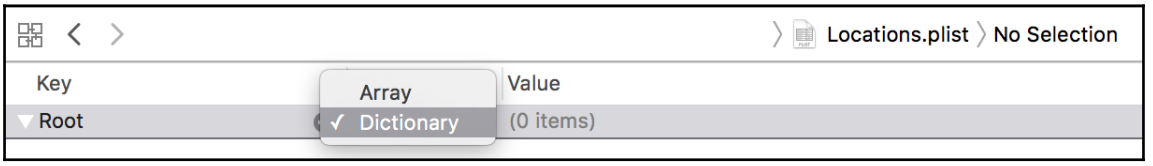

Carrier
3:55 PM
Cancel
Done
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
A cell
iPhone 7 - iOS 11.1

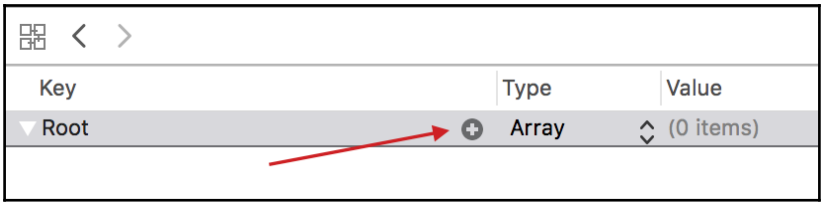
```swift
import UIKit

class LocationViewController: UIViewController, UITableViewDataSource {

    @IBOutlet weak var tableView:UITableView!

    let locations = ["Aspen", "Boston", "Charleston", "Chicago", "Houston", "Las Vegas", "Los Angeles", "Miami", "New Orleans",
        "New York", "Philadelphia", "Portland", "San Antonio", "San Francisco", "Washington District of Columbia"]

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return 15
    }

    func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "locationCell", for: indexPath) as UITableViewCell
        cell.textLabel?.text = "A cell"

        return cell
    }
}
```


Carrier
3:57 PM
Cancel
Done
Aspen
Boston
Charleston
Chicago
Houston
Las Vegas
Los Angeles
Miami
New Orleans
New York
Philadelphia
Portland
San Antonio
San Francisco
iPhone 7 - iOS 11.1


Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Property
Resource
PLIST
Property List
Cancel
Previous
Next


Locations.plist
No Selection
Key
Type
Value
Root
Dictionary
(0 items)


Locations.plist
No Selection
Key
Value
Root
Array
Dictionary
(0 items)



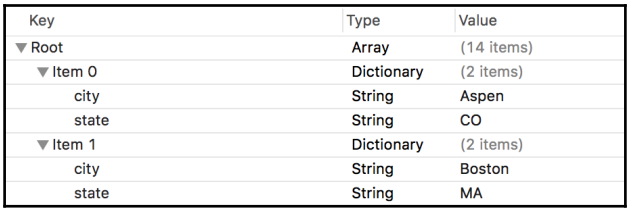
Key	Type	Value
▼ Root	Array	(1 item)
▼ Item 0	Dictionary	(0 items)

Key	Type	Value
▼ Root	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
state	String	CO

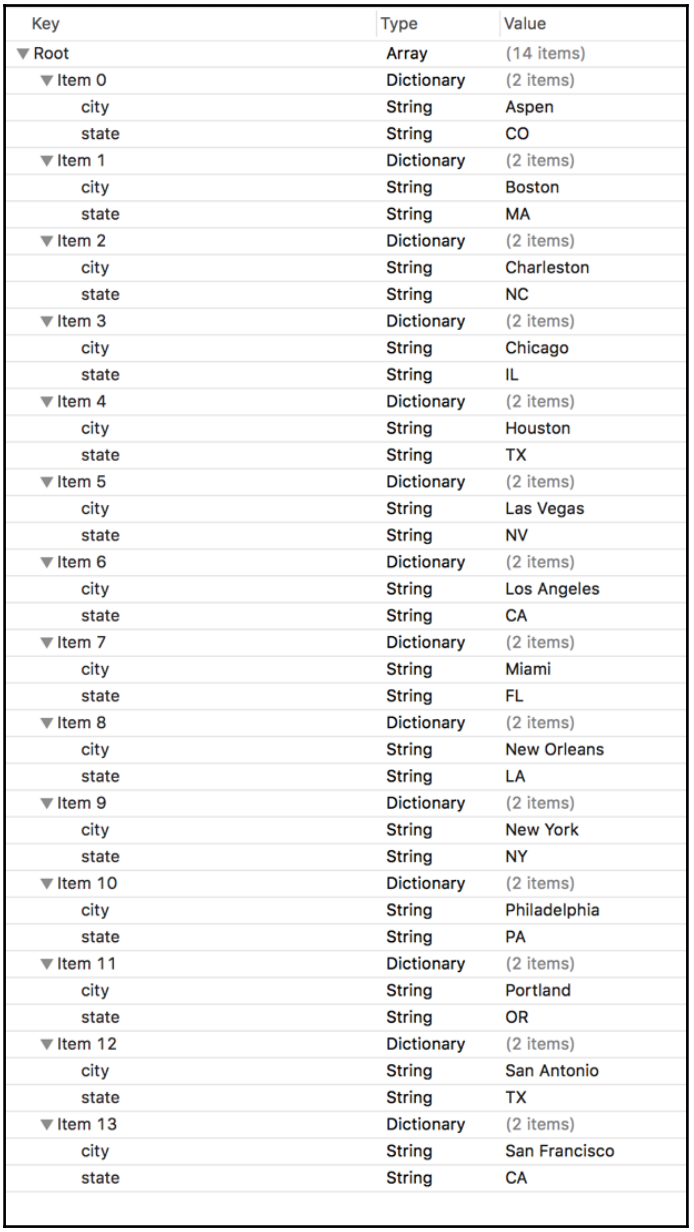
Key	Type	Value
▼ Root	Array	(1 item)
▼ Item 0	Dictionary	(2 items)
state	String	CO
city	String	Aspen

Key	Type	Value
▼ Root	Array	(1 item)
▶ Item 0	Dictionary	(2 items)

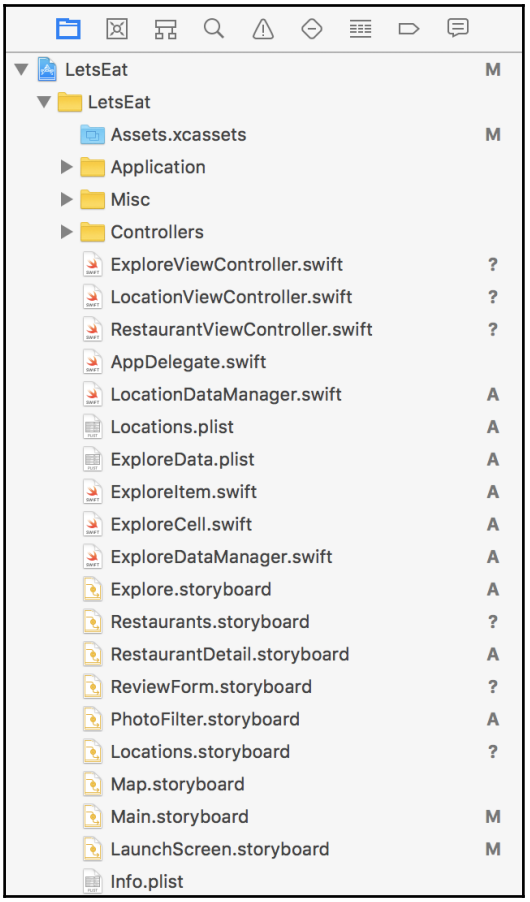
Key	Type	Value
▼ Root	Array	(2 items)
▶ Item 0	Dictionary	(2 items)
▶ Item 1	Dictionary	(2 items)

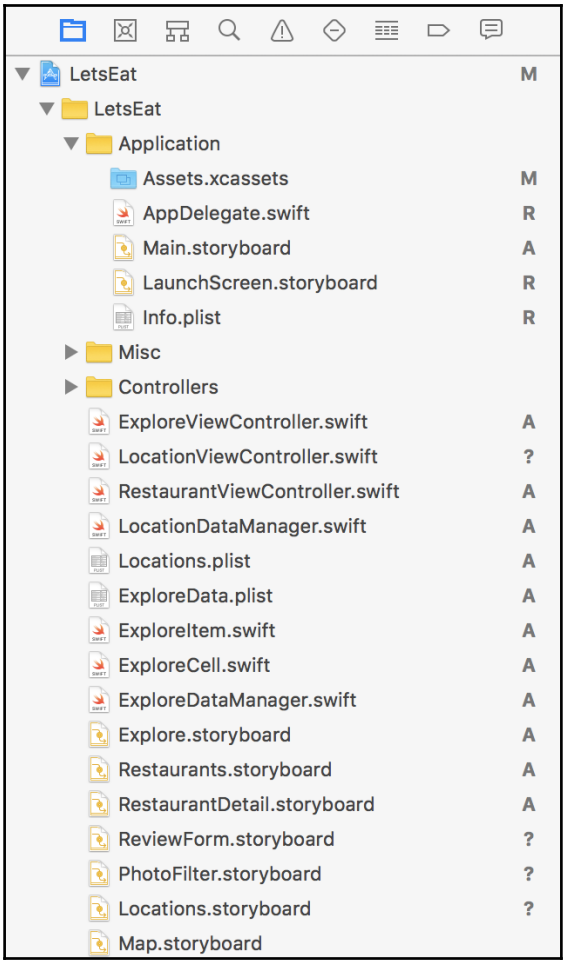


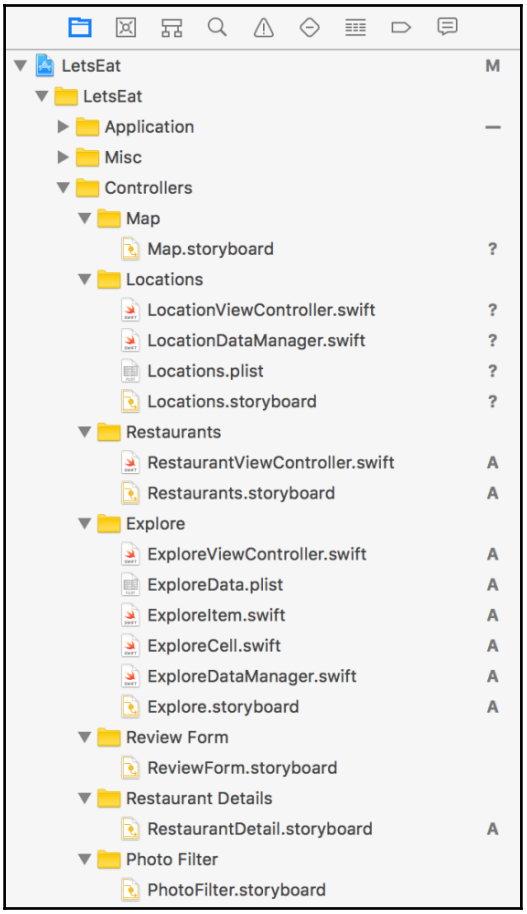
Key	Type	Value
▼ Root	Array	(14 items)
▼ Item 0	Dictionary	(2 items)
city	String	Aspen
state	String	CO
▼ Item 1	Dictionary	(2 items)
city	String	Boston
state	String	MA



Key	Type	Value
▼ Root	Array	(14 items)
▼ Item 0	Dictionary	(2 items)
city	String	Aspen
state	String	CO
▼ Item 1	Dictionary	(2 items)
city	String	Boston
state	String	MA
▼ Item 2	Dictionary	(2 items)
city	String	Charleston
state	String	NC
▼ Item 3	Dictionary	(2 items)
city	String	Chicago
state	String	IL
▼ Item 4	Dictionary	(2 items)
city	String	Houston
state	String	TX
▼ Item 5	Dictionary	(2 items)
city	String	Las Vegas
state	String	NV
▼ Item 6	Dictionary	(2 items)
city	String	Los Angeles
state	String	CA
▼ Item 7	Dictionary	(2 items)
city	String	Miami
state	String	FL
▼ Item 8	Dictionary	(2 items)
city	String	New Orleans
state	String	LA
▼ Item 9	Dictionary	(2 items)
city	String	New York
state	String	NY
▼ Item 10	Dictionary	(2 items)
city	String	Philadelphia
state	String	PA
▼ Item 11	Dictionary	(2 items)
city	String	Portland
state	String	OR
▼ Item 12	Dictionary	(2 items)
city	String	San Antonio
state	String	TX
▼ Item 13	Dictionary	(2 items)
city	String	San Francisco
state	String	CA


LetsEat M
LetsEat
Assets.xcassets M
Application
Misc
Controllers
ExploreViewController.swift ?
LocationViewController.swift ?
RestaurantViewController.swift ?
AppDelegate.swift
LocationDataManager.swift A
Locations.plist A
ExploreData.plist A
ExploreItem.swift A
ExploreCell.swift A
ExploreDataManager.swift A
Explore.storyboard A
Restaurants.storyboard ?
RestaurantDetail.storyboard A
ReviewForm.storyboard ?
PhotoFilter.storyboard A
Locations.storyboard ?
Map.storyboard
Main.storyboard M
LaunchScreen.storyboard M
Info.plist


LetsEat M
LetsEat
Application
Assets.xcassets M
AppDelegate.swift R
Main.storyboard A
LaunchScreen.storyboard R
Info.plist R
Misc
Controllers
ExploreViewController.swift A
LocationViewController.swift ?
RestaurantViewController.swift A
LocationDataManager.swift A
Locations.plist A
ExploreData.plist A
ExploreItem.swift A
ExploreCell.swift A
ExploreDataManager.swift A
Explore.storyboard A
Restaurants.storyboard A
RestaurantDetail.storyboard A
ReviewForm.storyboard ?
PhotoFilter.storyboard ?
Locations.storyboard ?
Map.storyboard


LetsEat M
LetsEat
Application –
Misc
Controllers
Map
Map.storyboard ?
Locations
LocationViewController.swift ?
LocationDataManager.swift ?
Locations.plist ?
Locations.storyboard ?
Restaurants
RestaurantViewController.swift A
Restaurants.storyboard A
Explore
ExploreViewController.swift A
ExploreData.plist A
ExploreItem.swift A
ExploreCell.swift A
ExploreDataManager.swift A
Explore.storyboard A
Review Form
ReviewForm.storyboard
Restaurant Details
RestaurantDetail.storyboard A
Photo Filter
PhotoFilter.storyboard

Chapter 16: Where Are We?

Key	Type	Value
▼ Root	Array	(5 items)
▼ Item 0	Dictionary	(16 items)
address	String	108 West 2nd Street #104
area	String	Los Angeles / Orange County
city	String	Los Angeles
▼ cuisines	Array	(2 items)
Item 0	String	Indian
Item 1	String	Gastropubs
country	String	US
id	Number	104,173
image_url	String	https://www.opentable.com/img/restimages/104173.jpg
lat	Number	34.051061
lng	Number	-118.244705
mobile_reserve_url	String	http://mobile.opentable.com/opentable/?restId=104173
name	String	Badmaash
phone	String	2132217466x
postal_code	String	90012
price	Number	2
reserve_url	String	http://www.opentable.com/single.aspx?rid=104173
state	String	CA
▶ Item 1	Dictionary	(15 items)
▶ Item 2	Dictionary	(16 items)
▶ Item 3	Dictionary	(16 items)
▶ Item 4	Dictionary	(16 items)

```
import Foundation
import MapKit

class RestaurantItem: NSObject, MKAnnotation {   Type 'RestaurantItem' does not conform to protocol 'MKAnnotation'
    var name: String?
    var cuisines:[String] = []
    var latitude: Double?
    var longitude:Double?
    var address:String?
    var postalCode:String?
    var state:String?
    var imageURL:String?

    init(dict:[String:AnyObject]) {
        if let lat = dict["lat"] as? Double { self.latitude = lat }
        if let long = dict["lng"] as? Double { self.longitude = long }
        if let name = dict["name"] as? String { self.name = name }
        if let cuisines = dict["cuisines"] as? [String] { self.cuisines = cuisines }
        if let address = dict["address"] as? String { self.address = address }
        if let postalCode = dict["postal_code"] as? String { self.postalCode = postalCode }
        if let state = dict["state"] as? String { self.state = state }
        if let image = dict["image_url"] as? String { self.imageURL = image }
    }
}
```

```
import Foundation
import MapKit

class RestaurantItem: NSObject, MKAnnotation {
    var name: String?
    var cuisines:[String] = []
    var latitude: Double?
    var longitude:Double?
    var address:String?
    var postalCode:String?
    var state:String?
    var imageURL:String?

    init(dict:[String:AnyObject]) {
        if let lat = dict["lat"] as? Double { self.latitude = lat }
        if let long = dict["lng"] as? Double { self.longitude = long }
        if let name = dict["name"] as? String { self.name = name }
        if let cuisines = dict["cuisines"] as? [String] { self.cuisines = cuisines }
        if let address = dict["address"] as? String { self.address = address }
        if let postalCode = dict["postal_code"] as? String { self.postalCode = postalCode }
        if let state = dict["state"] as? String { self.state = state }
        if let image = dict["image_url"] as? String { self.imageURL = image }
    }

    var title: String? {
        return name
    }

    var subtitle: String? {
        if cuisines.isEmpty { return "" }
        else if cuisines.count == 1 { return cuisines.first }
        else { return cuisines.joined(separator: ", ") }
    }

    var coordinate: CLLocationCoordinate2D {
        guard let lat = latitude, let long = longitude else { return
            CLLocationCoordinate2D() }
        return CLLocationCoordinate2D(latitude: lat, longitude: long )
    }
}
```

```
import Foundation

class MapDataManager {

    fileprivate var items:[RestaurantItem] = []

    var annotations:[RestaurantItem] {
        return items
    }

    func fetch(completion:(_ annotations:[RestaurantItem]) -> ()) {
        if items.count > 0 { items.removeAll() }

        for data in loadData() {
            items.append(RestaurantItem(dict: data))
        }
        completion(items)
    }

    fileprivate func loadData() -> [[String:AnyObject]] {
        guard let path = Bundle.main.path(forResource: "MapLocations", ofType: "plist"),
            let items = NSArray(contentsOfFile: path) else { return [[:]] }

        return items as! [[String: AnyObject]]
    }
}
```

Key	Type	Value
▼ Root	Array	(5 items)
▶ Item 0	Dictionary	(16 items)
▶ Item 1	Dictionary	(16 items)
▶ Item 2	Dictionary	(16 items)
▶ Item 3	Dictionary	(16 items)
▶ Item 4	Dictionary	(16 items)

```
import Foundation

protocol DataManager {
    func load(file name:String) -> [[String:AnyObject]]
}

extension DataManager {
    func load(file name:String) -> [[String:AnyObject]] {
        guard let path = Bundle.main.path(forResource: name, ofType: "plist"),
            let items = NSArray(contentsOfFile: path) else { return [[:]] }

        return items as! [[String : AnyObject]]
    }
}
```

```
import Foundation

class MapDataManager {

    fileprivate var items:[RestaurantItem] = []

    var annotations:[RestaurantItem] {
        return items
    }


    func fetch(completion:(_ annotations:[RestaurantItem]) -> ()) {
        if items.count > 0 { items.removeAll() }
        for data in loadData() {
            items.append(RestaurantItem(dict: data))
        }
        completion(items)
    }

    fileprivate func loadData() -> [[String:AnyObject]] {
        guard let path = Bundle.main.path(forResource: "MapLocations", ofType: "plist"), let items =
            NSArray(contentsOfFile: path) else { return [[:]] }

        return items as! [[String : AnyObject]]
    }
}
```

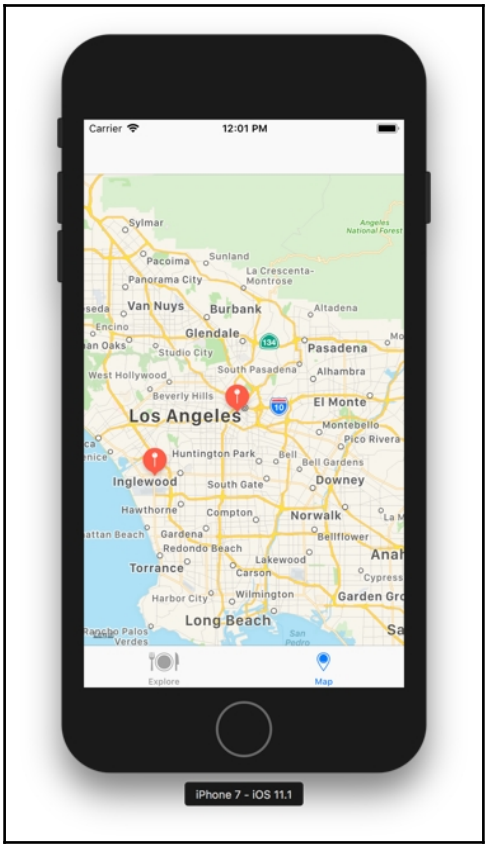
```
func currentRegion(latDelta:CLLocationDegrees, longDelta:CLLocationDegrees) -> MKCoordinateRegion {
    guard let item = items.first else { return MKCoordinateRegion() }
    let span = MKCoordinateSpanMake(latDelta, longDelta)

    return MKCoordinateRegion(center: item.coordinate, span: span)
}
```

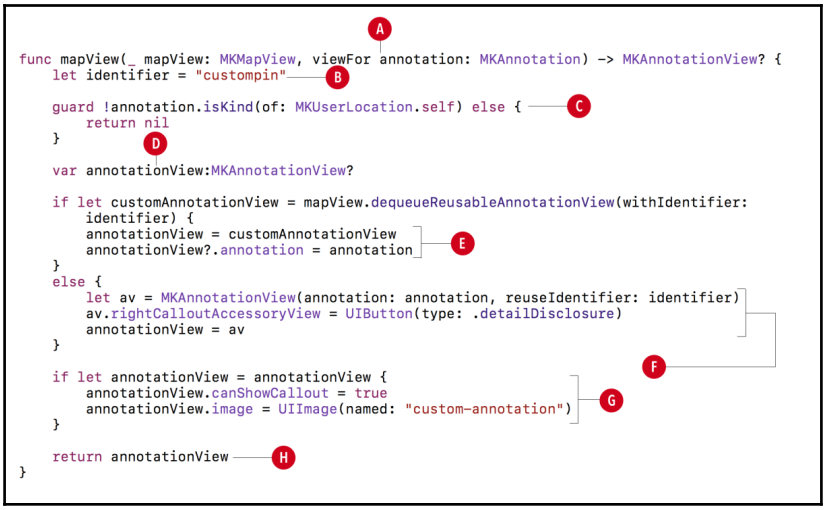
A

B

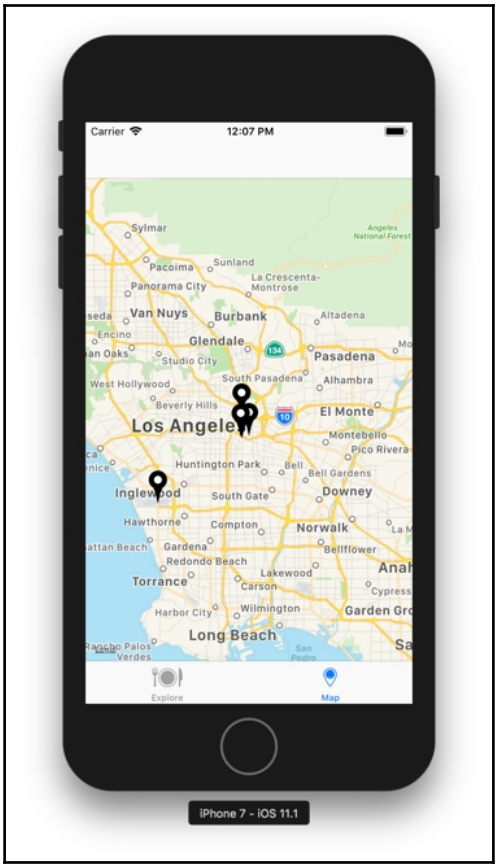
C

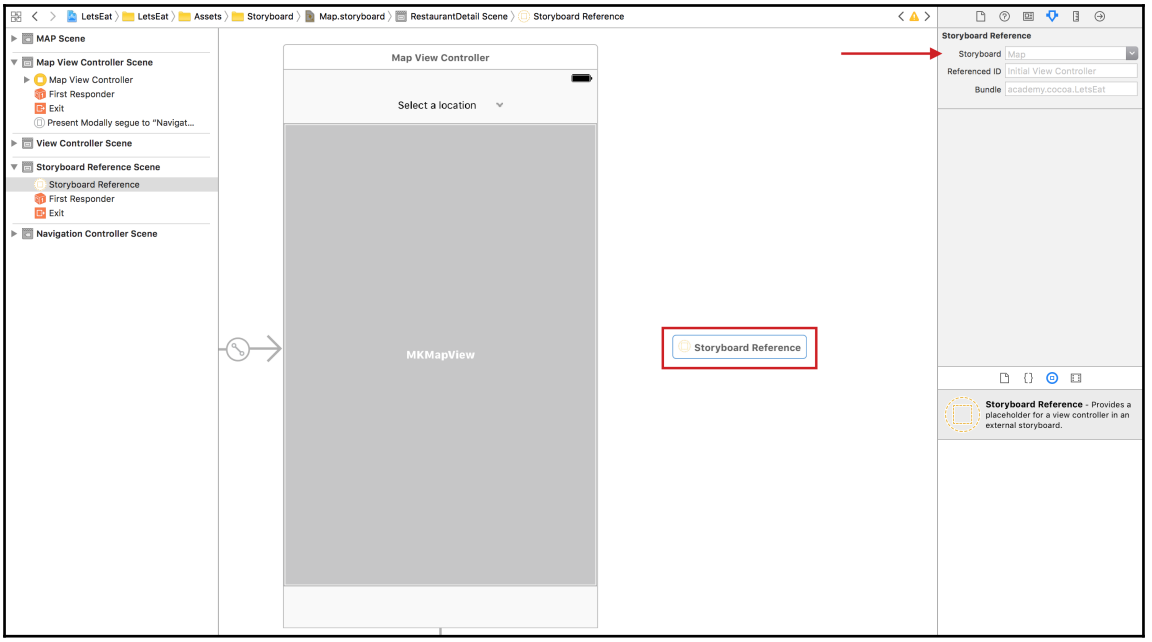
D

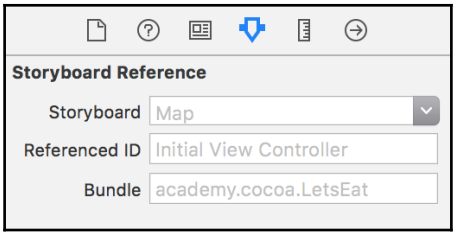

Carrier
12:01 PM
Sylmar
Pacoima
Sunland
La Crescenta-
Montrose
Panorama City
Van Nuys
Burbank
Altadena
Glendale
Pasadena
Studio City
West Hollywood
South Pasadena
Alhambra
Beverly Hills
El Monte
Los Angeles
Montebello
Pico Rivera
Huntington Park
Bell
Bell Gardens
Inglewood
South Gate
Downey
Hawthorne
Compton
Norwalk
Gardena
Bellflower
Redondo Beach
Torrance
Lakewood
Carson
Harbor City
Wilmington
Long Beach
Explore
Map
iPhone 7 - iOS 11.1

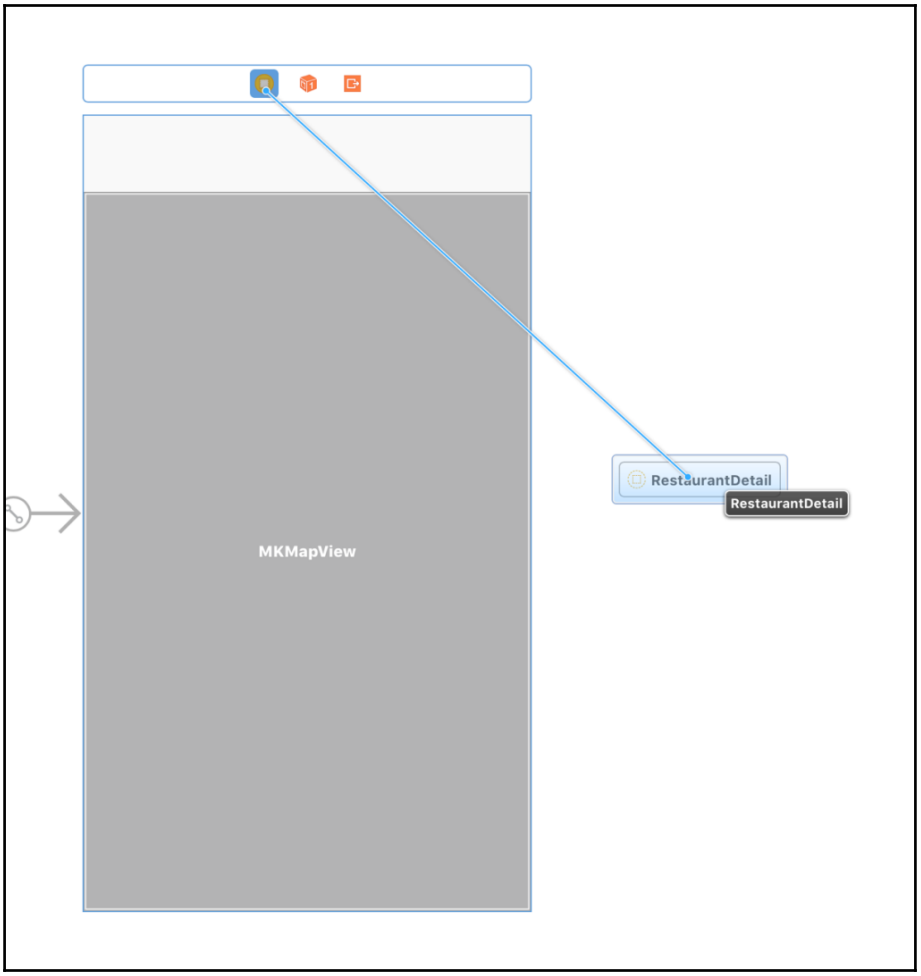


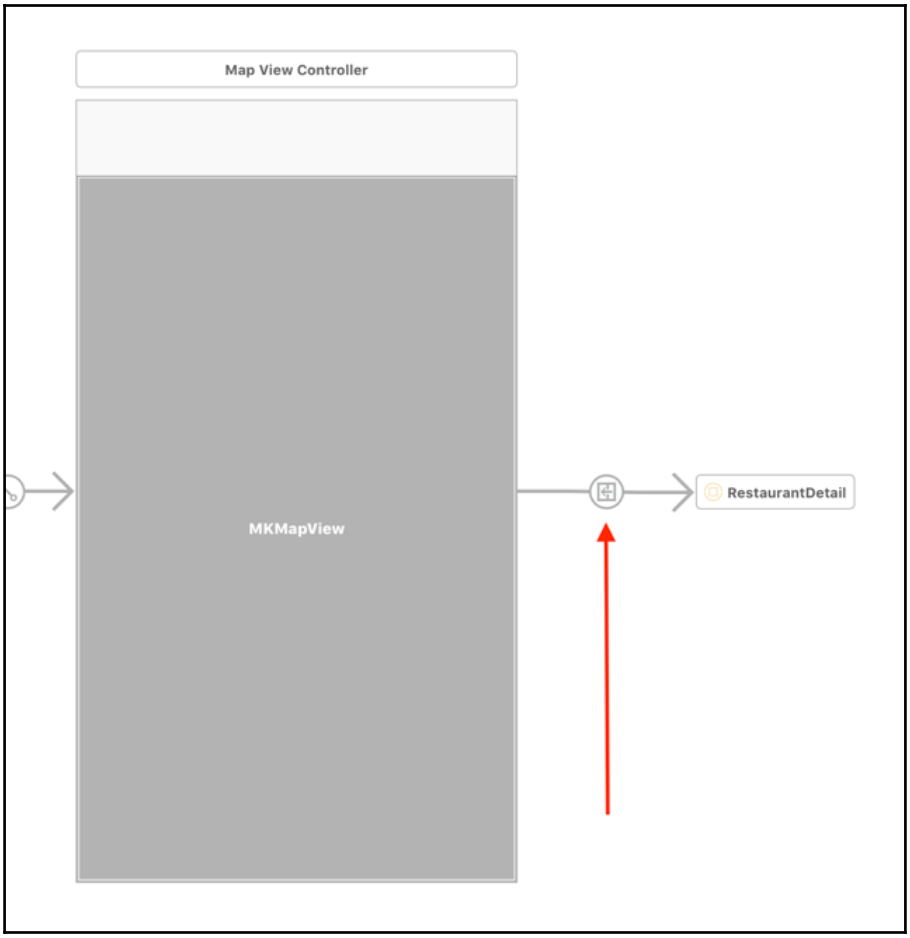

```
func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
    let identifier = "custompin"

    guard !annotation.isKind(of: MKUserLocation.self) else {
        return nil
    }

    var annotationView:MKAnnotationView?

    if let customAnnotationView = mapView.dequeueReusableAnnotationView(withIdentifier:
        identifier) {
        annotationView = customAnnotationView
        annotationView?.annotation = annotation
    }
    else {
        let av = MKAnnotationView(annotation: annotation, reuseIdentifier: identifier)
        av.rightCalloutAccessoryView = UIButton(type: .detailDisclosure)
        annotationView = av
    }

    if let annotationView = annotationView {
        annotationView.canShowCallout = true
        annotationView.image = UIImage(named: "custom-annotation")
    }

    return annotationView
}
```

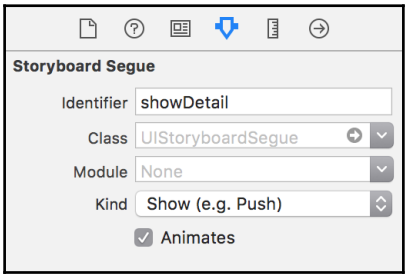

Carrier
12:07 PM
Sylmar
Pacoima
Sunland
La Crescenta-
Montrose
Panorama City
Van Nuys
Burbank
Altadena
Glendale
Pasadena
Studio City
West Hollywood
South Pasadena
Alhambra
Beverly Hills
El Monte
Los Angeles
Montebello
Pico Rivera
Huntington Park
Bell
Bell Gardens
Inglewood
South Gate
Downey
Hawthorne
Compton
Norwalk
Gardena
Bellflower
Redondo Beach
Lakewood
Torrance
Carson
Harbor City
Wilmington
Long Beach
Explore
Map
iPhone 7 - iOS 11.1


LetsEat
LetsEat
Assets
Storyboard
Map.storyboard
RestaurantDetail Scene
Storyboard Reference
MAP Scene
Map View Controller Scene
Map View Controller
First Responder
Exit
Present Modally segue to "Navigat...
View Controller Scene
Storyboard Reference Scene
Storyboard Reference
First Responder
Exit
Navigation Controller Scene
Map View Controller
Select a location
MKMapView
Storyboard Reference
Storyboard Reference
Storyboard Map
Referenced ID Initial View Controller
Bundle academy.cocoa.LetsEat
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.


Storyboard Reference
Storyboard Map
Referenced ID Initial View Controller
Bundle academy.cocoa.LetsEat


MKMapView
RestaurantDetail
RestaurantDetail


Map View Controller
MKMapView
RestaurantDetail


Storyboard Segue
Identifier
showDetail
Class
UIStoryboardSegue
Module
None
Kind
Show (e.g. Push)
Animates

```
    mapView.delegate = self

    manager.fetch { (annotations) in
        addMap(annotations)
    }
}

func addMap(_ annotations:[RestaurantAnnotation]) {
    mapView.setRegion(manager.currentRegion(latDelta: 0.5, longDelta: 0.5), animated: true)
    mapView.addAnnotations(annotations)
}

func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView,
    calloutAccessoryControlTapped control: UIControl) {
    self.performSegue(withIdentifier: Segue.showDetail.rawValue, sender: self)
}

func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
    let identifier = "custompin"

    guard !annotation.isKind(of: MKUserLocation.self) else {
        return nil
    }

    var annotationView:MKAnnotationView?
```

```
//
//  RestaurantDetailViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 11/15/16.
//  Copyright © 2016 Craig Clayton. All rights reserved.
//

import UIKit

class RestaurantDetailViewController: UITableViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

    }
}
```

```
//
//  RestaurantDetailViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 11/12/17.
//  Copyright © 2017 Cocoa Academy. All rights reserved.
//

import UIKit

class RestaurantDetailViewController: UITableViewController {

    var selectedRestaurant:RestaurantItem?

    override func viewDidLoad() {
        super.viewDidLoad()
        dump(selectedRestaurant as Any)
    }
}
```

```
func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
    guard let annotation = mapView.selectedAnnotations.first else { return }
    selectedRestaurant = annotation as? RestaurantItem

    self.performSegue(withIdentifier: Segue.showDetail.rawValue, sender: self)
}
```

```
override func viewDidLoad() {
    super.viewDidLoad()

    initialize()
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    switch segue.identifier! {
    case Segue.showDetail.rawValue:
        showRestaurantDetail(segue: segue)
    default:
        print("Segue not added")
    }
}

func initialize() {
    mapView.delegate = self

    manager.fetch { (annotations) in
        addMap(annotations)
    }
}

func addMap(_ annotations:[RestaurantAnnotation]) {
    mapView.setRegion(manager.currentRegion(latDelta: 0.5, longDelta: 0.5), animated: true)
    mapView.addAnnotations(annotations)
}

func showRestaurantDetail(segue:UIStoryboardSegue) {
    if let viewController = segue.destination as? RestaurantDetailViewController, let restaurant = selectedRestaurant  {
        viewController.selectedRestaurant = restaurant
    }
}
```

```
2017-11-12 12:32:42.020113-0500 LetsEat[11094:1811464] Could not inset legal attribution from corner 4
▿ Optional(<LetsEat.RestaurantItem: 0x608000391c60>)
  ▿ some: <LetsEat.RestaurantItem: 0x608000391c60> #0
    - super: NSObject
    ▿ name: Optional("Maria\'s Italian Kitchen - Downtown")
      - some: "Maria\'s Italian Kitchen - Downtown"
    ▿ cuisines: 2 elements
      - "Indian"
      - "Gastropubs"
    ▿ latitude: Optional(34.04934200000001)
      - some: 34.04934200000001
    ▿ longitude: Optional(-118.258174)
      - some: -118.258174
    ▿ address: Optional("615 S. Flower Street")
      - some: "615 S. Flower Street"
    ▿ postalCode: Optional("90017")
      - some: "90017"
    ▿ state: Optional("CA")
      - some: "CA"
    ▿ imageURL: Optional("https://www.opentable.com/img/restimages/19183.jpg")
      - some: "https://www.opentable.com/img/restimages/19183.jpg"
2017-11-12 12:32:45.572309-0500 LetsEat[11094:1811464] [Warning] Warning once only: Detected a case where constraints ambiguously suggest
a height of zero for a tableview cell's content view. We're considering the collapse unintentional and using standard height instead.
```

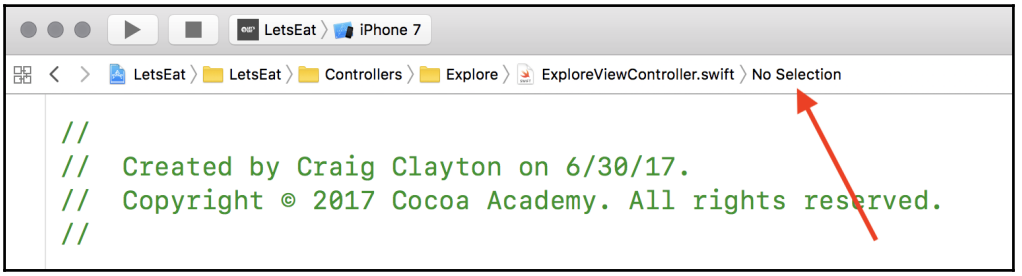
```
class ExploreViewController: UIViewController, UICollectionViewDataSource {

    @IBOutlet weak var collectionView:UICollectionView!

    let manager = ExploreDataManager()

    override func viewDidLoad() {
        super.viewDidLoad()
```

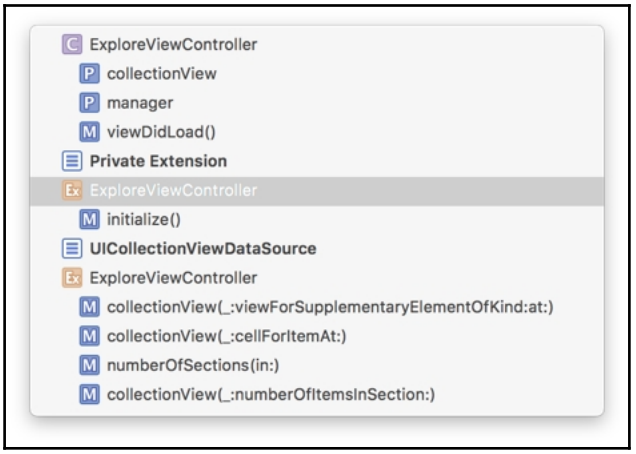
delete

```swift
import UIKit

class ExploreViewController: UIViewController {

    @IBOutlet weak var collectionView:UICollectionView!

    let manager = ExploreDataManager()

    override func viewDidLoad() {
        super.viewDidLoad()

        manager.fetch()
    }

    func collectionView(_ collectionView: UICollectionView, viewForSupplementaryElementOfKind kind: String, at indexPath: IndexPath) -> UICollectionReusableView {
        let headerView = collectionView.dequeueReusableSupplementaryView(ofKind: kind, withReuseIdentifier: "header", for: indexPath)
        return headerView
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "exploreCell", for: indexPath) as! ExploreCell

        let item = manager.explore(at: indexPath)
        if let name = item.name { cell.lblName.text = name }
        if let image = item.image { cell.imgExplore.image = UIImage(named: image) }

        return cell
    }

    func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return manager.numberOfItems()
    }

    @IBAction func unwindLocationCancel(segue:UIStoryboardSegue) {}
}

// MARK: Private Extension
private extension ExploreViewController {
    // code goes here
}

// MARK: UICollectionViewDataSource
extension ExploreViewController: UICollectionViewDataSource {
    // code goes here
}
```

Move all the code marked above to here

```swift
import UIKit

class ExploreViewController: UIViewController {

    @IBOutlet weak var collectionView:UICollectionView!

    let manager = ExploreDataManager()

    override func viewDidLoad() {
        super.viewDidLoad()

        manager.fetch()
    }


    @IBAction func unwindLocationCancel(segue:UIStoryboardSegue) {}
}

// MARK: Private Extension
private extension ExploreViewController {
    // code goes here
}

// MARK: UICollectionViewDataSource
extension ExploreViewController: UICollectionViewDataSource {
    func collectionView(_ collectionView: UICollectionView, viewForSupplementaryElementOfKind kind: String, at indexPath: IndexPath) -> UICollectionReusableView {
        let headerView = collectionView.dequeueReusableSupplementaryView(ofKind: kind, withReuseIdentifier: "header", for: indexPath)
        return headerView
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "exploreCell", for: indexPath) as! ExploreCell

        let item = manager.explore(at: indexPath)
        if let name = item.name { cell.lblName.text = name }
        if let image = item.image { cell.imgExplore.image = UIImage(named: image) }

        return cell
    }

    func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return manager.numberOfItems()
    }
}
```






```swift
import UIKit

class RestaurantViewController: UIViewController, UICollectionViewDataSource {

    @IBOutlet var collectionView:UICollectionView!

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        return collectionView.dequeueReusableCell(withReuseIdentifier: "restaurantCell", for: indexPath)
    }

    func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return 10
    }
}

// MARK: Private Extension
private extension RestaurantListViewController {

}

// MARK: UICollectionViewDataSource
extension RestaurantListViewController: UICollectionViewDataSource {

}
```

Move all the code marked above to here

```swift
import UIKit

class RestaurantViewController: UIViewController {

    @IBOutlet var collectionView:UICollectionView!

    override func viewDidLoad() {
        super.viewDidLoad()
    }
}

// MARK: Private Extension
private extension RestaurantViewController {

}

// MARK: UICollectionViewDataSource
extension RestaurantViewController: UICollectionViewDataSource {
    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        return collectionView.dequeueReusableCell(withReuseIdentifier: "restaurantCell", for: indexPath)
    }

    func numberOfSections(in collectionView: UICollectionView) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return 10
    }
}
```

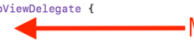
```swift
class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!

    let manager = LocationDataManager()

    override func viewDidLoad() {
        super.viewDidLoad()
        manager.fetch()
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return manager.numberOfItems()
    }

    func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "locationCell", for: indexPath) as UITableViewCell
        cell.textLabel?.text = manager.locationItem(at:indexPath)

        return cell
    }
}

// MARK: Private Extension
private extension LocationViewController {


}

// MARK: UITableViewDataSource
extension LocationViewController: UITableViewDataSource {

}
```

Move all the code marked above to here

```
import UIKit

class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!

    let manager = LocationDataManager()

    override func viewDidLoad() {
        super.viewDidLoad()
        manager.fetch()
    }
}

// MARK: Private Extension
private extension LocationViewController {


}

// MARK: UITableViewDataSource
extension LocationViewController: UITableViewDataSource {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return manager.numberOfItems()
    }

    func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "locationCell", for: indexPath) as UITableViewCell
        cell.textLabel?.text = manager.locationItem(at:indexPath)

        return cell
    }
}
```

```
class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!

    let manager = LocationDataManager()

    override func viewDidLoad() {
        super.viewDidLoad()
        initialize()
    }
}

// MARK: Private Extension
private extension LocationViewController {
    func initialize() {
        manager.fetch()
    }
}
```

```swift
import UIKit
import MapKit

class MapViewController: UIViewController  {

    @IBOutlet var mapView: MKMapView!

    let manager = MapDataManager()
    var selectedRestaurant:RestaurantItem?

    override func viewDidLoad() {
        super.viewDidLoad()
        initialize()
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        switch segue.identifier! {
        case Segue.showDetail.rawValue:
            showRestaurantDetail(segue: segue)
        default:
            print("Segue not added")
        }
    }

    func initialize() {
        mapView.delegate = self
        manager.fetch { (annotations) in
            addMap(annotations)
        }
    }

    func addMap(_ annotations:[RestaurantItem]) {
        mapView.setRegion(manager.currentRegion(latDelta: 0.5, longDelta: 0.5), animated: true)
        mapView.addAnnotations(manager.annotations)
    }

    func showRestaurantDetail(segue:UIStoryboardSegue) {
        if let viewController = segue.destination as? RestaurantDetailViewController, let restaurant = selectedRestaurant  {
            viewController.selectedRestaurant = restaurant
        }
    }

    func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
        let identifier = "custompin"

        guard !annotation.isKind(of: MKUserLocation.self) else { return nil }
        var annotationView: MKAnnotationView?

        if let customAnnotationView = mapView.dequeueReusableAnnotationView(withIdentifier: identifier) {
            annotationView = customAnnotationView
            annotationView?.annotation = annotation
        }
        else {
            let av = MKAnnotationView(annotation: annotation, reuseIdentifier: identifier)
            av.rightCalloutAccessoryView = UIButton(type: .detailDisclosure)
            annotationView = av
        }

        if let annotationView = annotationView {
            annotationView.canShowCallout = true
            annotationView.image = UIImage(named: "custom-annotation")
        }

        return annotationView
    }

    func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
        guard let annotation = mapView.selectedAnnotations.first else { return }
        selectedRestaurant = annotation as? RestaurantItem

        self.performSegue(withIdentifier: Segue.showDetail.rawValue, sender: self)
    }
}

// MARK: Private Extension
private extension MapViewController {

}

// MARK: MKMapViewDelegate
extension MapViewController: MKMapViewDelegate {

}
```



```
// MARK: MKMapViewDelegate
extension MapViewController: MKMapViewDelegate {
    func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
        let identifier = "custompin"

        guard !annotation.isKind(of: MKUserLocation.self) else { return nil }
        var annotationView: MKAnnotationView?

        if let customAnnotationView = mapView.dequeueReusableAnnotationView(withIdentifier: identifier) {
            annotationView = customAnnotationView
            annotationView?.annotation = annotation
        }
        else {
            let av = MKAnnotationView(annotation: annotation, reuseIdentifier: identifier)
            av.rightCalloutAccessoryView = UIButton(type: .detailDisclosure)
            annotationView = av
        }

        if let annotationView = annotationView {
            annotationView.canShowCallout = true
            annotationView.image = UIImage(named: "custom-annotation")
        }

        return annotationView
    }

    func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
        guard let annotation = mapView.selectedAnnotations.first else { return }
        selectedRestaurant = annotation as? RestaurantItem

        self.performSegue(withIdentifier: Segue.showDetail.rawValue, sender: self)
    }
}
```

```
import UIKit
import MapKit

class MapViewController: UIViewController  {

    @IBOutlet var mapView: MKMapView!

    let manager = MapDataManager()
    var selectedRestaurant:RestaurantItem?

    override func viewDidLoad() {
        super.viewDidLoad()
        initialize()
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        switch segue.identifier! {
        case Segue.showDetail.rawValue:
            showRestaurantDetail(segue: segue)
        default:
            print("Segue not added")
        }
    }

    func initialize() {
        mapView.delegate = self
        manager.fetch { (annotations) in
            addMap(annotations)
        }
    }

    func addMap(_ annotations:[RestaurantItem]) {
        mapView.setRegion(manager.currentRegion(latDelta: 0.5, longDelta: 0.5), animated: true)
        mapView.addAnnotations(manager.annotations)
    }

    func showRestaurantDetail(segue:UIStoryboardSegue) {
        if let viewController = segue.destination as? RestaurantDetailViewController, let restaurant = selectedRestaurant  {
            viewController.selectedRestaurant = restaurant
        }
    }
}

// MARK: Private Extension
private extension MapViewController {

}
```

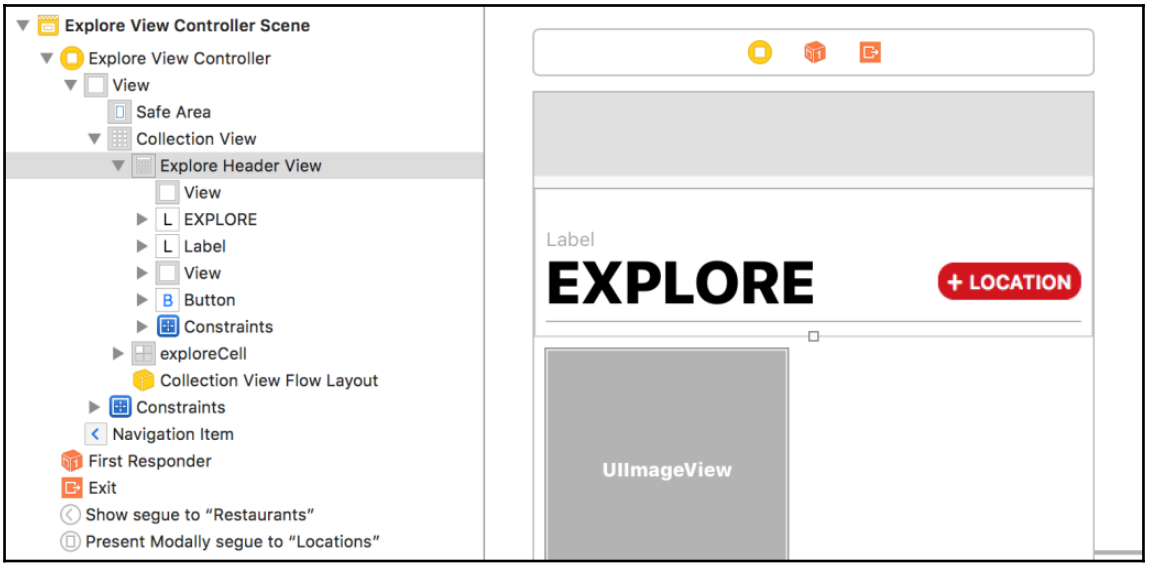
Move all the code marked above to here

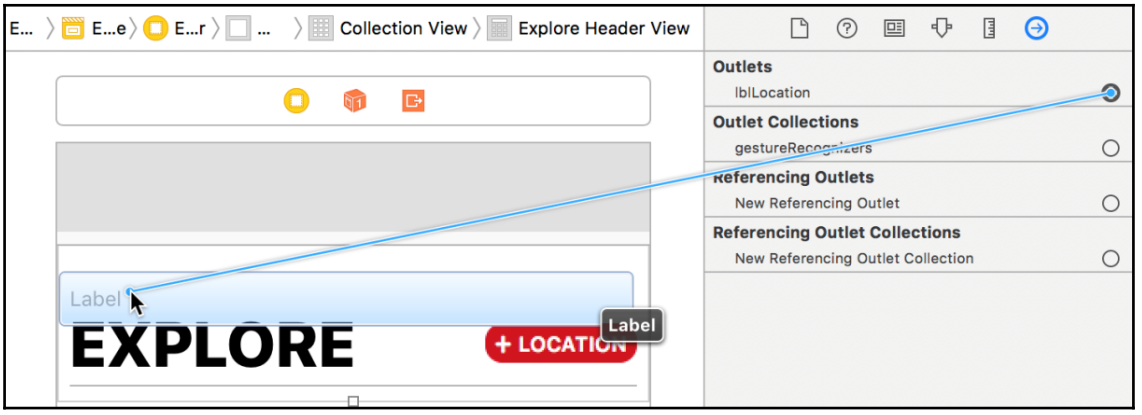
```
import UIKit
import MapKit

class MapViewController: UIViewController  {

    @IBOutlet var mapView: MKMapView!

    let manager = MapDataManager()
    var selectedRestaurant:RestaurantItem?

    override func viewDidLoad() {
        super.viewDidLoad()
        initialize()
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        switch segue.identifier! {
        case Segue.showDetail.rawValue:
            showRestaurantDetail(segue: segue)
        default:
            print("Segue not added")
        }
    }
}

// MARK: Private Extension
private extension MapViewController {
    func initialize() {
        mapView.delegate = self
        manager.fetch { (annotations) in
            addMap(annotations)
        }
    }

    func addMap(_ annotations:[RestaurantItem]) {
        mapView.setRegion(manager.currentRegion(latDelta: 0.5, longDelta: 0.5), animated: true)
        mapView.addAnnotations(manager.annotations)
    }

    func showRestaurantDetail(segue:UIStoryboardSegue) {
        if let viewController = segue.destination as? RestaurantDetailViewController, let restaurant = selectedRestaurant  {
            viewController.selectedRestaurant = restaurant
        }
    }
}
```

Chapter 17: Working with an API

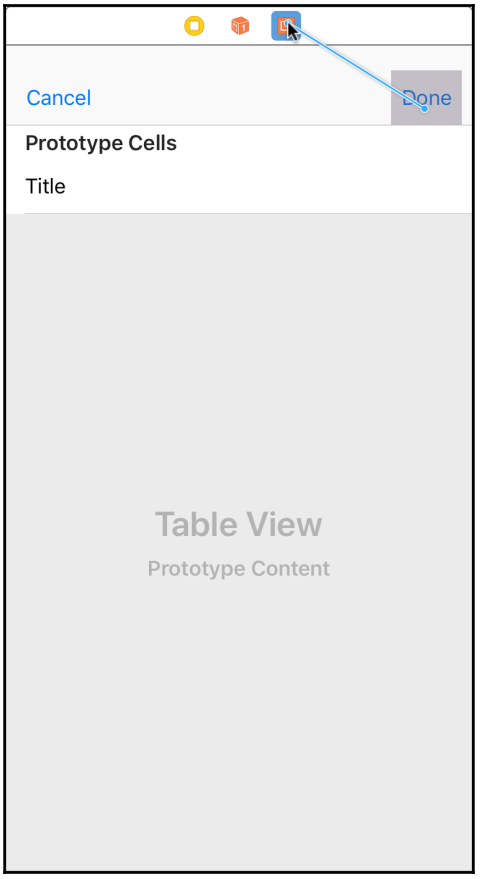

```
{
    "total_entries": 67,
    "per_page": 25,
    "current_page": 1,
    "restaurants": [
        {
            "id": 147475,
            "name": "Union Provisions",
            "address": "513 King Street",
            "city": "Charleston",
            "state": "SC",
            "area": "South Carolina",
            "postal_code": "29403",
            "country": "US",
            "phone": "8436410821x",
            "lat": 32.790291,
            "lng": -79.93936,
            "price": 2,
            "reserve_url": "http://www.opentable.com/single.aspx?rid=147475",
            "mobile_reserve_url": "http://mobile.opentable.com/opentable/?restId=147475",
            "image_url": "https://www.opentable.com/img/restimages/147475.jpg",
            "cuisines": [
                {
                    "cuisine": "American"
                },
                {
                    "cuisine": "Bar"
                }
            ]
        },
```

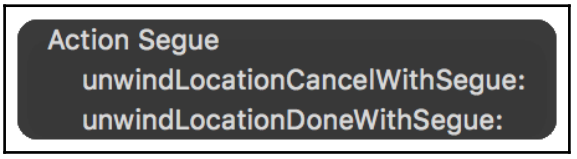

```
import Foundation

struct RestaurantAPIManager {
    static func loadJSON(file name:String) -> [[String:AnyObject]] {
        var items = [[String : AnyObject]]()

        guard let path = Bundle.main.path(forResource: name, ofType: "json"),
              let data = NSData(contentsOfFile: path) else {
            return [[:]]
        }

        do {
            let json = try JSONSerialization.jsonObject(with: data as Data, options: .allowFragments) as AnyObject
            if let restaurants = json["restaurants"] as? [[String: AnyObject]] {
                items = restaurants as [[String : AnyObject]]
            }
        }
        catch {
            print("error serializing JSON: \(error)")
            items = [[:]]
        }

        return items
    }
}
```

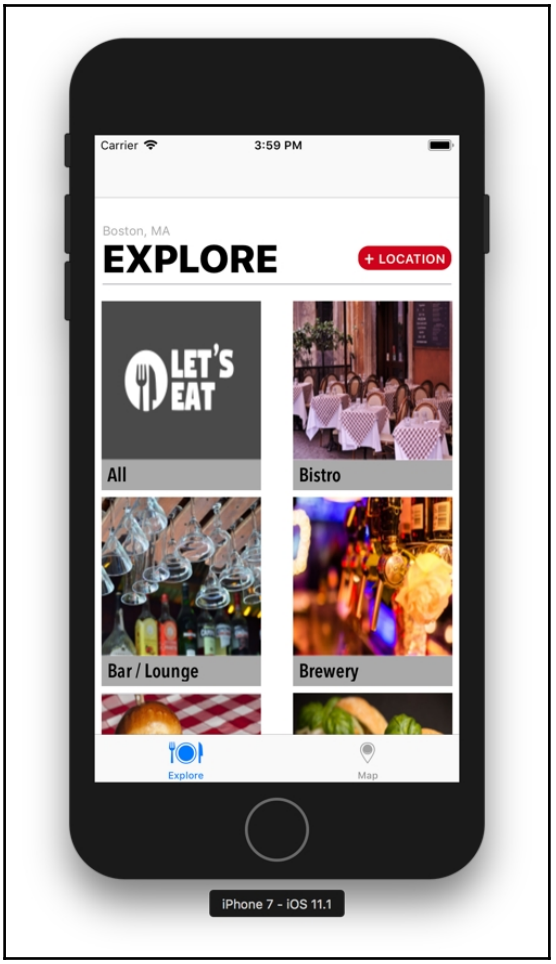

Explore View Controller Scene
Explore View Controller
View
Safe Area
Collection View
Explore Header View
View
EXPLORE
Label
View
Button
Constraints
exploreCell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Show segue to "Restaurants"
Present Modally segue to "Locations"
Label
EXPLORE
+ LOCATION
UIImageView

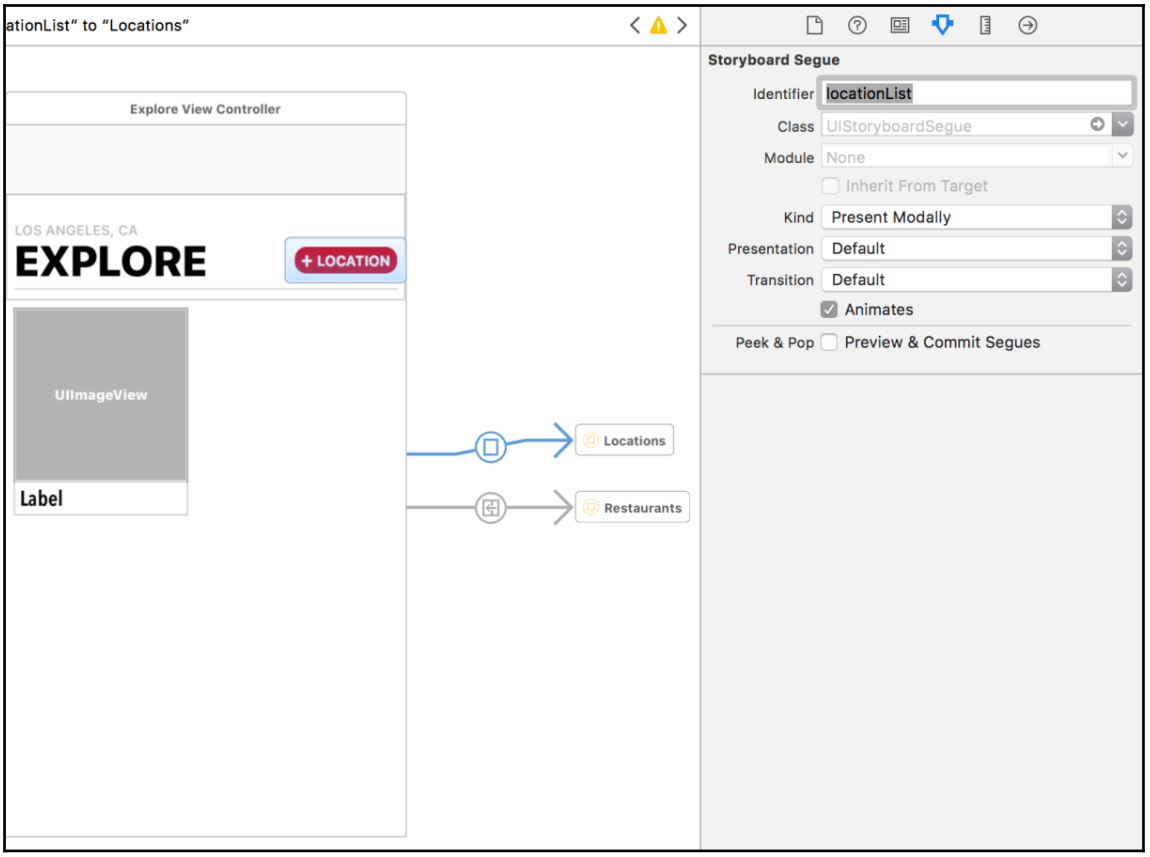

E...
E...e
E...r
...
Collection View
Explore Header View
Outlets
lblLocation
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label
EXPLORE
+ LOCATION
Label

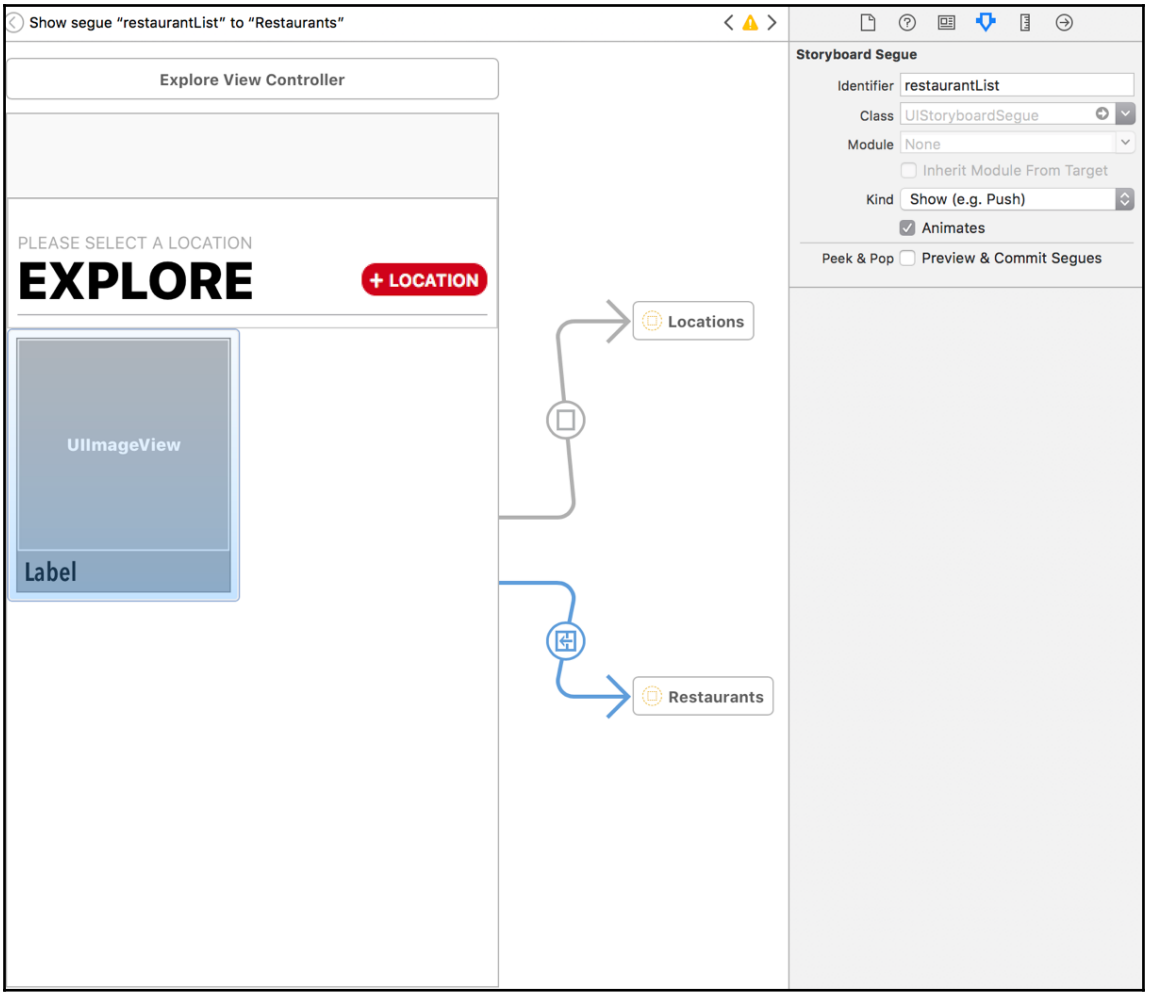

PLEASE SELECT A LOCATION
EXPLORE
+ LOCATION
UIImageView
Label


Cancel
Done
Prototype Cells
Title
Table View
Prototype Content

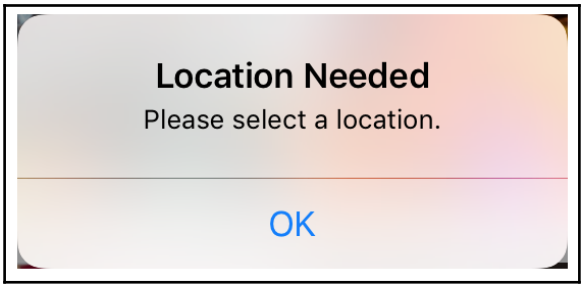

Action Segue
unwindLocationCancelWithSegue:
unwindLocationDoneWithSegue:


Carrier
3:59 PM
Boston, MA
EXPLORE
+ LOCATION
LET'S EAT
All
Bistro
Bar / Lounge
Brewery
Explore
Map
iPhone 7 - iOS 11.1


ationList" to "Locations"
Explore View Controller
LOS ANGELES, CA
EXPLORE
+ LOCATION
UIImageView
Label
Locations
Restaurants
Storyboard Segue
Identifier locationList
Class UIStoryboardSegue
Module None
Inherit From Target
Kind Present Modally
Presentation Default
Transition Default
Animates
Peek & Pop Preview & Commit Segues

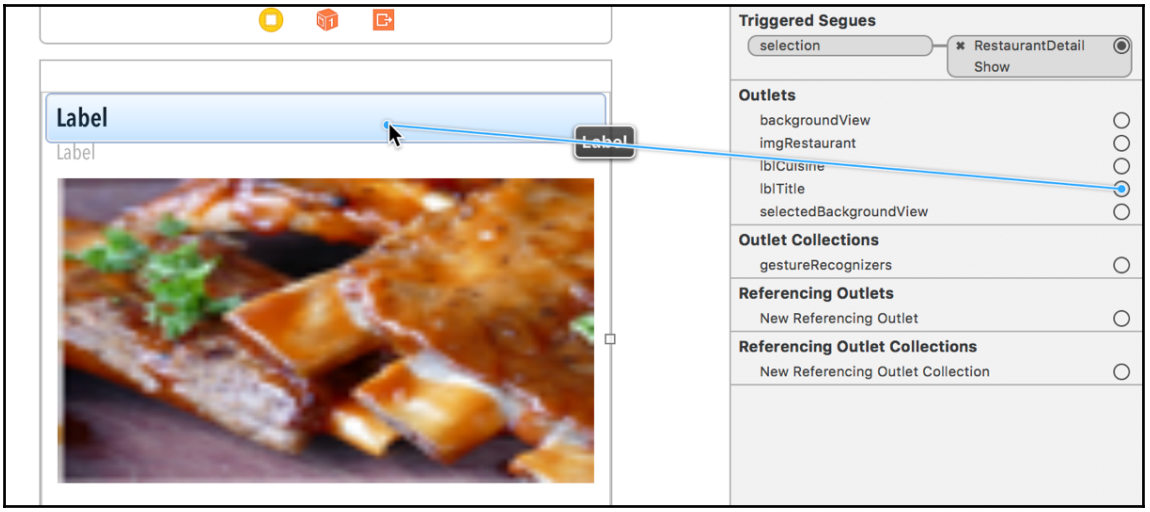


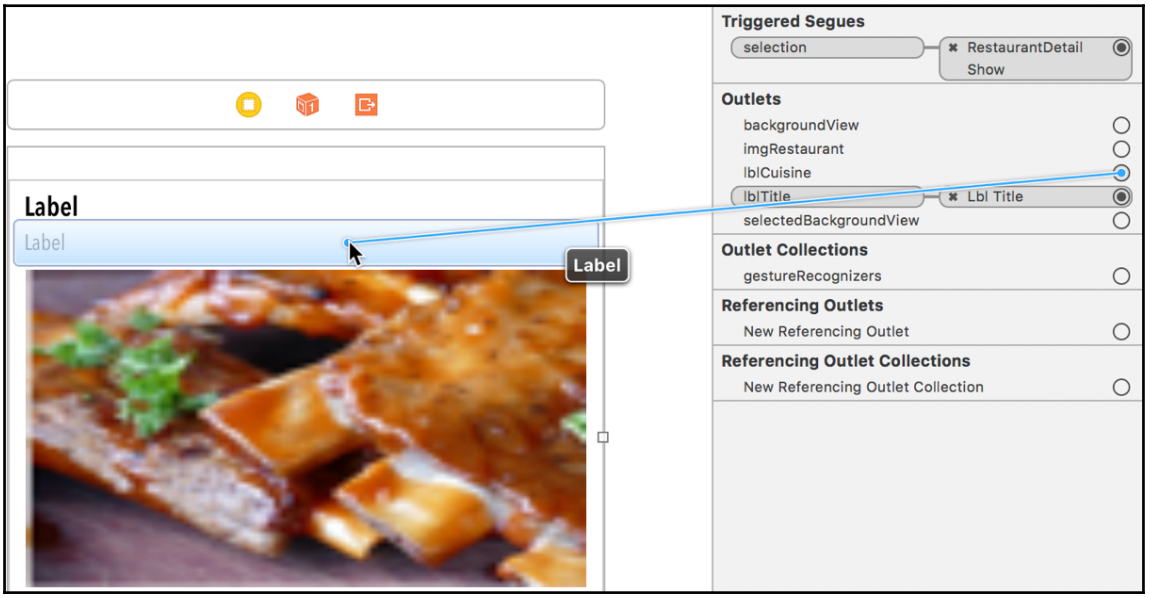

```
func set(selected cell:UITableViewCell, at indexPath:IndexPath) {
    if let city = selectedCity {
        let data = manager.findLocation(by: city)
        if data.isFound {
            if indexPath.row == data.position {
                cell.accessoryType = .checkmark
            }
            else { cell.accessoryType = .none }
        }
    }
    else { cell.accessoryType = .none }
}
```

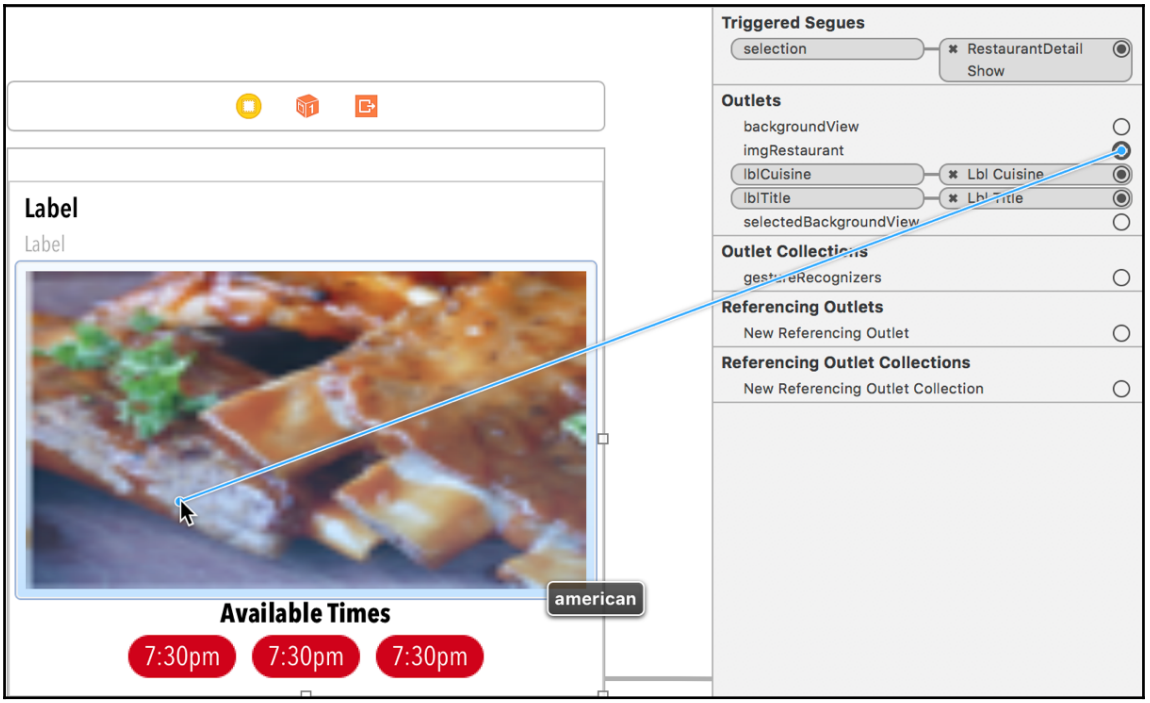



```
LetsEat
selected city Optional(LetsEat.LocationItem(state: Optional("NC"), city: Optional("Charleston")))
selected type Optional("Bistro")
```

```
LetsEat
type Bistro
[["state": SC, "city": Charleston, "country": US, "name": Union Provisions, "address": 513 King Street, "lat":
32.790291, "price": 2, "reserve_url": http://www.opentable.com/single.aspx?rid=147475, "long":
-79.93935999999999, "id": 147475, "phone": 8436410821x, "image_url": https://www.opentable.com/img/restimages/
147475.jpg, "mobile_reserve_url": http://mobile.opentable.com/opentable/?restId=147475, "area": South
Carolina, "postal_code": 29403, "cuisines": <__NSArrayI 0x608000232e80>(
{
    cuisine = Pizza;
},
{
    cuisine = Italian;
}
)
], ["state": SC, "city": Charleston, "country": US, "name": McCrady's, "address": 2 Unity Alley, "lat":
32.778, "price": 4, "reserve_url": http://www.opentable.com/single.aspx?rid=3751, "long": -79.92700000000001,
"id": 3751, "phone": 8435770025x1, "image_url": https://www.opentable.com/img/restimages/3751.jpg,
"mobile_reserve_url": http://mobile.opentable.com/opentable/?restId=3751, "area": South Carolina,
"postal_code": 29401, "cuisines": <__NSArrayI 0x6080002311c0>(
{
```


Label
Label
Label
Triggered Segues
selection
RestaurantDetail
Show
Outlets
backgroundView
imgRestaurant
lblCuisine
lblTitle
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

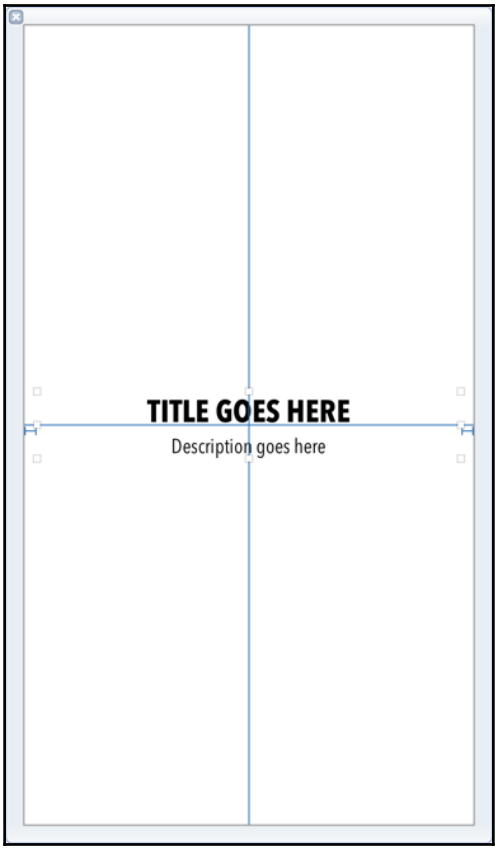

Label
Label
Label
Triggered Segues
selection
RestaurantDetail
Show
Outlets
backgroundView
imgRestaurant
lblCuisine
lblTitle
Lbl Title
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

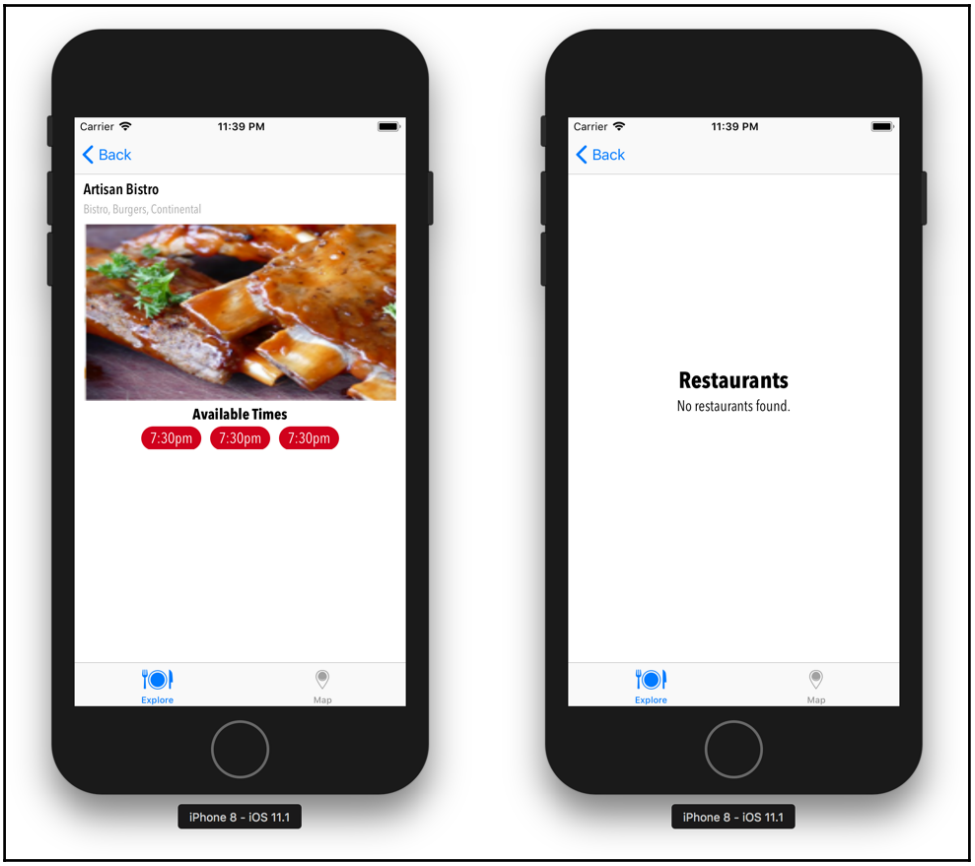

Triggered Segues
selection
RestaurantDetail
Show
Outlets
backgroundView
imgRestaurant
lblCuisine
Lbl Cuisine
lblTitle
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label
Label
american
Available Times
7:30pm
7:30pm
7:30pm

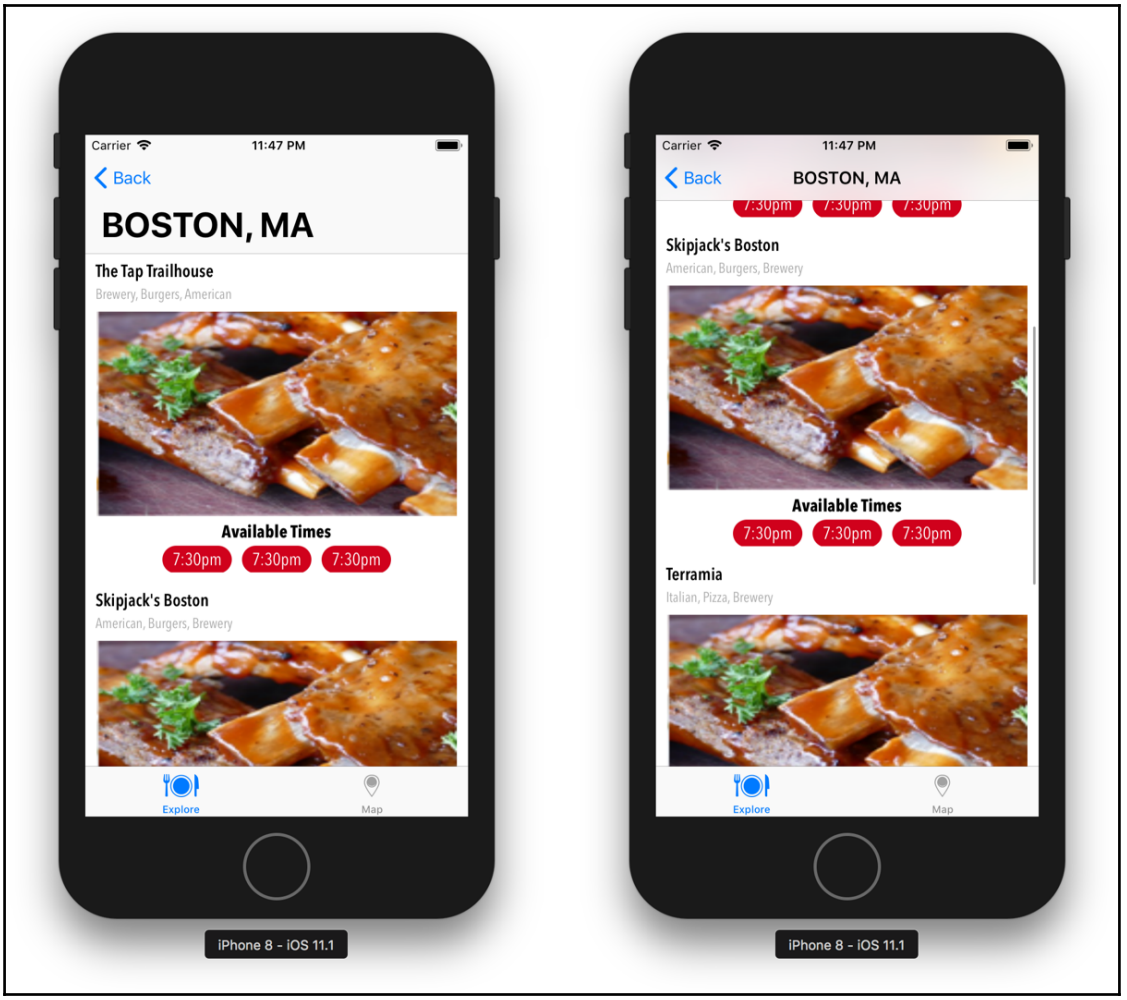
```
import Foundation

class RestaurantDataManager {

    private var items:[RestaurantItem] = []                                    A

    func fetch(by location:String, withFilter:String="All",  completionHandler:() -> Swift.Void) {
        var restaurants:[RestaurantItem] = []

        for restaurant in RestaurantAPIManager.loadJSON(file: location) {
            restaurants.append(RestaurantItem(dict: restaurant))
        }

        if withFilter != "All" {
            items = restaurants.filter({ $0.cuisines.contains(withFilter) })
        }
        else { items = restaurants }

        completionHandler()
    }

    func numberOfItems() -> Int {
        return items.count
    }

    func restaurantItem(at index:IndexPath) -> RestaurantItem {
        return items[index.item]
    }
}
```

C B D E F G H

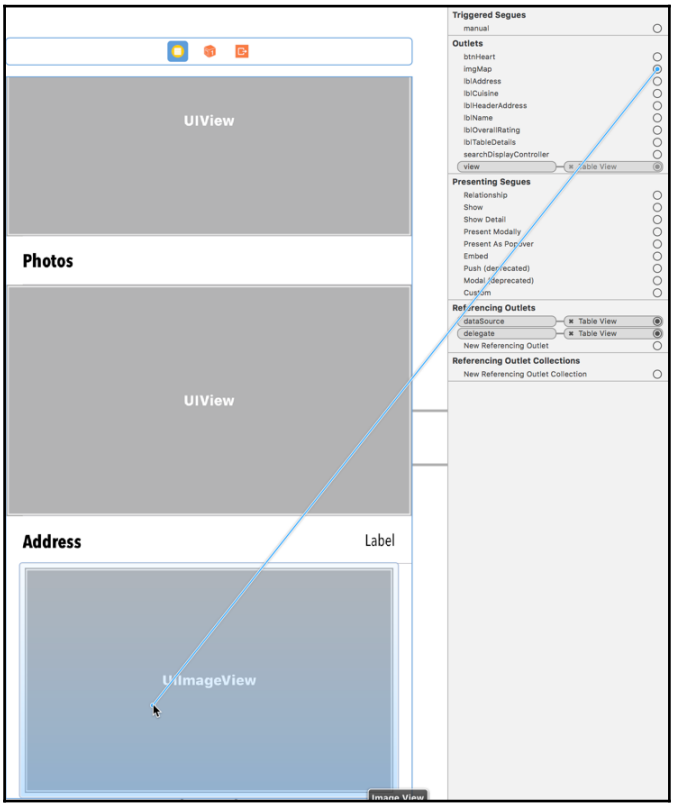
```
M          Void fetch(by: String, completionHandler: () -> Void)
M          Void fetch(by: String, withFilter: String, completionHandler: () -> Void)
M           Int numberOfItems()
M RestaurantItem restaurantItem(at: IndexPath)
```

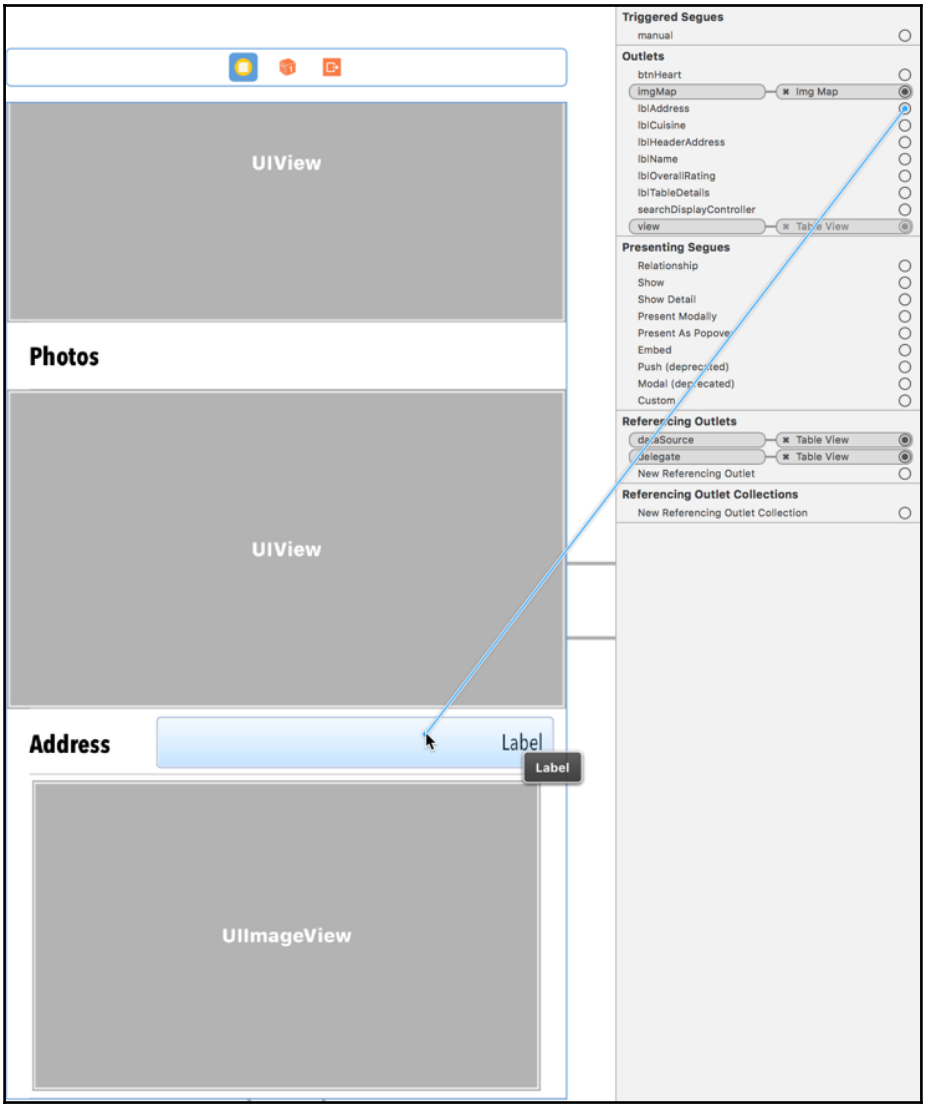

TITLE GOES HERE
Description goes here

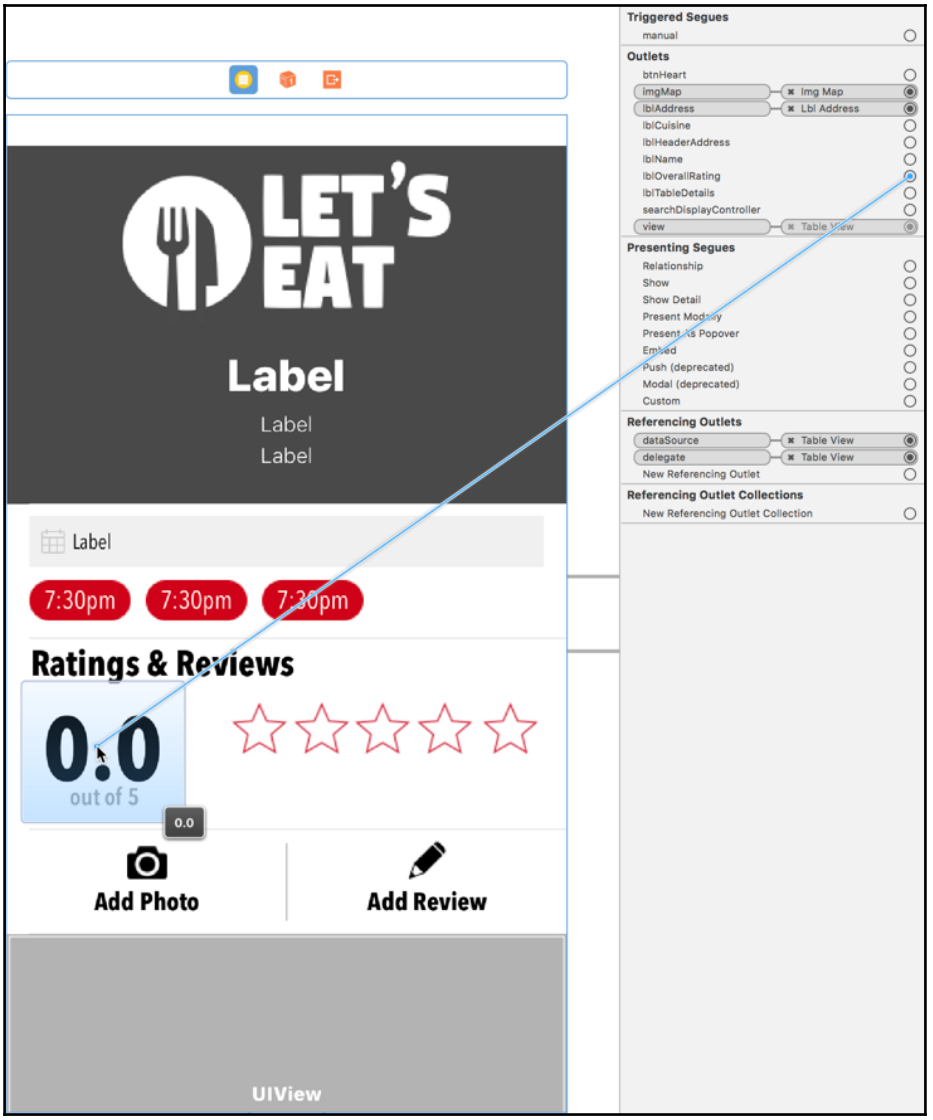

Carrier
11:39 PM
Back
Artisan Bistro
Bistro, Burgers, Continental
Available Times
7:30pm
7:30pm
7:30pm
Explore
Map
iPhone 8 - iOS 11.1
Carrier
11:39 PM
Back
Restaurants
No restaurants found.
Explore
Map
iPhone 8 - iOS 11.1

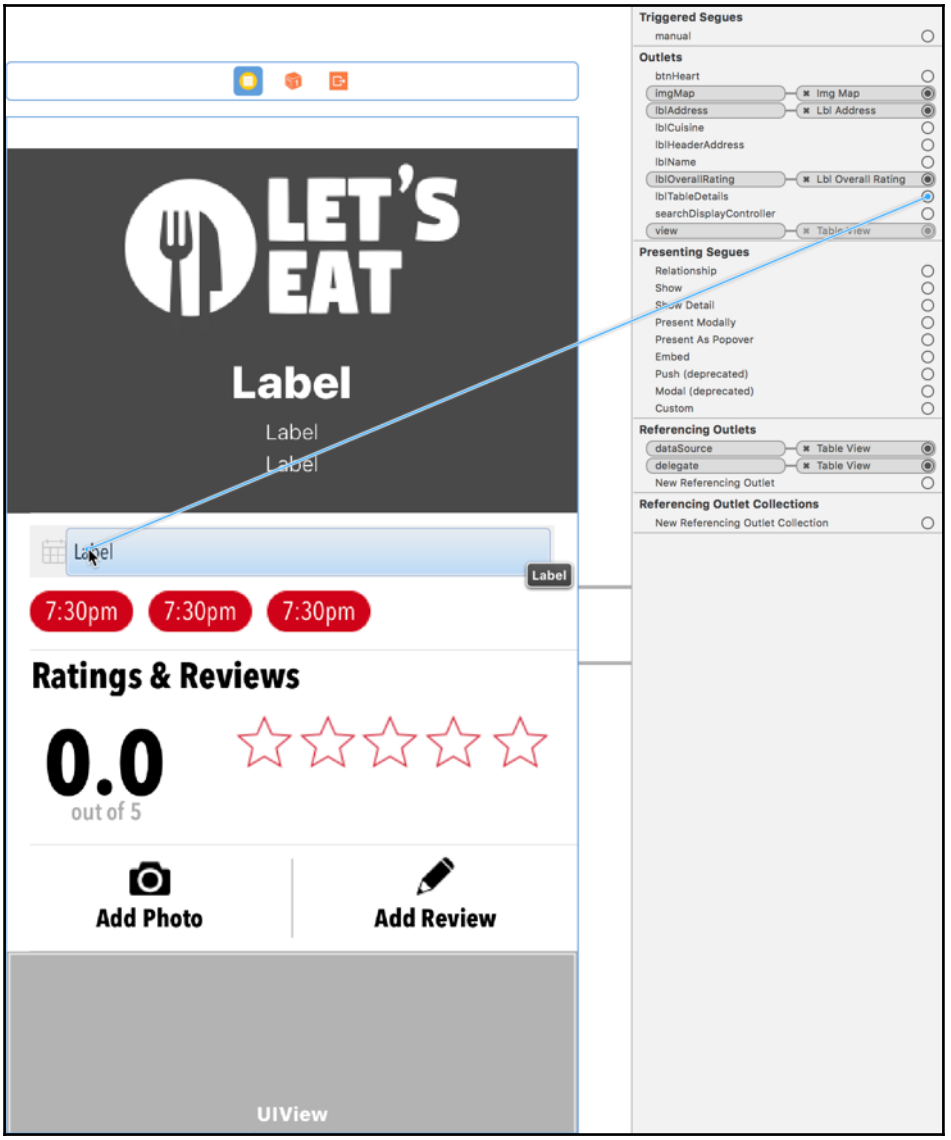

Carrier
11:47 PM
Back
BOSTON, MA
The Tap Trailhouse
Brewery, Burgers, American
Available Times
7:30pm
7:30pm
7:30pm
Skipjack's Boston
American, Burgers, Brewery
Explore
Map
iPhone 8 - iOS 11.1
Carrier
11:47 PM
Back
BOSTON, MA
Skipjack's Boston
American, Burgers, Brewery
Available Times
7:30pm
7:30pm
7:30pm
Terramia
Italian, Pizza, Brewery
Explore
Map
iPhone 8 - iOS 11.1

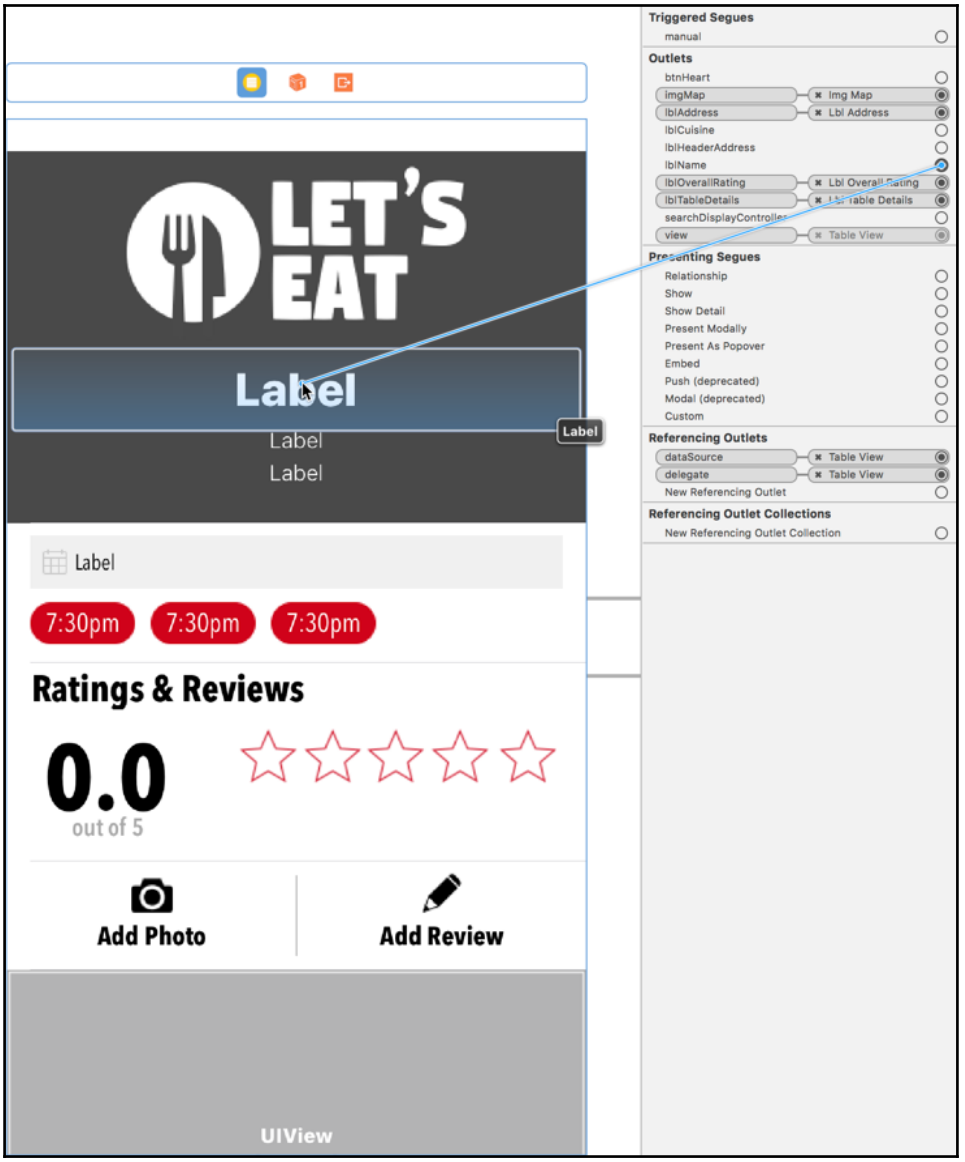
Chapter 18: Displaying Data in Restaurant Detail

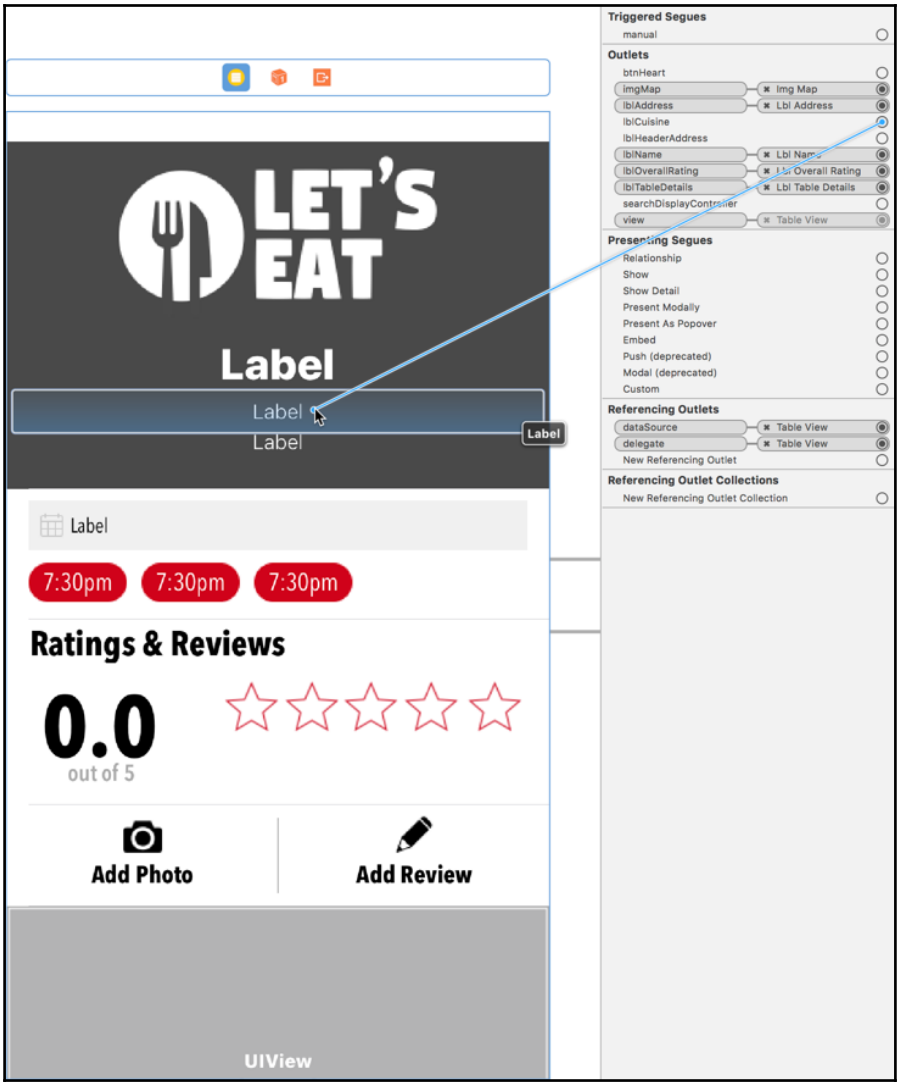


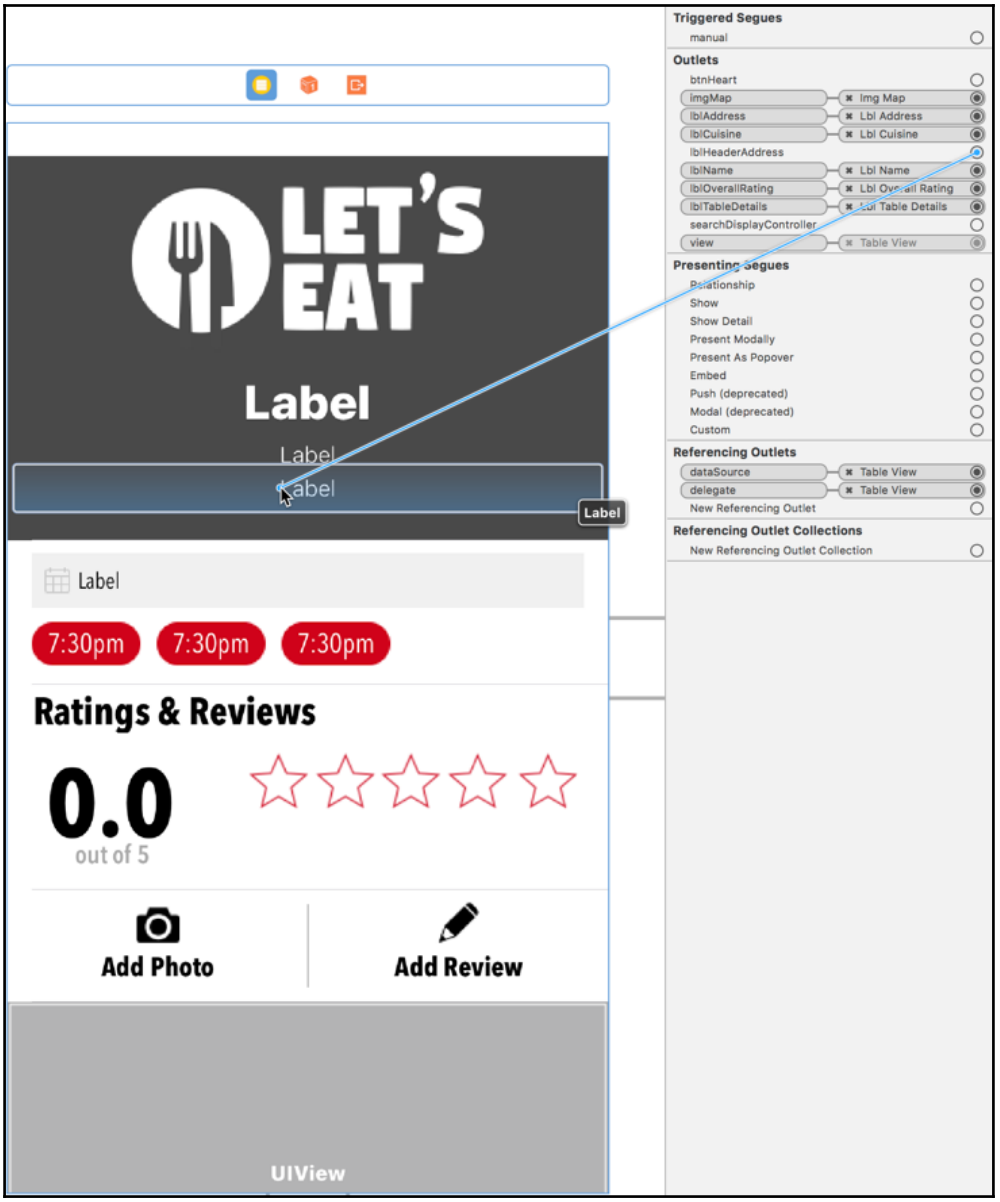

Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
searchDisplayController
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popove
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
UIView
Photos
UIView
Address
Label
Label
UIImageView

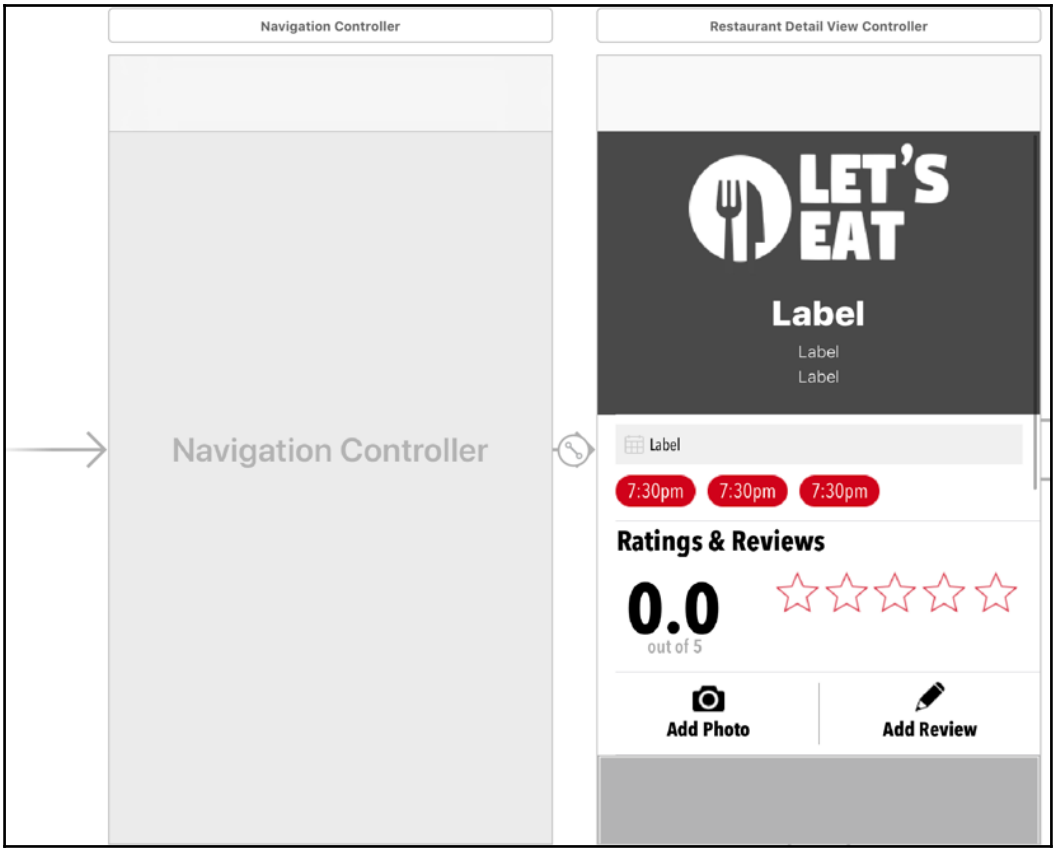

LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
0.0
Add Photo
Add Review
UIView
Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
Lbl Address
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
searchDisplayController
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present as Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

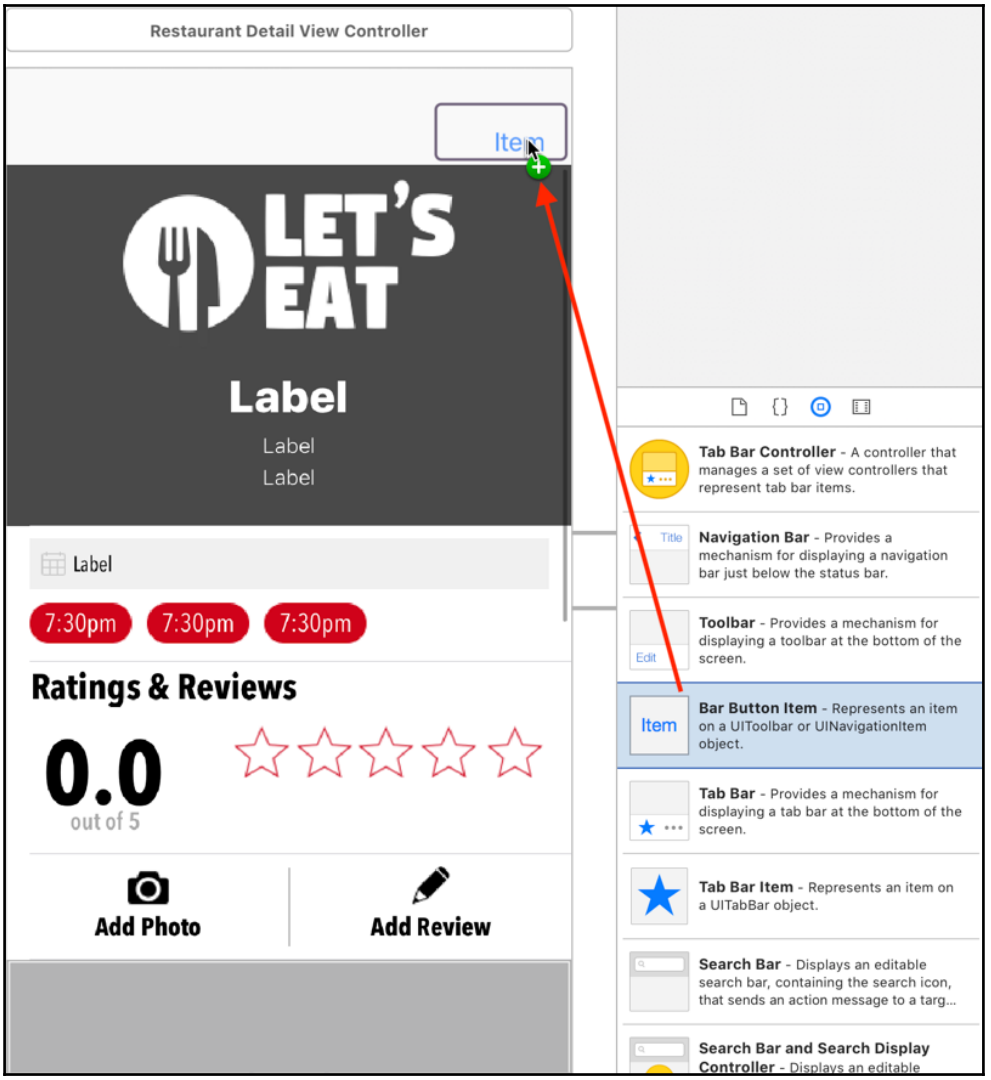

Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
Lbl Address
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
Lbl Overall Rating
lblTableDetails
searchDisplayController
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView

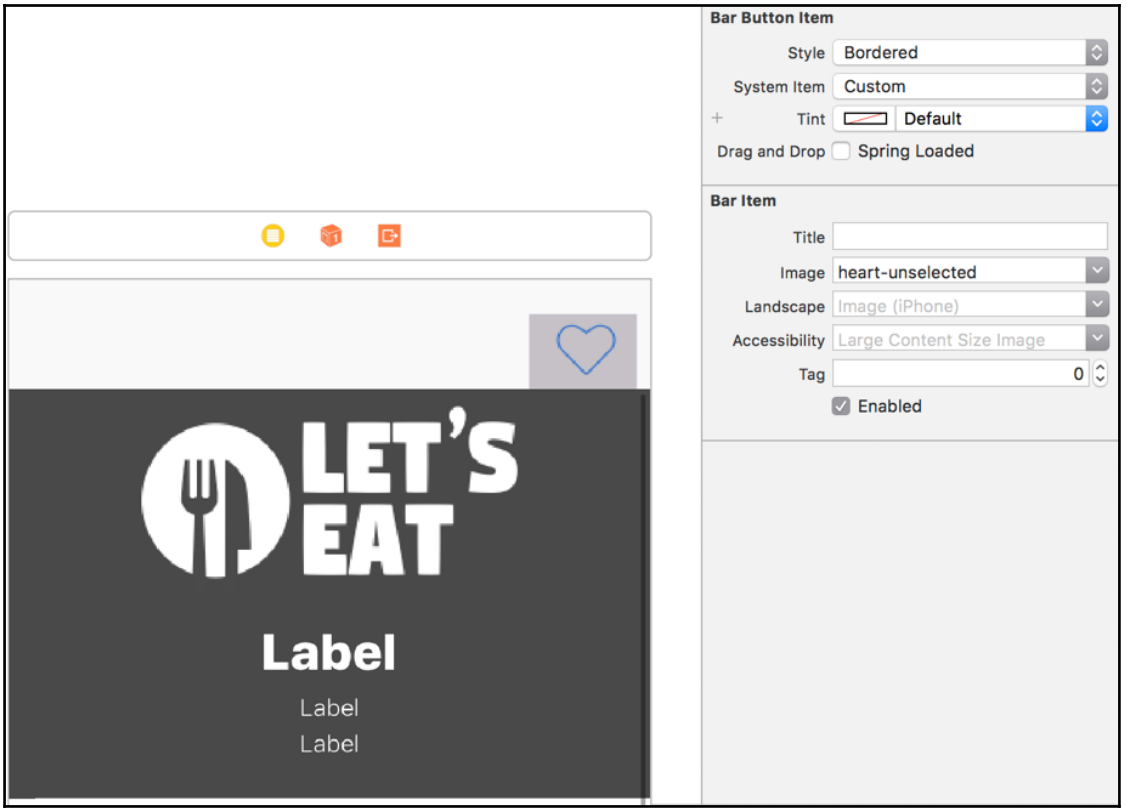

LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
Lbl Address
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
searchDisplayController
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

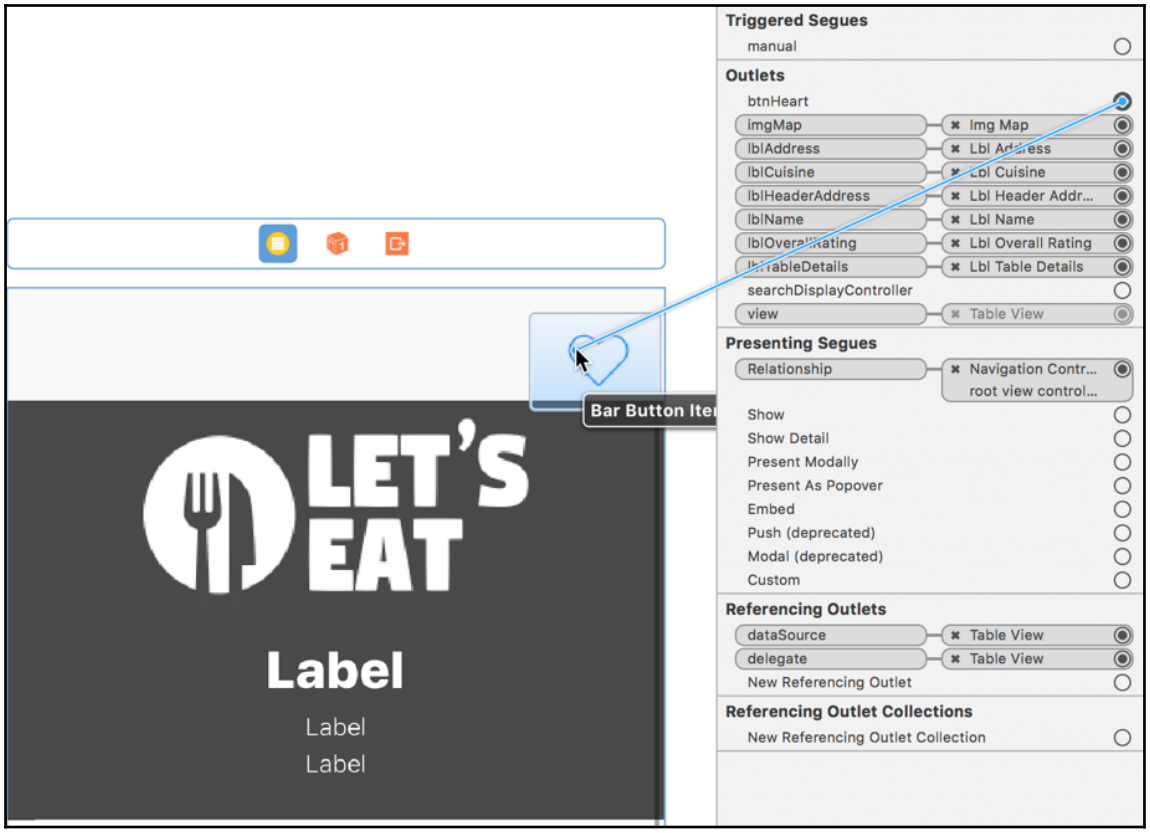

LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
Lbl Address
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
Lbl Table Details
searchDisplayController
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

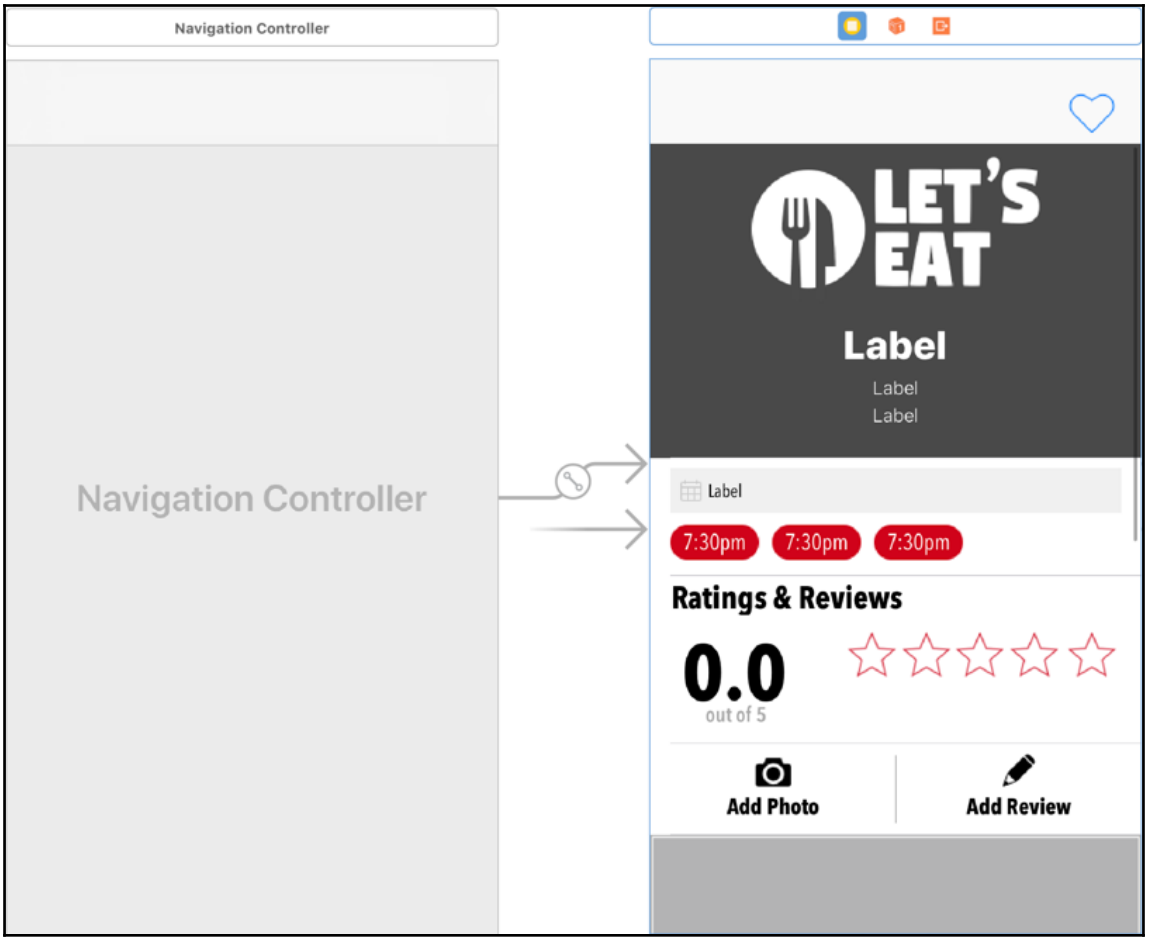

LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView
Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
Lbl Address
lblCuisine
Lbl Cuisine
lblHeaderAddress
lblName
Lbl Name
lblOverallRating
lblTableDetails
searchDisplayController
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

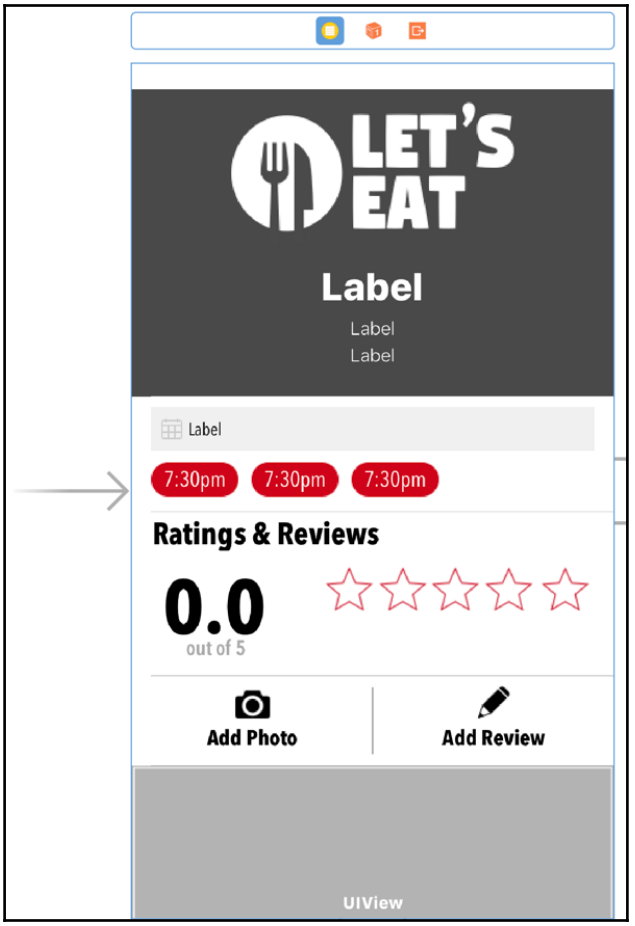

Navigation Controller
Restaurant Detail View Controller
Navigation Controller
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review

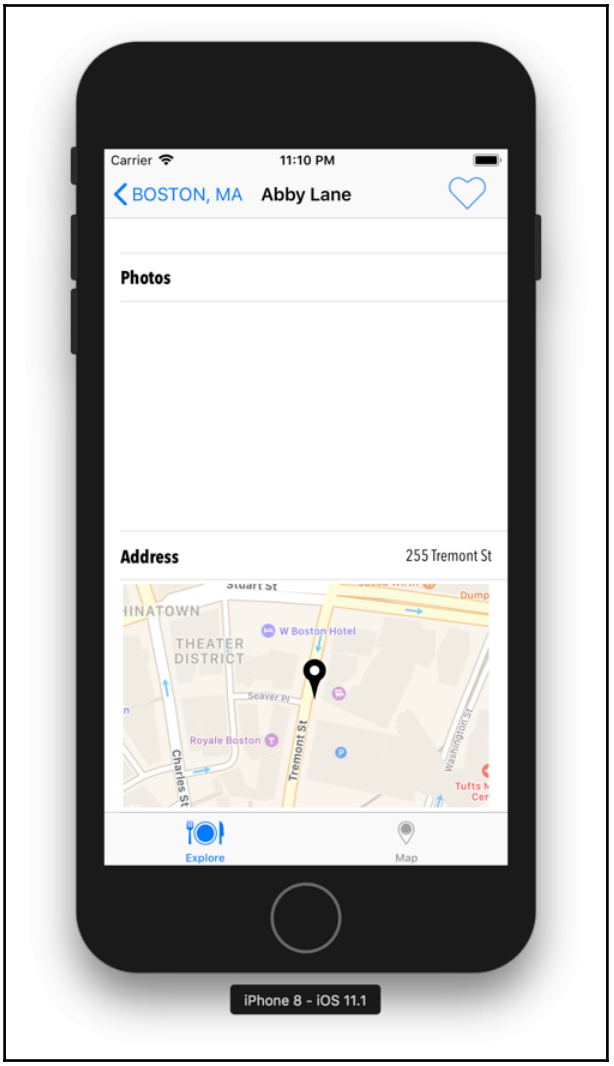

Restaurant Detail View Controller
Item
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.
Navigation Bar - Provides a mechanism for displaying a navigation bar just below the status bar.
Toolbar - Provides a mechanism for displaying a toolbar at the bottom of the screen.
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.
Tab Bar - Provides a mechanism for displaying a tab bar at the bottom of the screen.
Tab Bar Item - Represents an item on a UITabBar object.
Search Bar - Displays an editable search bar, containing the search icon, that sends an action message to a targ...
Search Bar and Search Display Controller - Displays an editable


Bar Button Item
Style
Bordered
System Item
Custom
Tint
Default
Drag and Drop
Spring Loaded
Bar Item
Title
Image
heart-unselected
Landscape
Image (iPhone)
Accessibility
Large Content Size Image
Tag
0
Enabled
LET'S EAT
Label
Label
Label


Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
lblCuisine
Lbl Cuisine
lblHeaderAddress
Lbl Header Addr...
lblName
Lbl Name
Lbl Overall Rating
Lbl Table Details
searchDisplayController
view
Table View
Presenting Segues
Relationship
Navigation Contr...
root view control...
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
LET'S EAT
Label
Label
Label


Navigation Controller
Navigation Controller
LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review


LET'S EAT
Label
Label
Label
Label
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
UIView

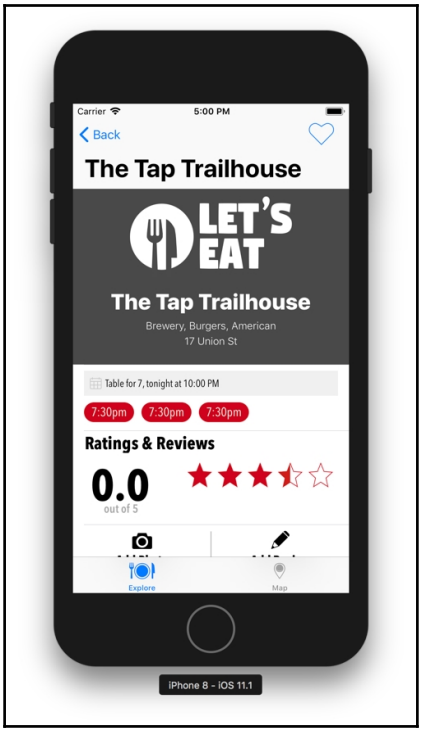

Carrier
11:10 PM
BOSTON, MA
Abby Lane
Photos
Address
255 Tremont St
W Boston Hotel
THEATER DISTRICT
Seaver Pl
Royale Boston
Tremont St
Charles St
Explore
Map
iPhone 8 - iOS 11.1

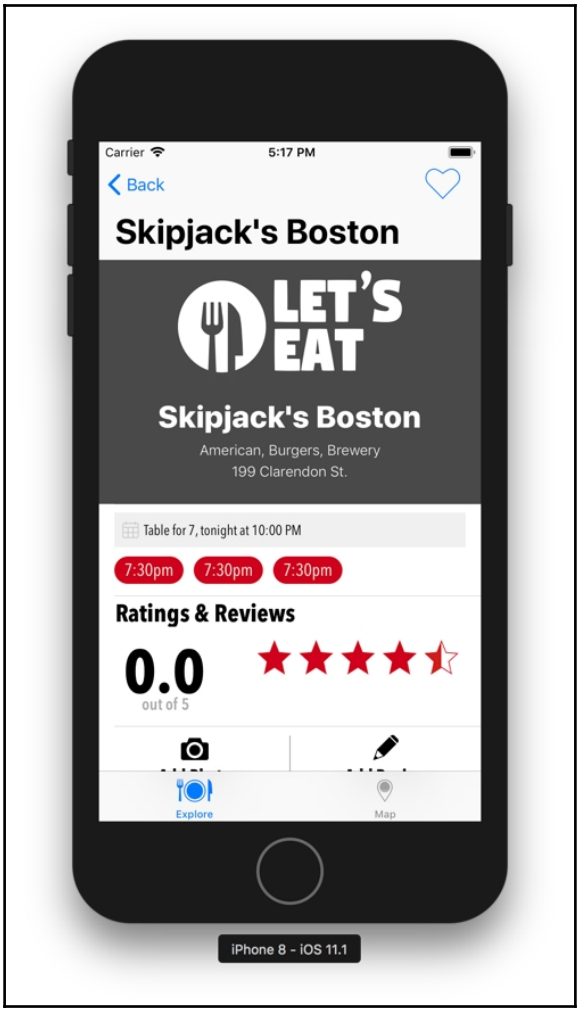
Chapter 19: Foodie Reviews

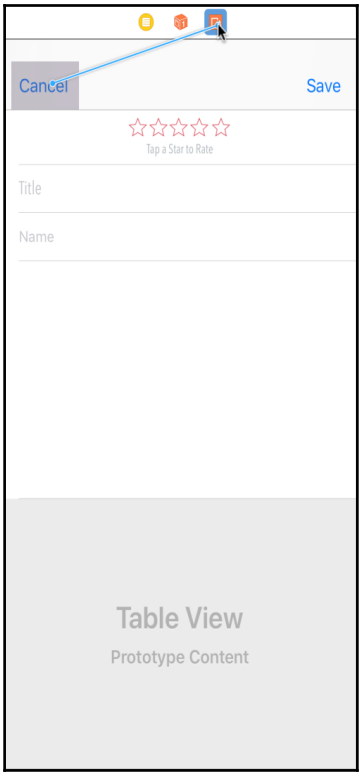
```
import UIKit

class RatingView: UIControl {

    let imgFilledStar =
}
```

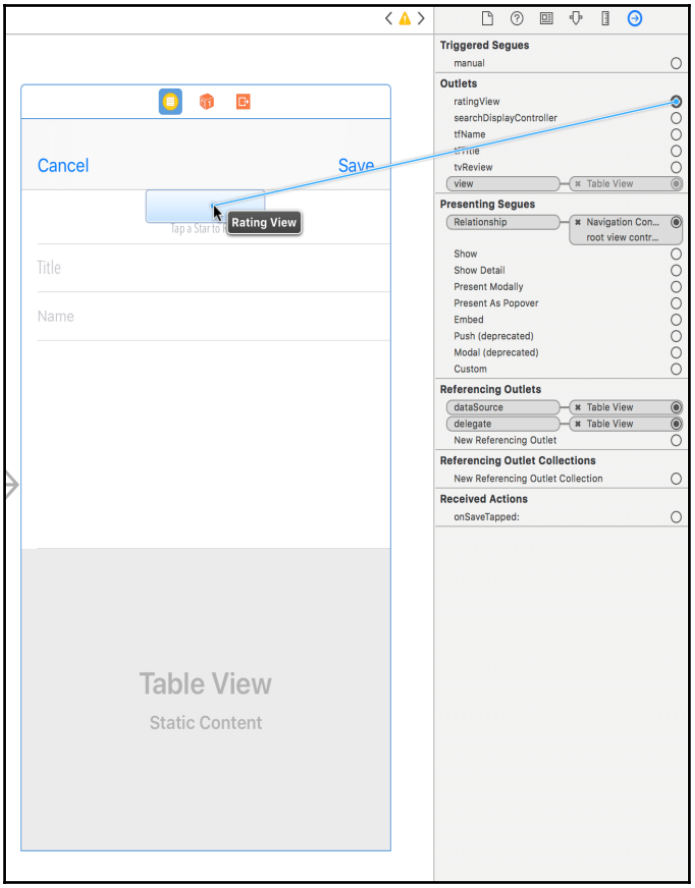
```
import UIKit

class RatingView: UIControl

    let imgFilledStar =
}
```

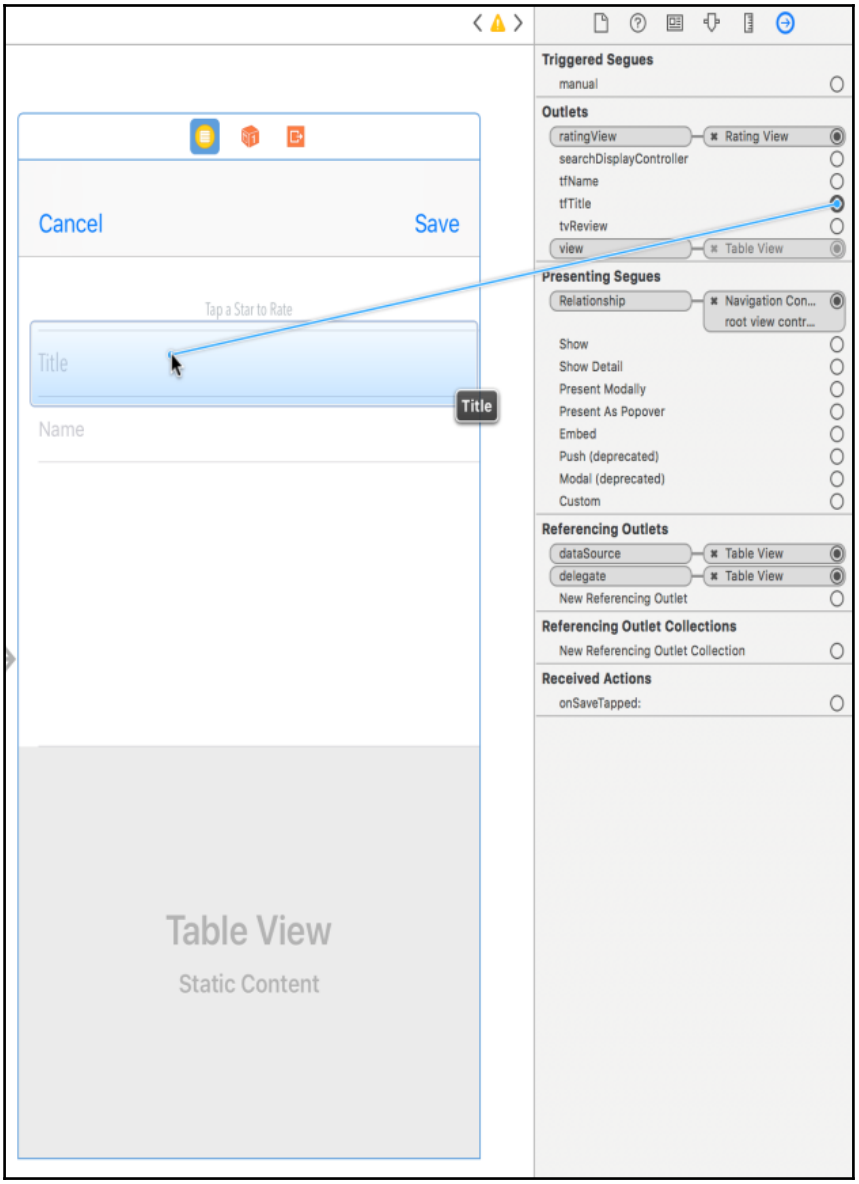

```
import UIKit

class RatingView: UIControl {

    let imgFilledStar =
    let imgHalfStar =
    let imgEmptyStar =

    let shouldBecomeFirstResponder = true
    var rating:CGFloat = 0.0
    var totalStars = 5
}
```

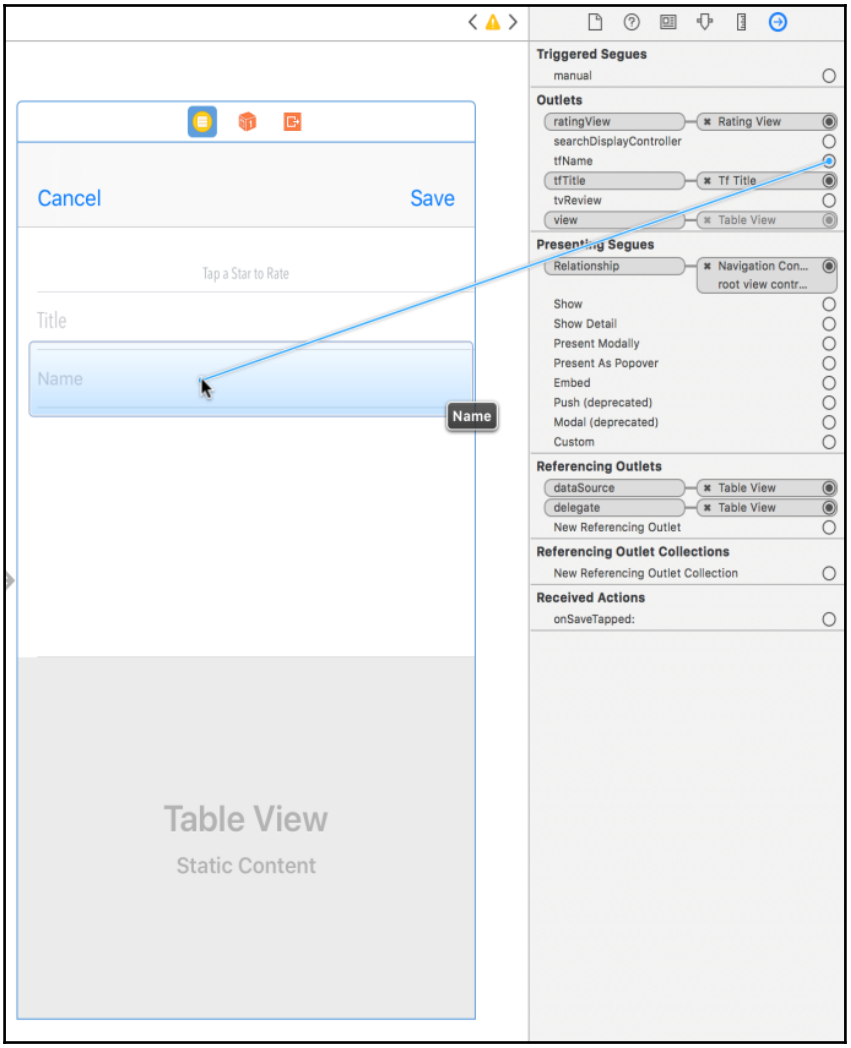

Carrier
5:00 PM
Back
The Tap Trailhouse
LET'S EAT
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Explore
Map
iPhone 8 - iOS 11.1

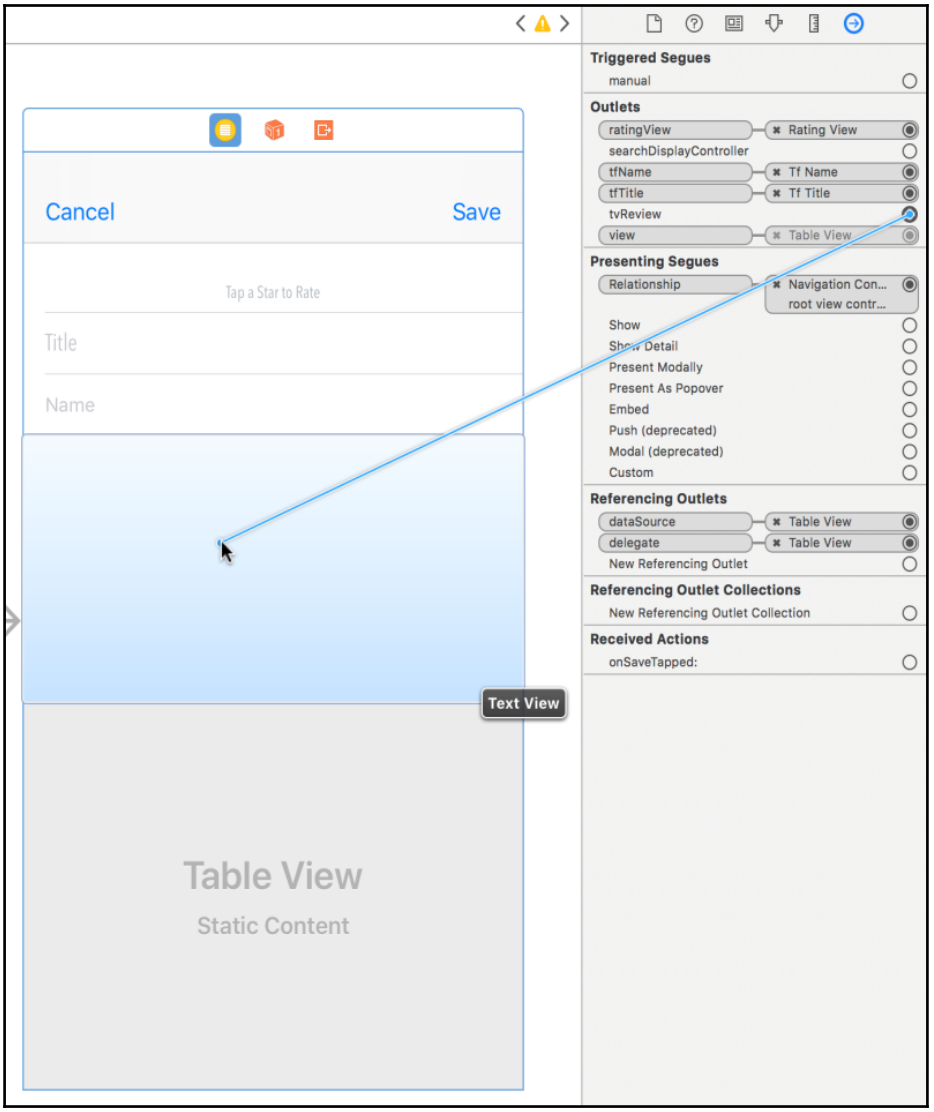

Carrier
5:17 PM
Back
Skipjack's Boston
LET'S EAT
Skipjack's Boston
American, Burgers, Brewery
199 Clarendon St.
Table for 7, tonight at 10:00 PM
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Explore
Map
iPhone 8 - iOS 11.1

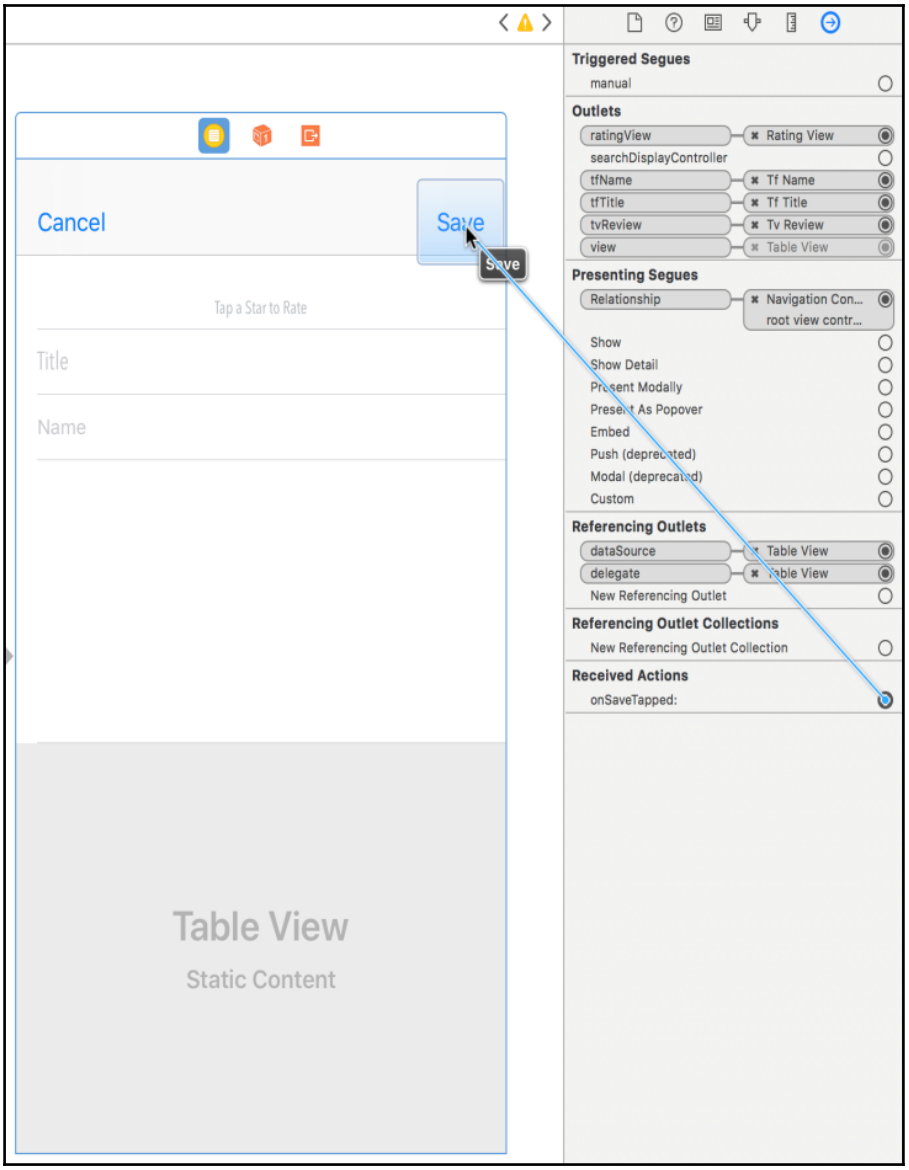

Cancel
Save
Tap a Star to Rate
Title
Name
Table View
Prototype Content


Triggered Segues
manual
Outlets
ratingView
searchDisplayController
tfName
tvReview
view
Table View
Presenting Segues
Relationship
Navigation Con...
root view contr...
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onSaveTapped:
Cancel
Save
Rating View
Title
Name
Table View
Static Content


Triggered Segues
manual
Outlets
ratingView
Rating View
searchDisplayController
tfName
tfTitle
tvReview
view
Table View
Presenting Segues
Relationship
Navigation Con...
root view contr...
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onSaveTapped:
Cancel
Save
Tap a Star to Rate
Title
Title
Name
Table View
Static Content


Cancel
Save
Tap a Star to Rate
Title
Name
Name
Table View
Static Content
Triggered Segues
manual
Outlets
ratingView
Rating View
searchDisplayController
tfName
tfTitle
Tf Title
tvReview
view
Table View
Presenting Segues
Relationship
Navigation Con...
root view contr...
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onSaveTapped:


Triggered Segues
manual
Outlets
ratingView
Rating View
searchDisplayController
tfName
Tf Name
tfTitle
Tf Title
tvReview
view
Table View
Presenting Segues
Relationship
Navigation Con...
root view contr...
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onSaveTapped:
Cancel
Save
Tap a Star to Rate
Title
Name
Text View
Table View
Static Content


Cancel
Save
Tap a Star to Rate
Title
Name
Table View
Static Content
Triggered Segues
manual
Outlets
ratingView
Rating View
searchDisplayController
tfName
Tf Name
tfTitle
Tf Title
tvReview
Tv Review
view
Table View
Presenting Segues
Relationship
Navigation Con...
root view contr...
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onSaveTapped:

Chapter 20: Working with Photo Filters

Key	Type	Value
▼ Root	Array	(10 items)
▼ Item 0	Dictionary	(2 items)
filter	String	None
name	String	None
▼ Item 1	Dictionary	(2 items)
filter	String	CIPhotoEffectMono
name	String	Mono
▼ Item 2	Dictionary	(2 items)
filter	String	CISepiaTone
name	String	Sepia
▼ Item 3	Dictionary	(2 items)
filter	String	CIPhotoEffectTonal
name	String	Tonal
▼ Item 4	Dictionary	(2 items)
filter	String	CIPhotoEffectNoir
name	String	Noir
▼ Item 5	Dictionary	(2 items)
filter	String	CIPhotoEffectFade
name	String	Fade
▼ Item 6	Dictionary	(2 items)
filter	String	CIPhotoEffectChrome
name	String	Chrome
▼ Item 7	Dictionary	(2 items)
filter	String	CIPhotoEffectProcess
name	String	Process
▼ Item 8	Dictionary	(2 items)
filter	String	CIPhotoEffectTransfer
name	String	Transfer
▼ Item 9	Dictionary	(2 items)
filter	String	CIPhotoEffectInstant
name	String	Instant


Carrier
6:23 PM
Cancel
Save
None
Mono

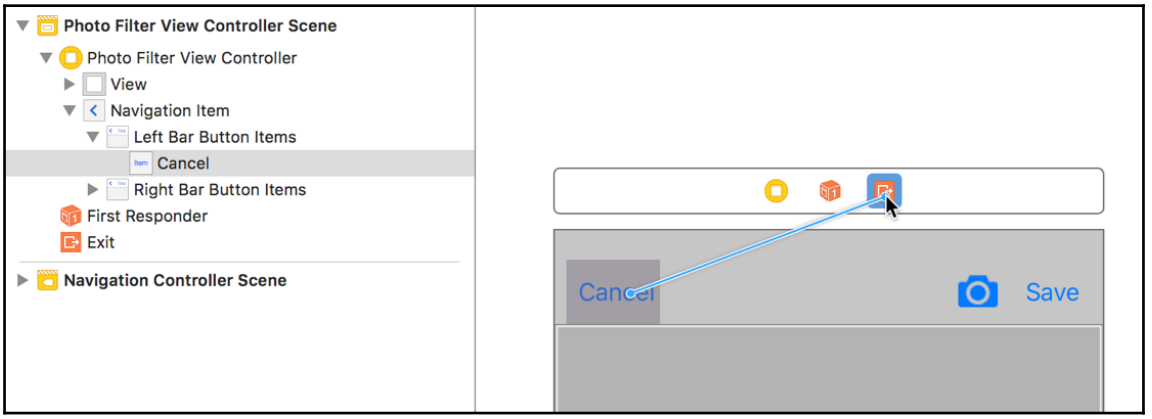
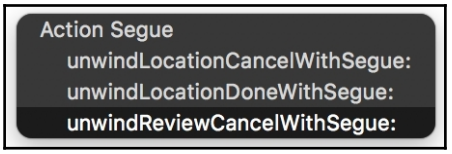
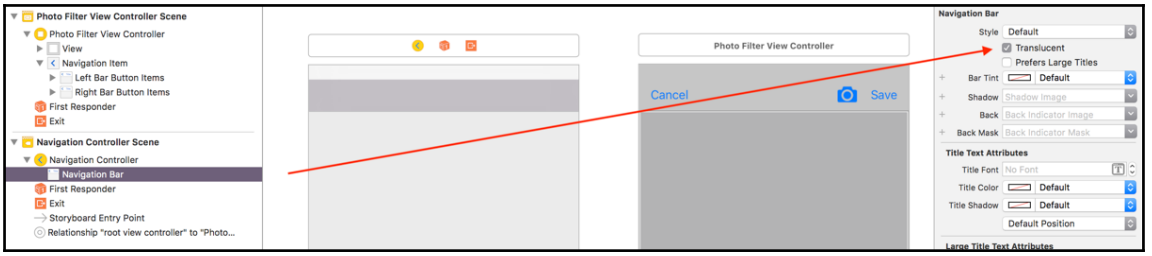
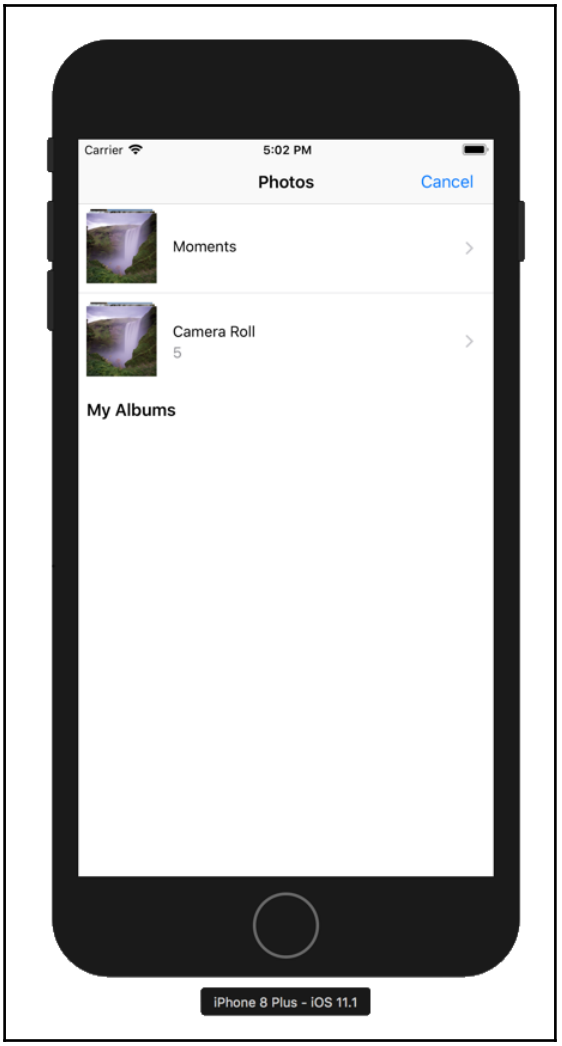


Photo Filter View Controller Scene
Photo Filter View Controller
View
Navigation Item
Left Bar Button Items
Cancel
Right Bar Button Items
First Responder
Exit
Navigation Controller Scene
Cancel
Save



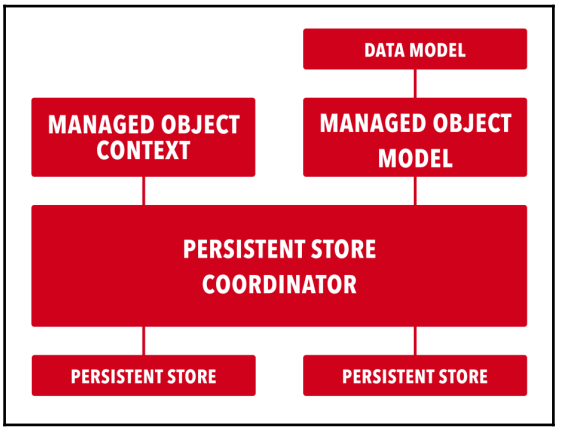


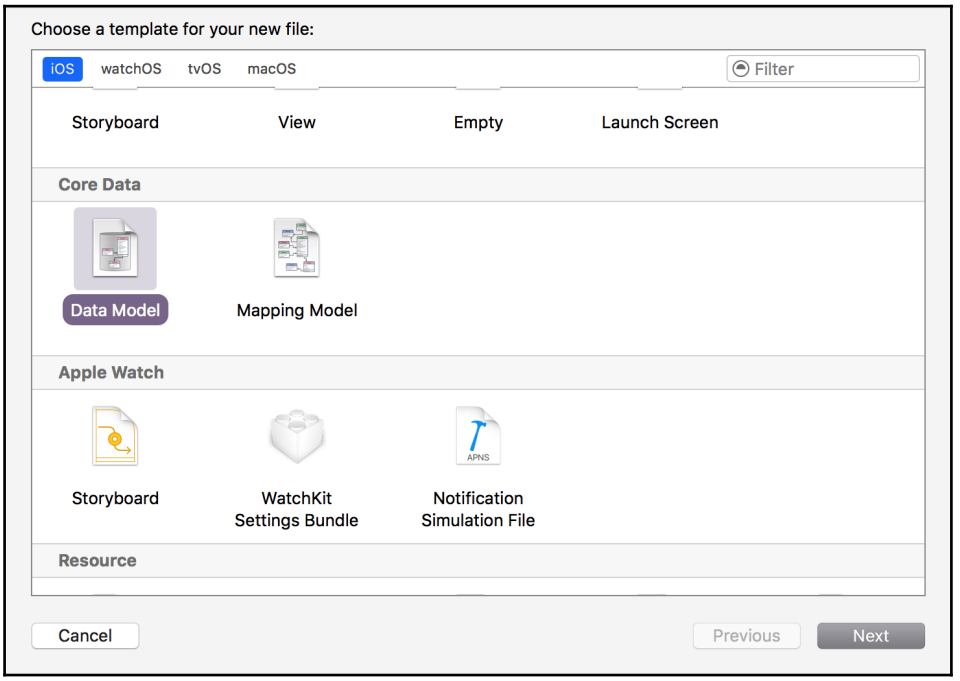
Key	Type	Value
▼ Information Property List	Dictionary	(16 items)
Localization native development re...	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
Privacy - Camera Usage Description	String	The app uses your camera to take pictures
Privacy - Photo Library Usage Des...	String	The app uses your camera to take pictures
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (i...	Array	(4 items)

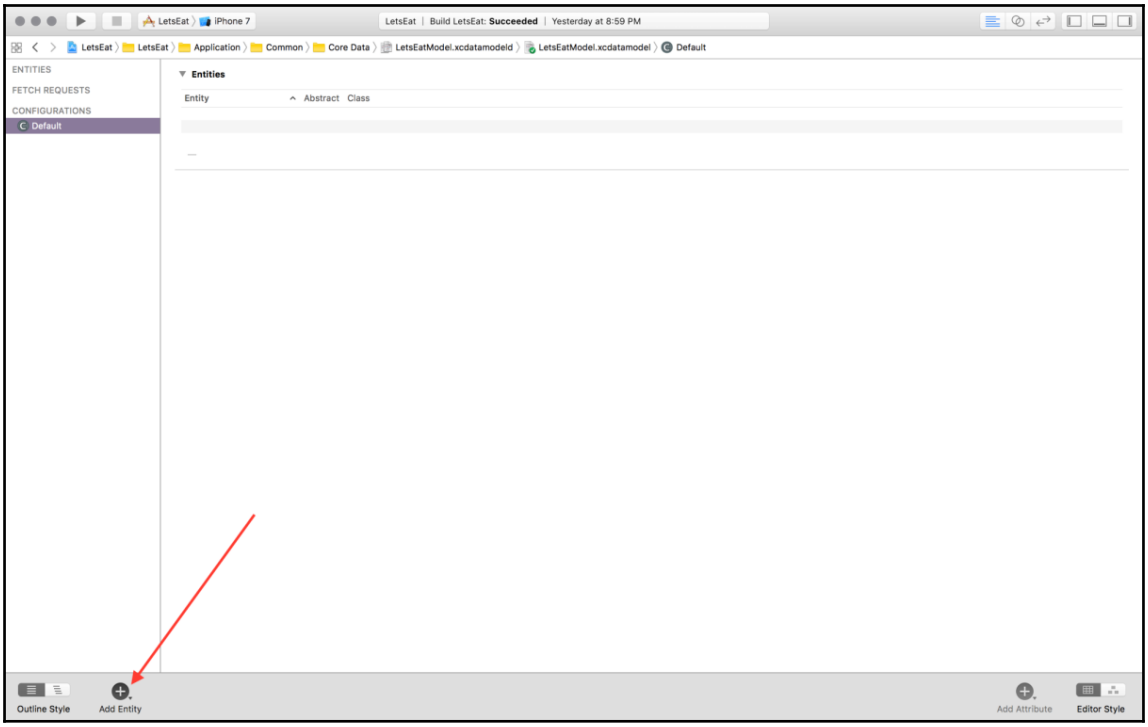

Carrier
5:02 PM
Photos
Cancel
Moments
Camera Roll
5
My Albums
iPhone 8 Plus - iOS 11.1

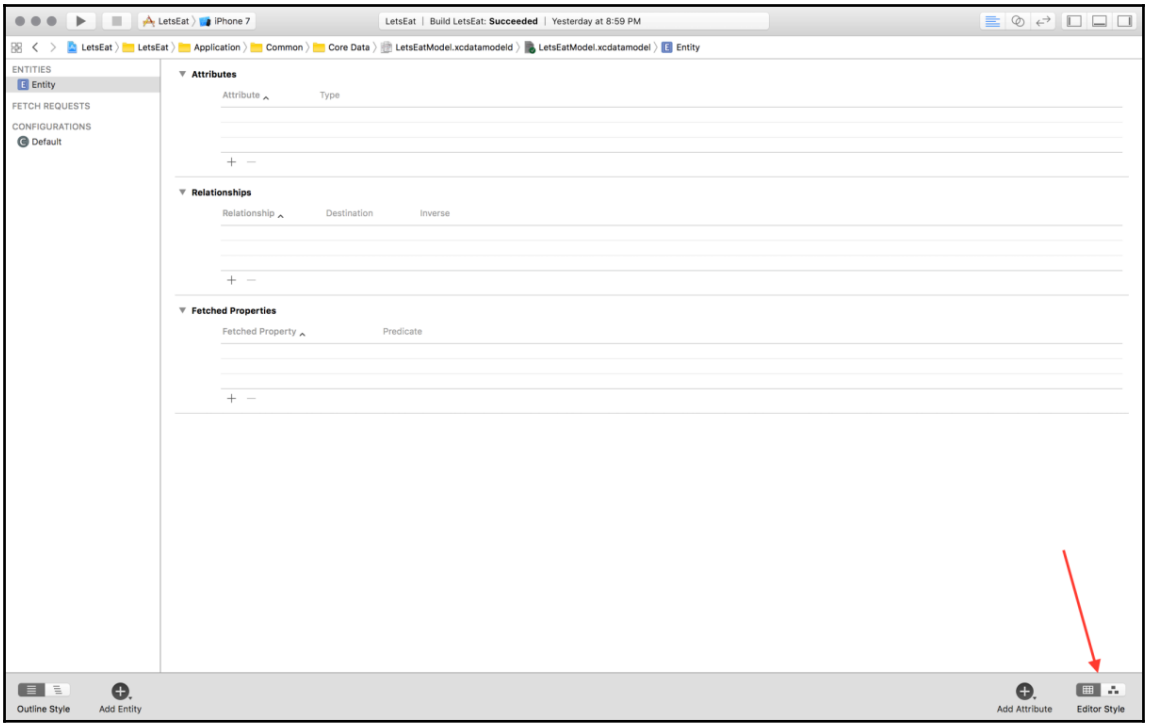

Carrier
5:02 PM
Cancel
Save
None
Mono
Sepia
iPhone 8 Plus - iOS 11.1

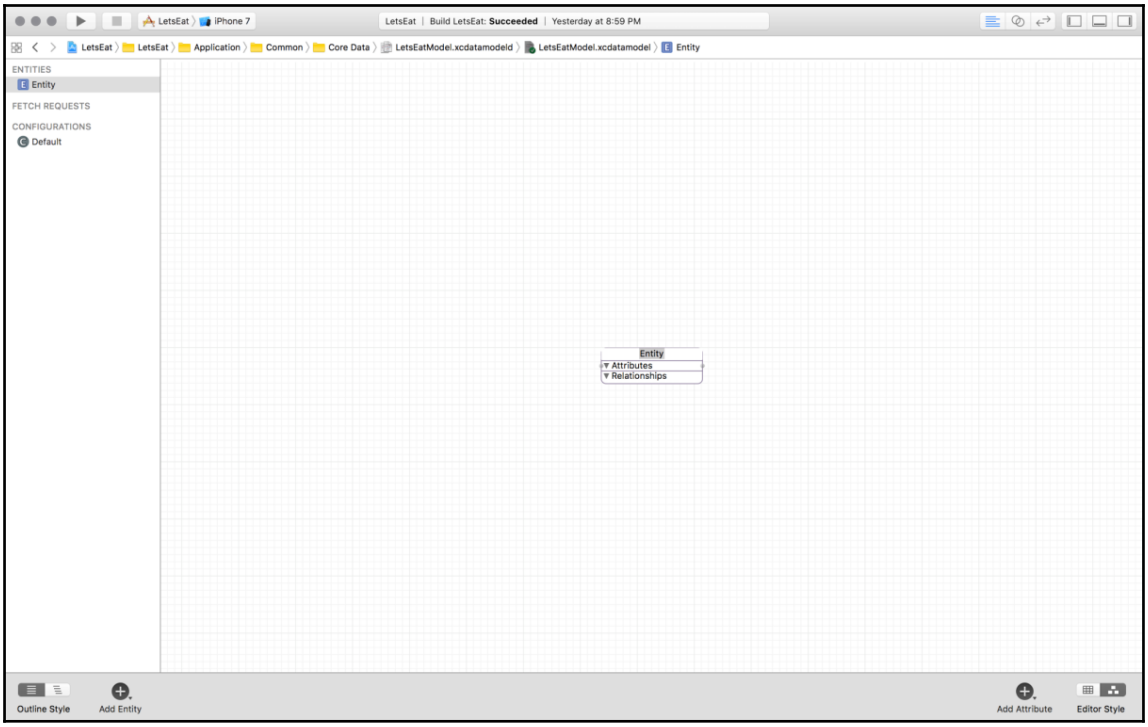
Chapter 21: Understanding Core Data

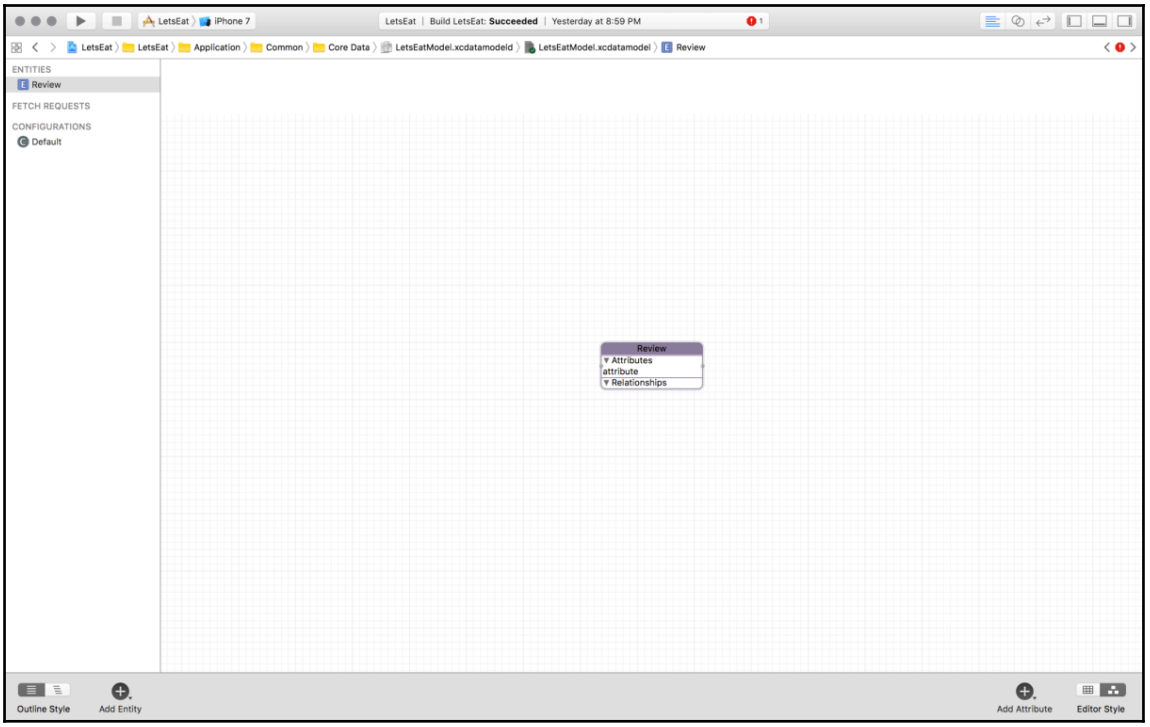


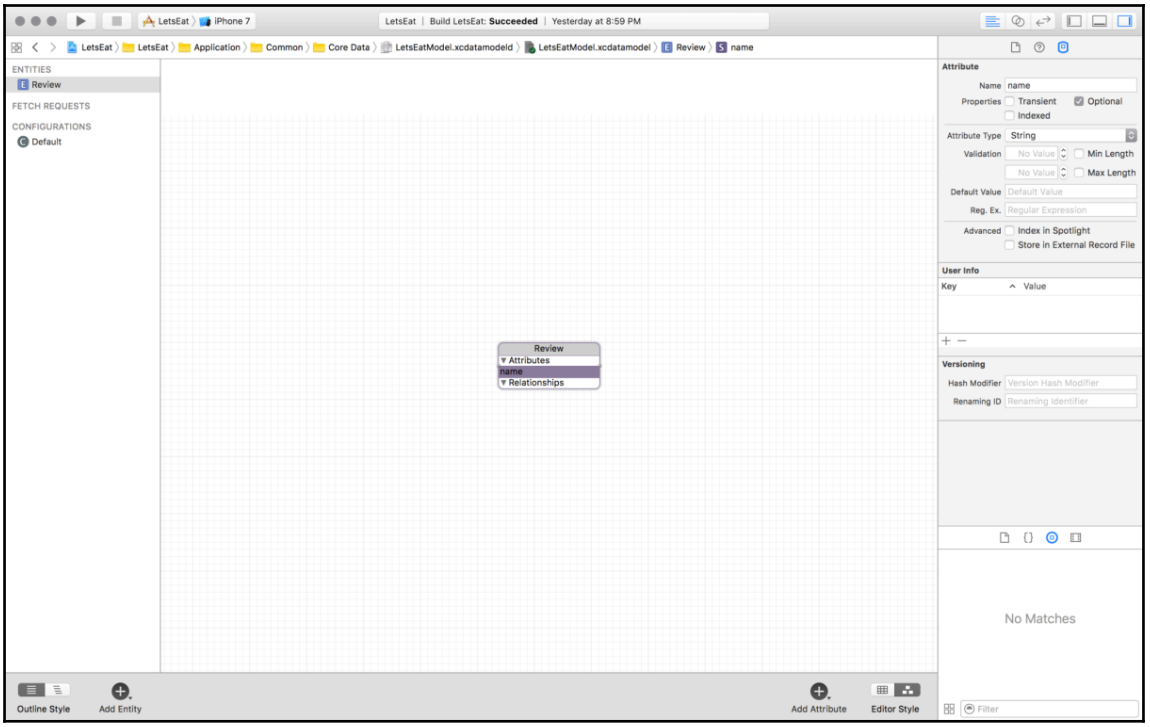


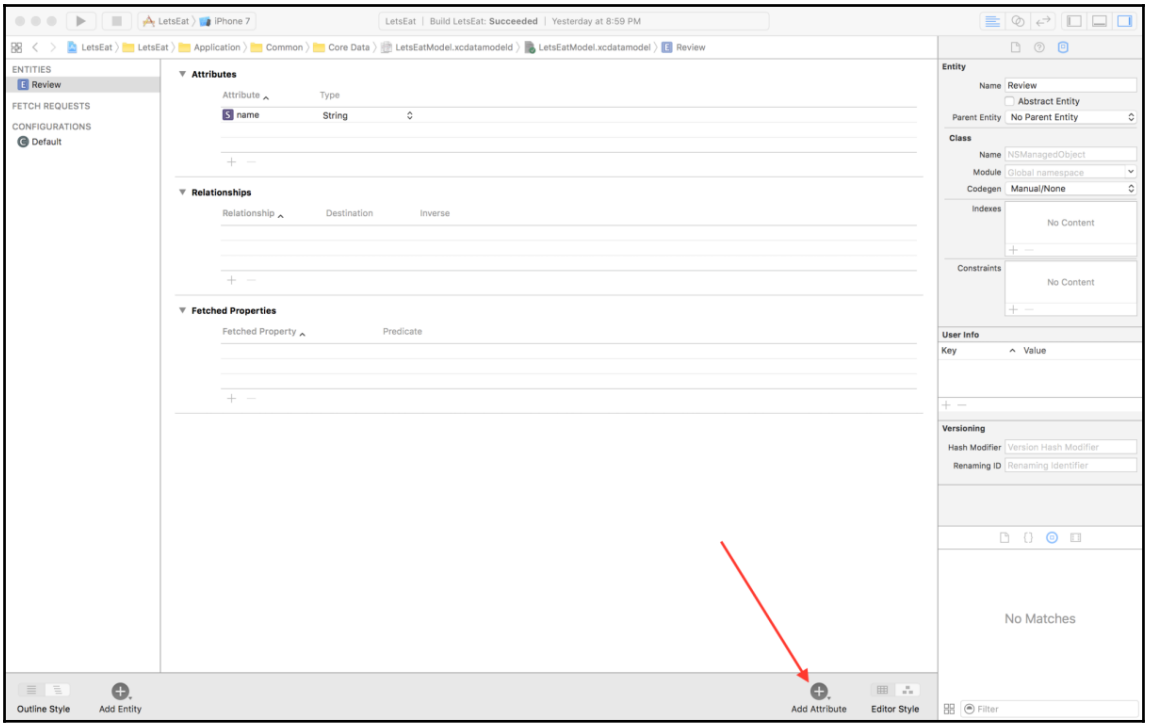

LetsEat
iPhone 7
LetsEat | Build LetsEat: Succeeded | Yesterday at 8:59 PM
LetsEat
LetsEat
Application
Common
Core Data
LetsEatModel.xcdatamodeld
LetsEatModel.xcdatamodel
Default
ENTITIES
FETCH REQUESTS
CONFIGURATIONS
Default
Entities
Entity
Abstract
Class
Outline Style
Add Entity
Add Attribute
Editor Style

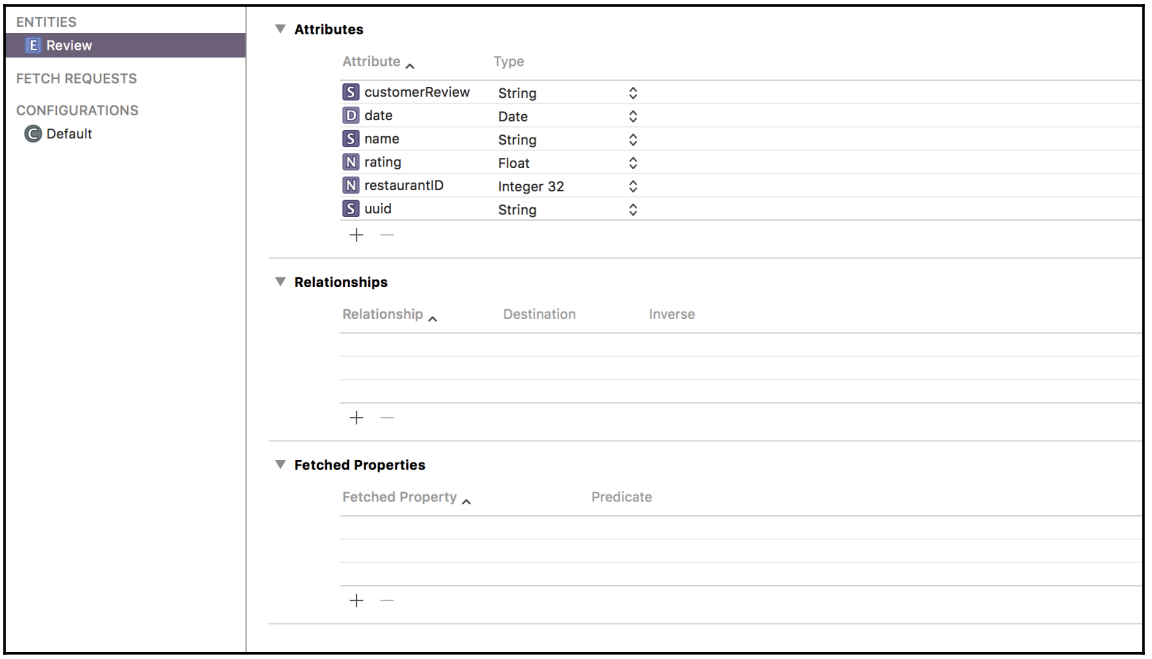

LetsEat
iPhone 7
LetsEat | Build LetsEat: Succeeded | Yesterday at 8:59 PM
LetsEat
LetsEat
Application
Common
Core Data
LetsEatModel.xcdatamodeld
LetsEatModel.xcdatamodel
Entity
ENTITIES
Entity
FETCH REQUESTS
CONFIGURATIONS
Default
Attributes
Attribute
Type
Relationships
Relationship
Destination
Inverse
Fetched Properties
Fetched Property
Predicate
Outline Style
Add Entity
Add Attribute
Editor Style

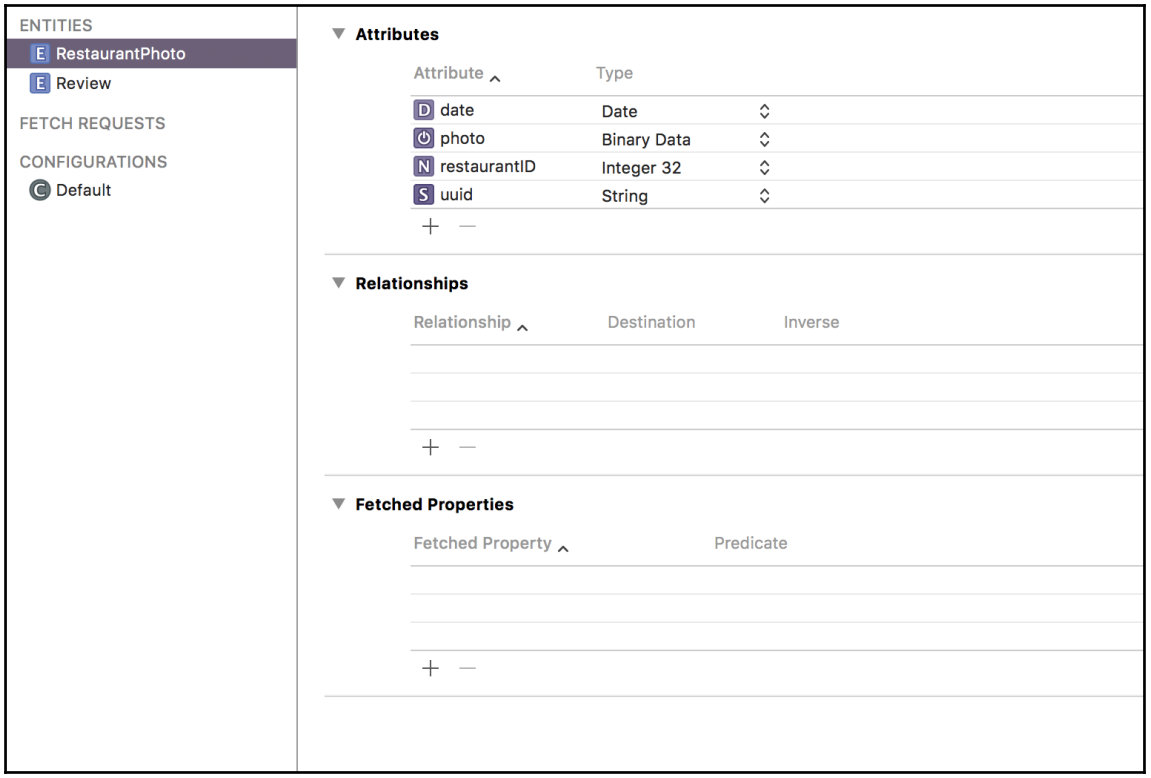

LetsEat
iPhone 7
LetsEat | Build LetsEat: Succeeded | Yesterday at 8:59 PM
LetsEat
LetsEat
Application
Common
Core Data
LetsEatModel.xcdatamodeld
LetsEatModel.xcdatamodel
Entity
ENTITIES
Entity
FETCH REQUESTS
CONFIGURATIONS
Default
Entity
Attributes
Relationships
Outline Style
Add Entity
Add Attribute
Editor Style


LetsEat
iPhone 7
LetsEat | Build LetsEat: Succeeded | Yesterday at 8:59 PM
LetsEat
LetsEat
Application
Common
Core Data
LetsEatModel.xcdatamodeld
LetsEatModel.xcdatamodel
Review
ENTITIES
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
attribute
Relationships
Outline Style
Add Entity
Add Attribute
Editor Style

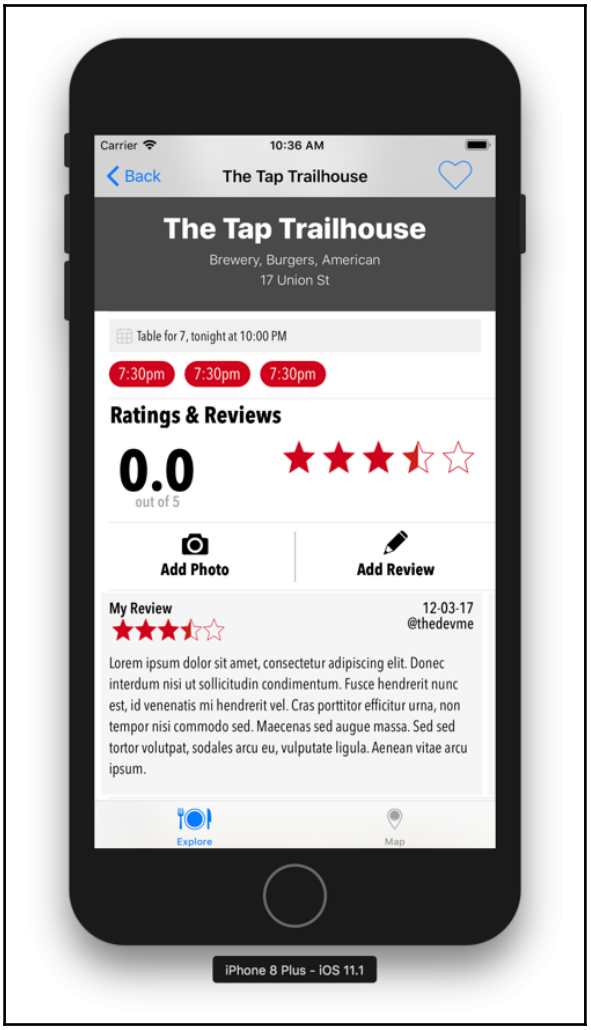

LetsEat
iPhone 7
LetsEat | Build LetsEat: Succeeded | Yesterday at 8:59 PM
LetsEat
LetsEat
Application
Common
Core Data
LetsEatModel.xcdatamodeld
LetsEatModel.xcdatamodel
Review
name
ENTITIES
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
name
Relationships
Attribute
Name
name
Properties
Transient
Optional
Indexed
Attribute Type
String
Validation
No Value
Min Length
No Value
Max Length
Default Value
Default Value
Reg. Ex.
Regular Expression
Advanced
Index in Spotlight
Store in External Record File
User Info
Key
Value
Versioning
Hash Modifier
Version Hash Modifier
Renaming ID
Renaming Identifier
No Matches
Filter
Outline Style
Add Entity
Add Attribute
Editor Style

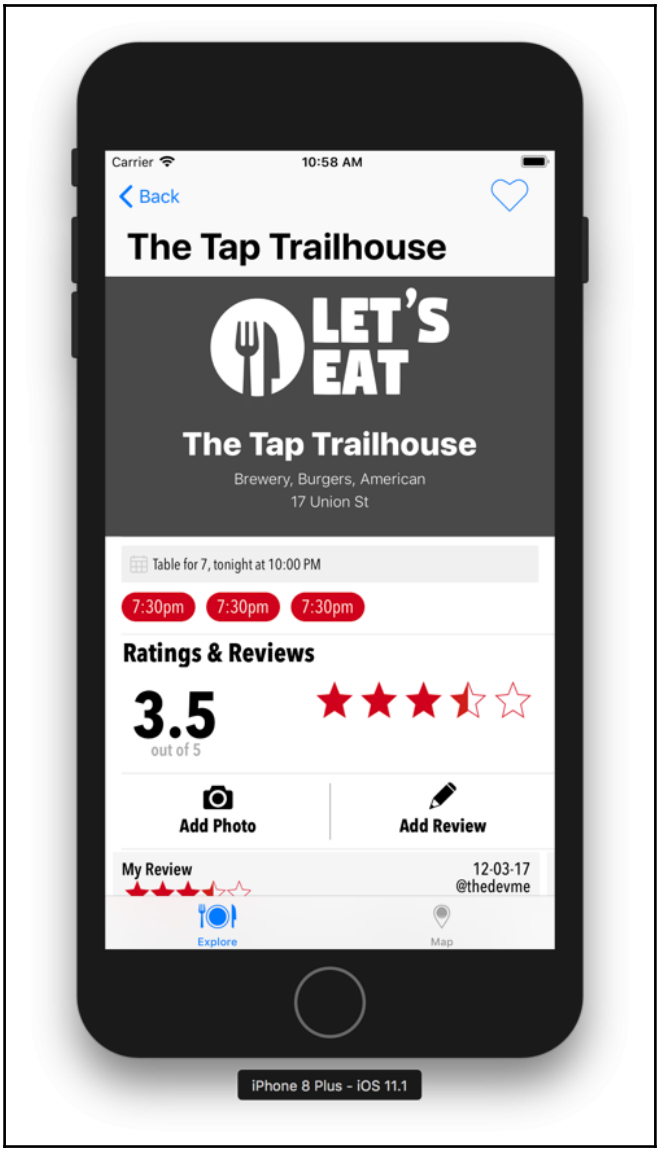

LetsEat
iPhone 7
LetsEat | Build LetsEat: Succeeded | Yesterday at 8:59 PM
LetsEat 〉 LetsEat 〉 Application 〉 Common 〉 Core Data 〉 LetsEatModel.xcdatamodeld 〉 LetsEatModel.xcdatamodel 〉 Review
ENTITIES
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Attributes
Attribute
Type
name
String
Relationships
Relationship
Destination
Inverse
Fetched Properties
Fetched Property
Predicate
Entity
Name
Review
Abstract Entity
Parent Entity
No Parent Entity
Class
Name
NSManagedObject
Module
Global namespace
Codegen
Manual/None
Indexes
No Content
Constraints
No Content
User Info
Key
Value
Versioning
Hash Modifier
Version Hash Modifier
Renaming ID
Renaming Identifier
No Matches
Filter
Outline Style
Add Entity
Add Attribute
Editor Style


ENTITIES
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Attributes
Attribute
Type
customerReview
String
date
Date
name
String
rating
Float
restaurantID
Integer 32
uuid
String
Relationships
Relationship
Destination
Inverse
Fetched Properties
Fetched Property
Predicate

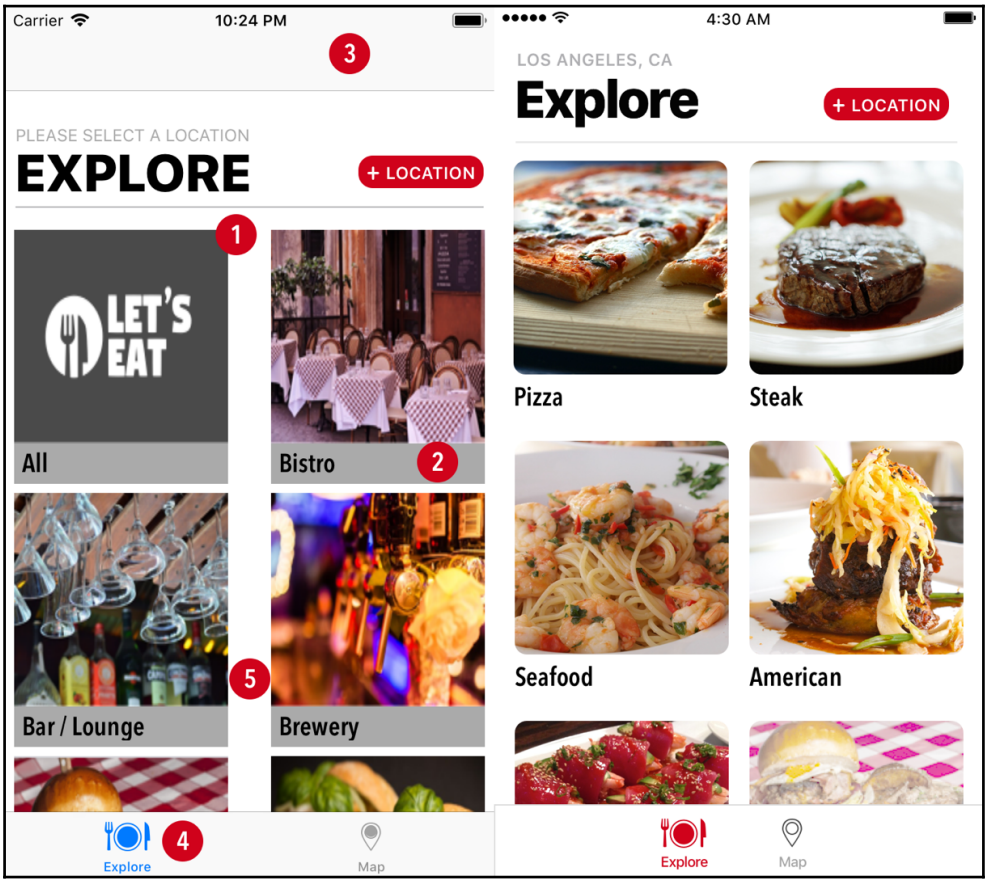

ENTITIES
RestaurantPhoto
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Attributes
Attribute
Type
date
Date
photo
Binary Data
restaurantID
Integer 32
uuid
String
Relationships
Relationship
Destination
Inverse
Fetched Properties
Fetched Property
Predicate

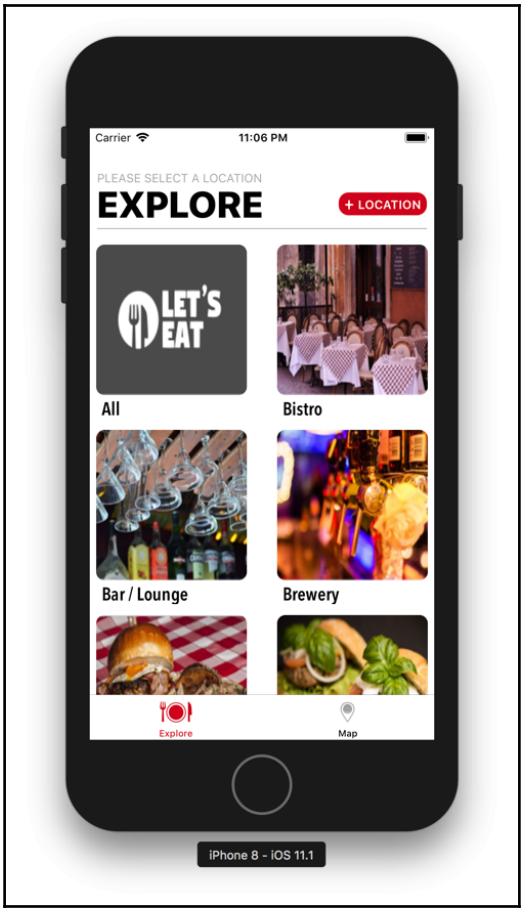
Chapter 22: Saving Reviews

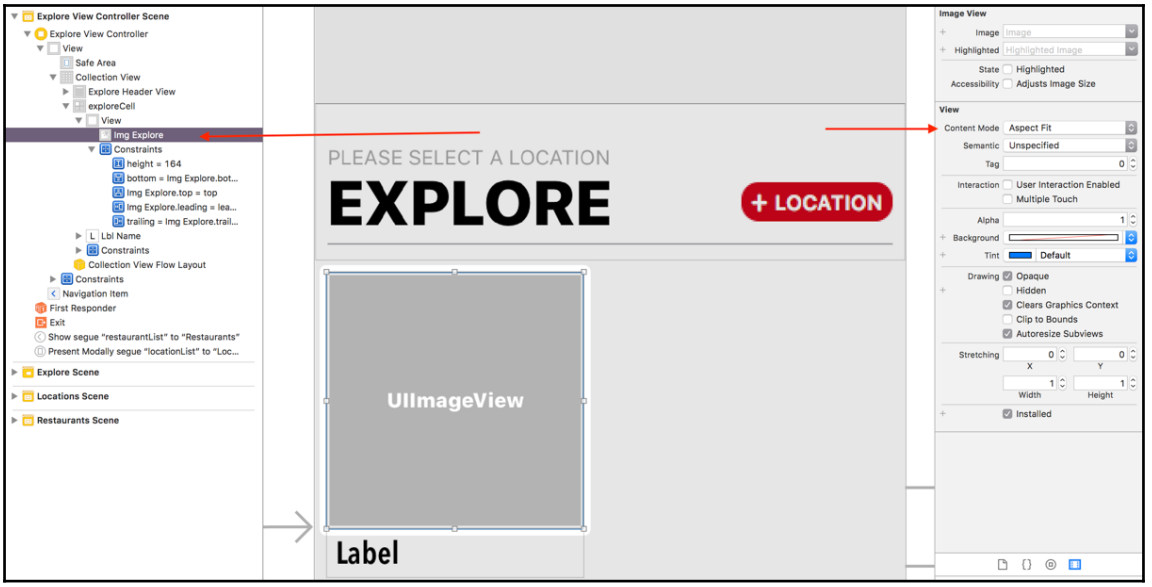

Carrier
10:36 AM
Back
The Tap Trailhouse
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
My Review
12-03-17
@thedevme
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec interdum nisi ut sollicitudin condimentum. Fusce hendrerit nunc est, id venenatis mi hendrerit vel. Cras porttitor efficitur urna, non tempor nisi commodo sed. Maecenas sed augue massa. Sed sed tortor volutpat, sodales arcu eu, vulputate ligula. Aenean vitae arcu ipsum.
Explore
Map
iPhone 8 Plus - iOS 11.1

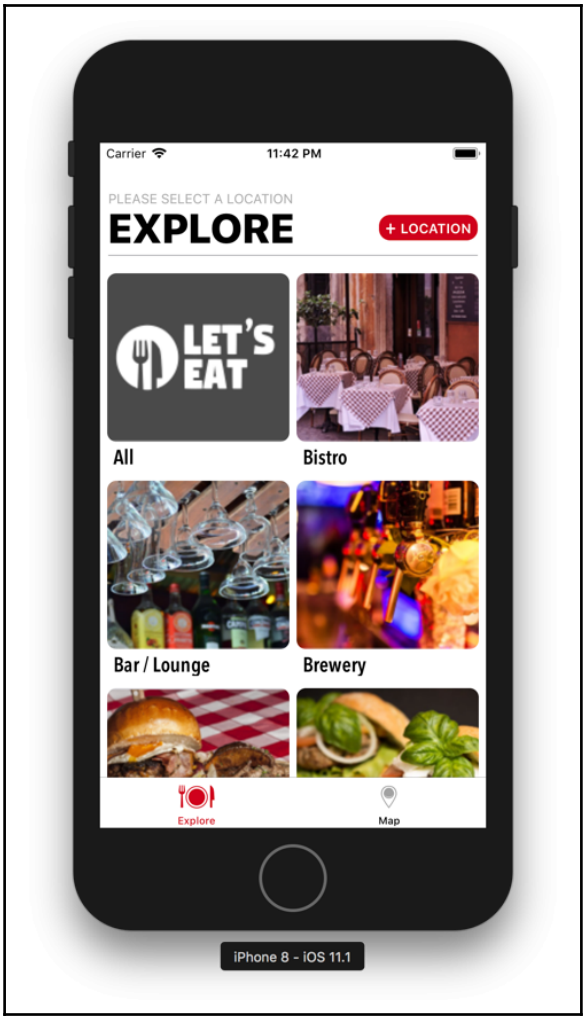

Carrier
10:58 AM
Back
The Tap Trailhouse
LET'S EAT
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
7:30pm
7:30pm
7:30pm
Ratings & Reviews
3.5
out of 5
Add Photo
Add Review
My Review
12-03-17
@thedevme
Explore
Map
iPhone 8 Plus - iOS 11.1

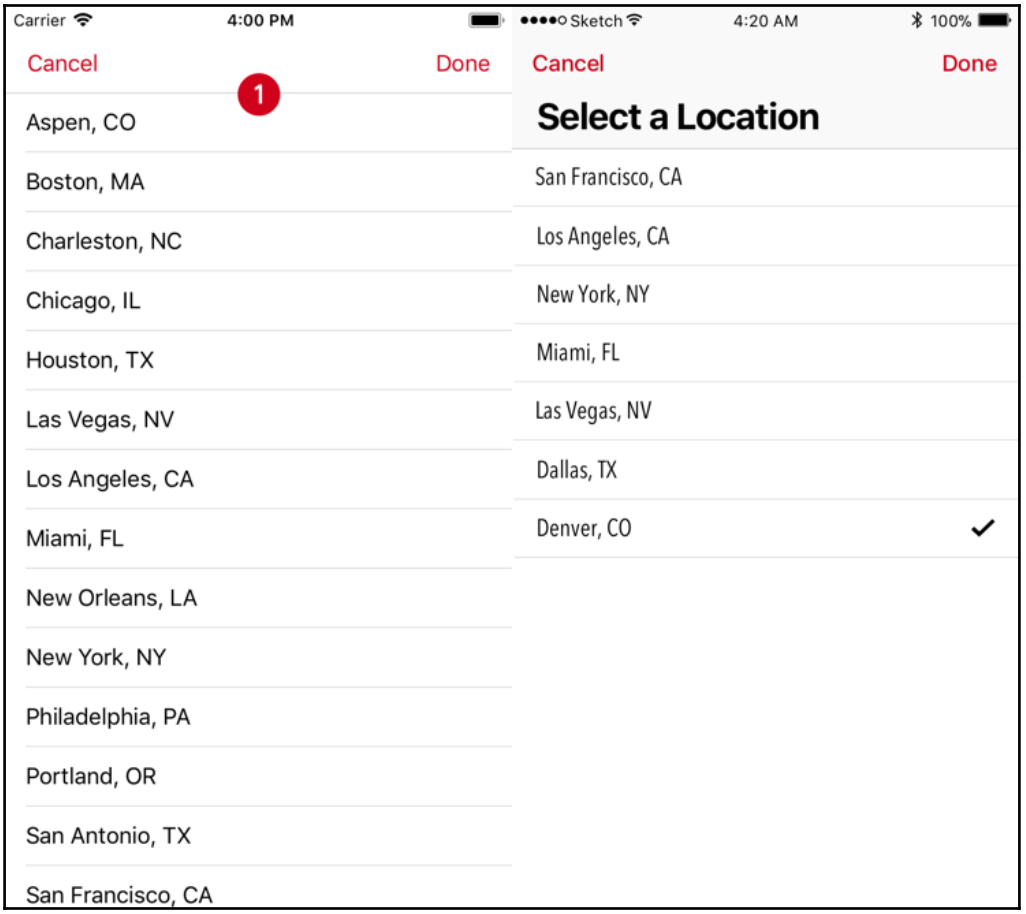
Chapter 23: Universal

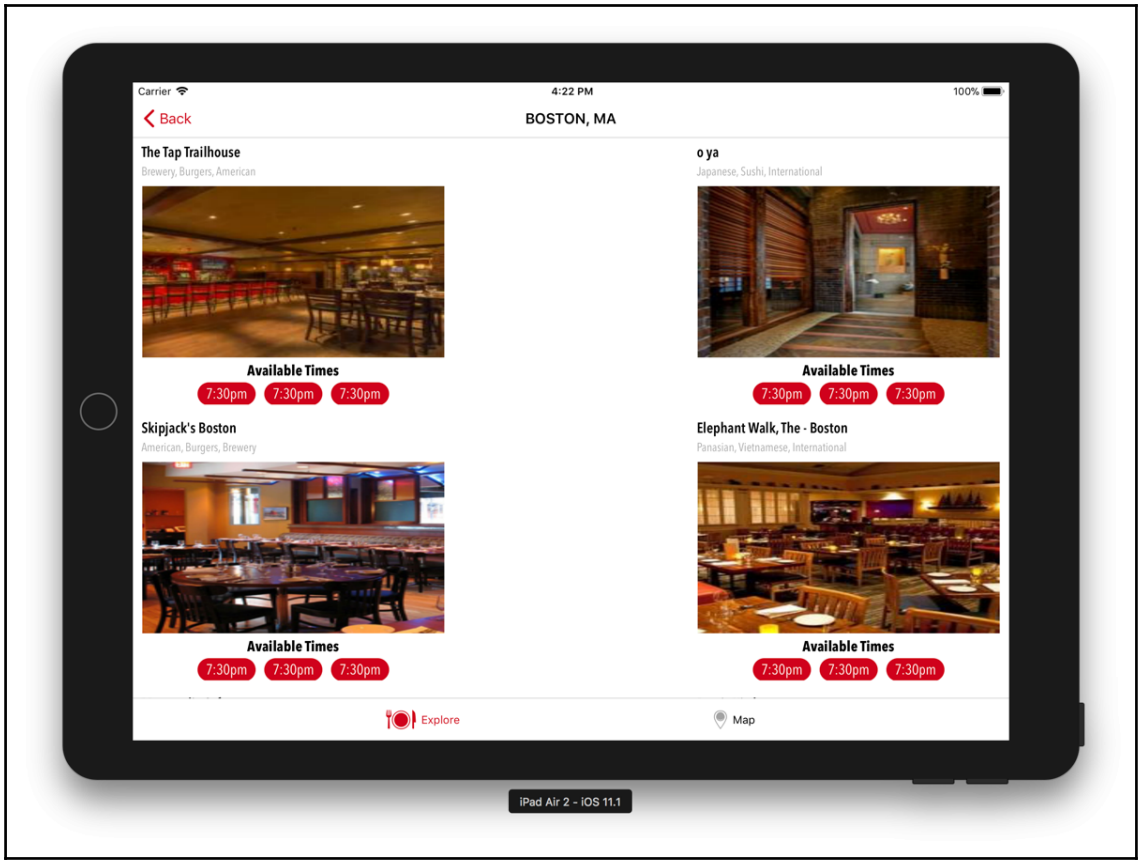


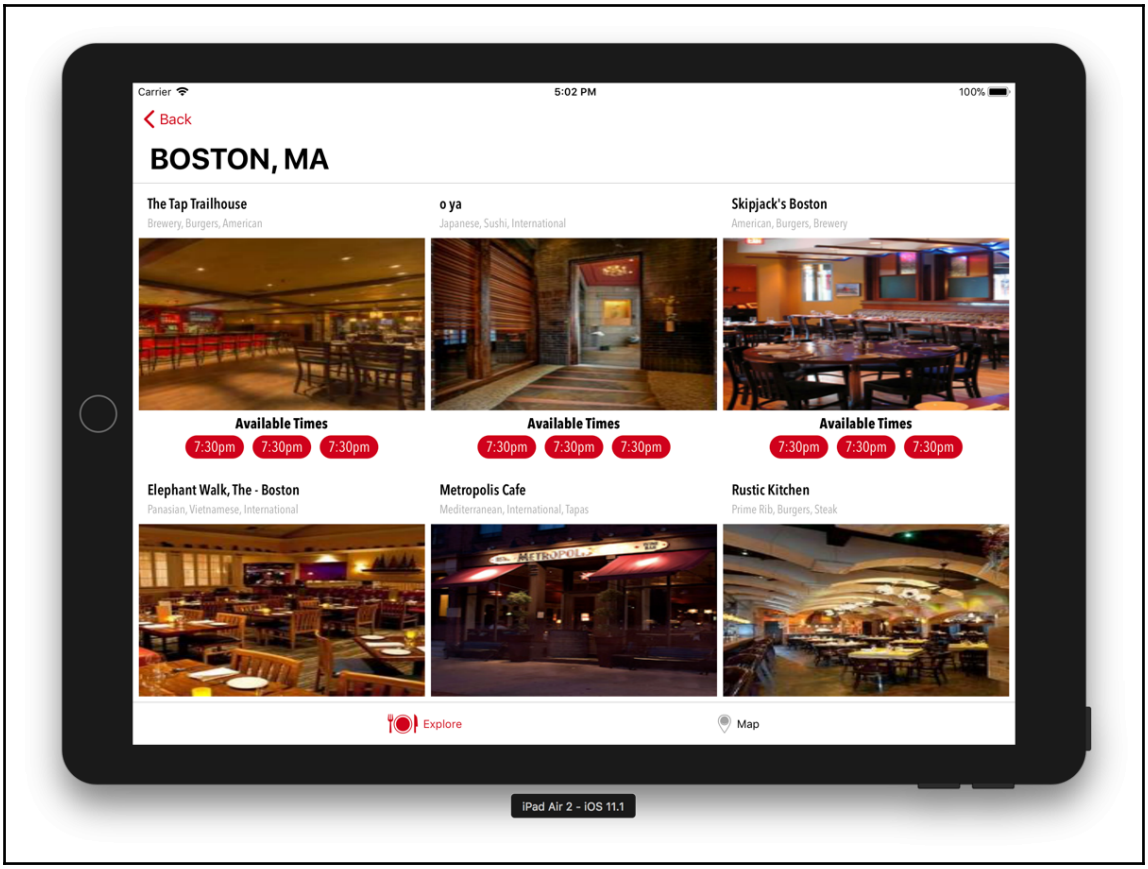

Carrier
11:06 PM
PLEASE SELECT A LOCATION
EXPLORE
+ LOCATION
LET'S EAT
All
Bistro
Bar / Lounge
Brewery
Explore
Map
iPhone 8 - iOS 11.1

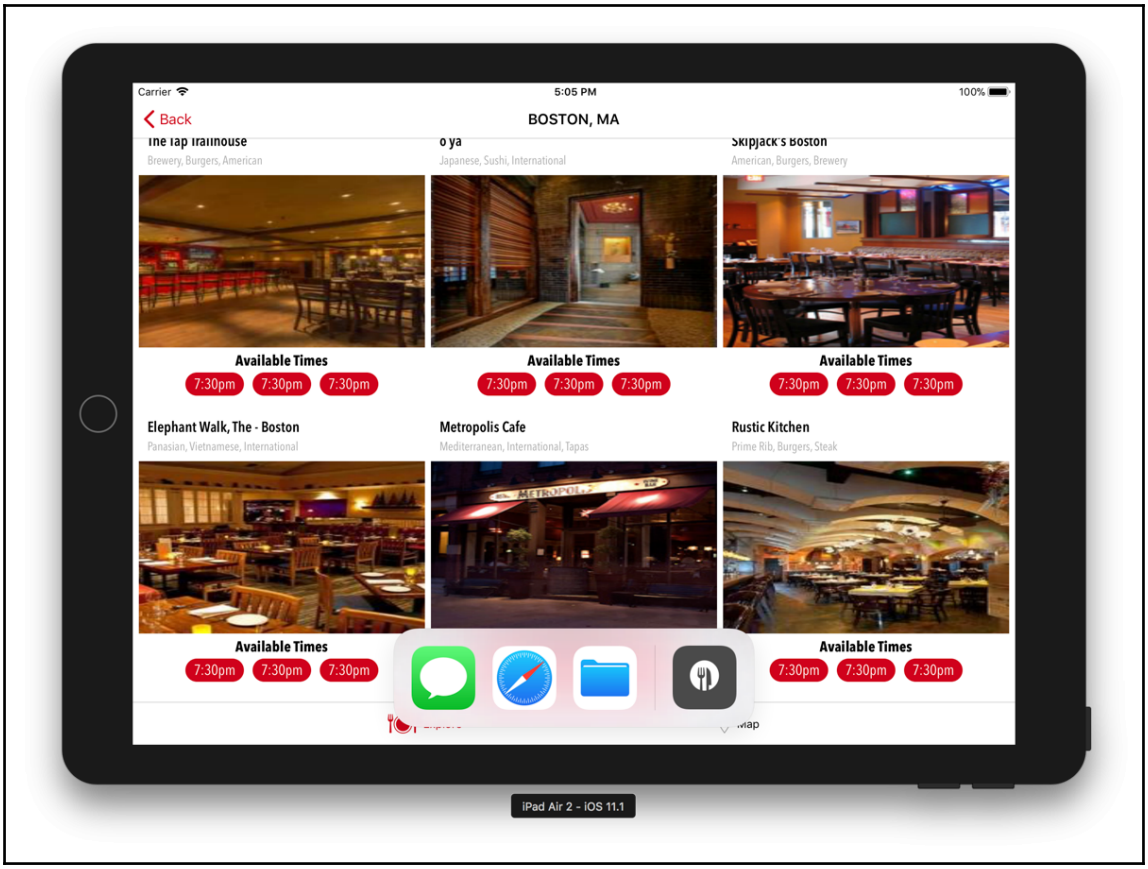

Explore View Controller Scene
Explore View Controller
View
Safe Area
Collection View
Explore Header View
exploreCell
View
Img Explore
Constraints
height = 164
bottom = Img Explore.bot...
Img Explore.top = top
Img Explore.leading = lea...
trailing = Img Explore.trail...
Lbl Name
Constraints
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Show segue "restaurantList" to "Restaurants"
Present Modally segue "locationList" to "Loc...
Explore Scene
Locations Scene
Restaurants Scene
PLEASE SELECT A LOCATION
EXPLORE
+ LOCATION
UIImageView
Label
Image View
Image
Highlighted
State
Highlighted
Accessibility
Adjusts Image Size
View
Content Mode
Aspect Fit
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip to Bounds
Autoresize Subviews
Stretching
X
Y
Width
Height
Installed

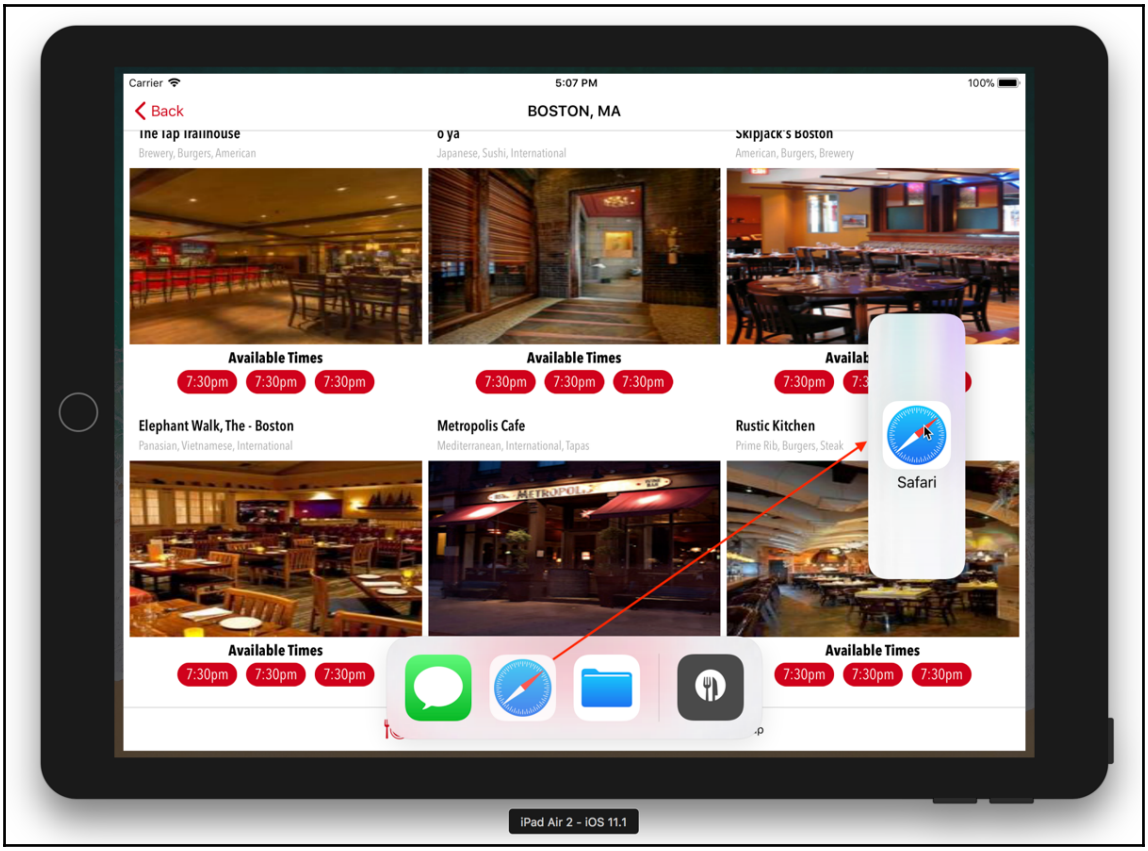

Carrier
11:42 PM
PLEASE SELECT A LOCATION
EXPLORE
+ LOCATION
LET'S EAT
All
Bistro
Bar / Lounge
Brewery
Explore
Map
iPhone 8 - iOS 11.1

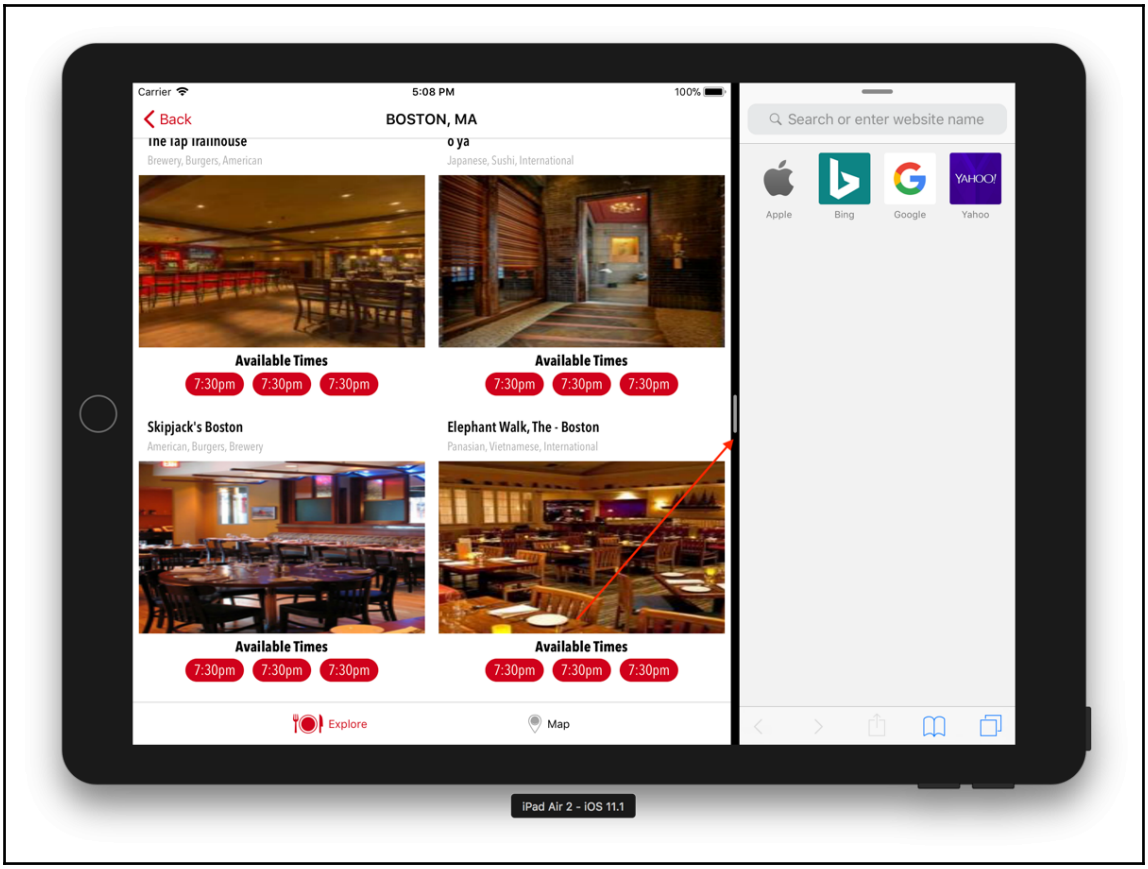

Carrier
4:00 PM
Cancel
Done
1
Aspen, CO
Boston, MA
Charleston, NC
Chicago, IL
Houston, TX
Las Vegas, NV
Los Angeles, CA
Miami, FL
New Orleans, LA
New York, NY
Philadelphia, PA
Portland, OR
San Antonio, TX
San Francisco, CA
Sketch
4:20 AM
100%
Cancel
Done
Select a Location
San Francisco, CA
Los Angeles, CA
New York, NY
Miami, FL
Las Vegas, NV
Dallas, TX
Denver, CO

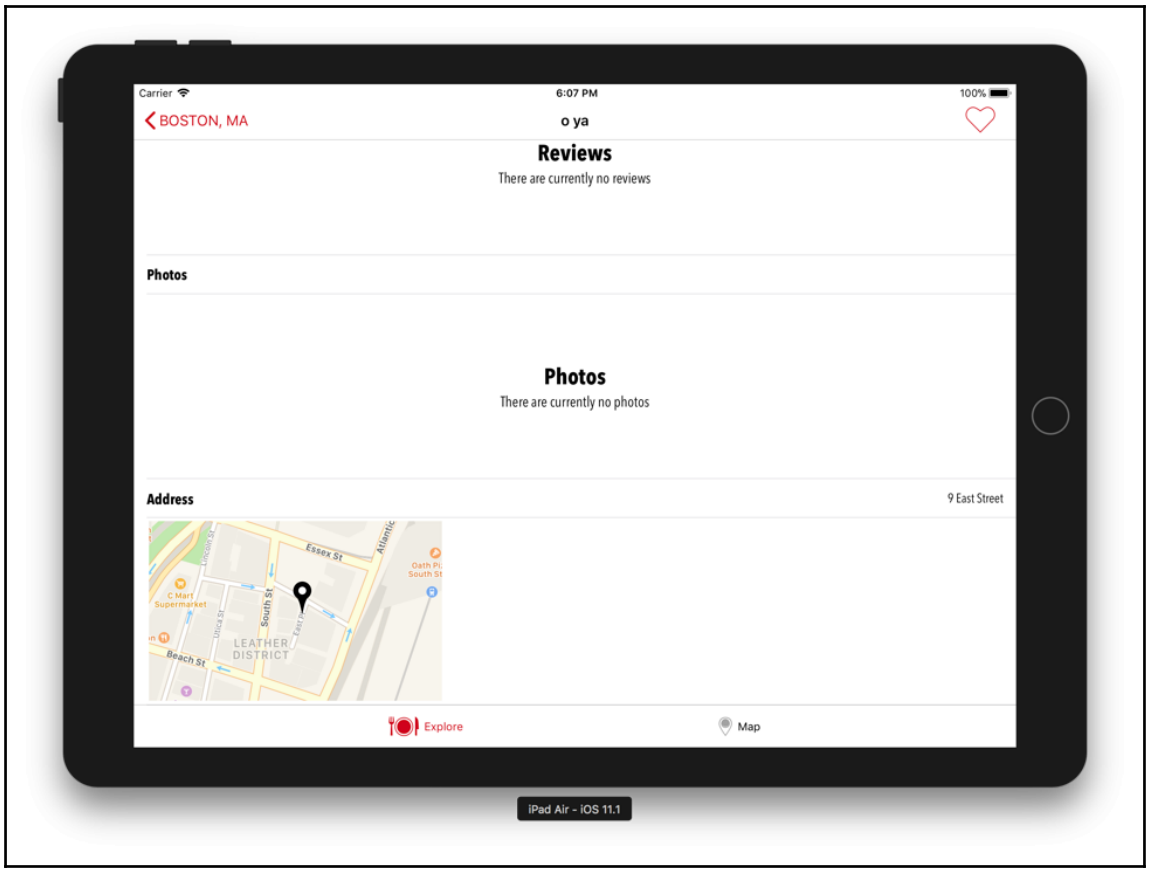

Carrier
4:22 PM
100%
Back
BOSTON, MA
The Tap Trailhouse
Brewery, Burgers, American
Available Times
7:30pm
7:30pm
7:30pm
o ya
Japanese, Sushi, International
Available Times
7:30pm
7:30pm
7:30pm
Skipjack's Boston
American, Burgers, Brewery
Available Times
7:30pm
7:30pm
7:30pm
Elephant Walk, The - Boston
Panasian, Vietnamese, International
Available Times
7:30pm
7:30pm
7:30pm
Explore
Map
iPad Air 2 - iOS 11.1


Carrier
5:02 PM
100%
Back
BOSTON, MA
The Tap Trailhouse
Brewery, Burgers, American
o ya
Japanese, Sushi, International
Skipjack's Boston
American, Burgers, Brewery
Available Times
7:30pm
7:30pm
7:30pm
Available Times
7:30pm
7:30pm
7:30pm
Available Times
7:30pm
7:30pm
7:30pm
Elephant Walk, The - Boston
Panasian, Vietnamese, International
Metropolis Cafe
Mediterranean, International, Tapas
Rustic Kitchen
Prime Rib, Burgers, Steak
Explore
Map
iPad Air 2 - iOS 11.1

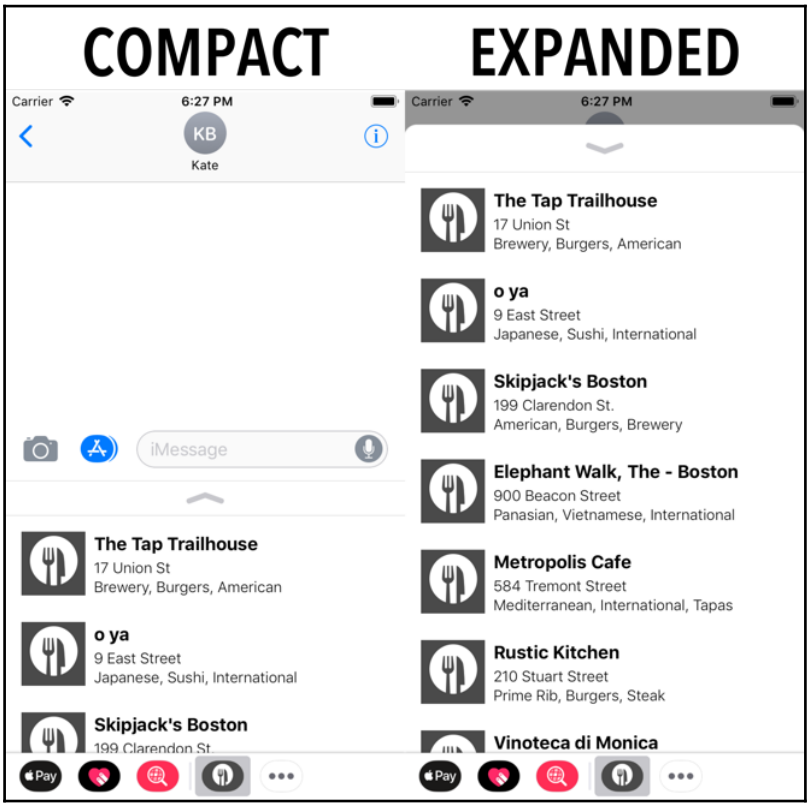

Carrier
5:05 PM
100%
Back
BOSTON, MA
Brewery, Burgers, American
Japanese, Sushi, International
American, Burgers, Brewery
Available Times
7:30pm
7:30pm
7:30pm
Elephant Walk, The - Boston
Panasian, Vietnamese, International
Metropolis Cafe
Mediterranean, International, Tapas
Rustic Kitchen
Prime Rib, Burgers, Steak
iPad Air 2 - iOS 11.1

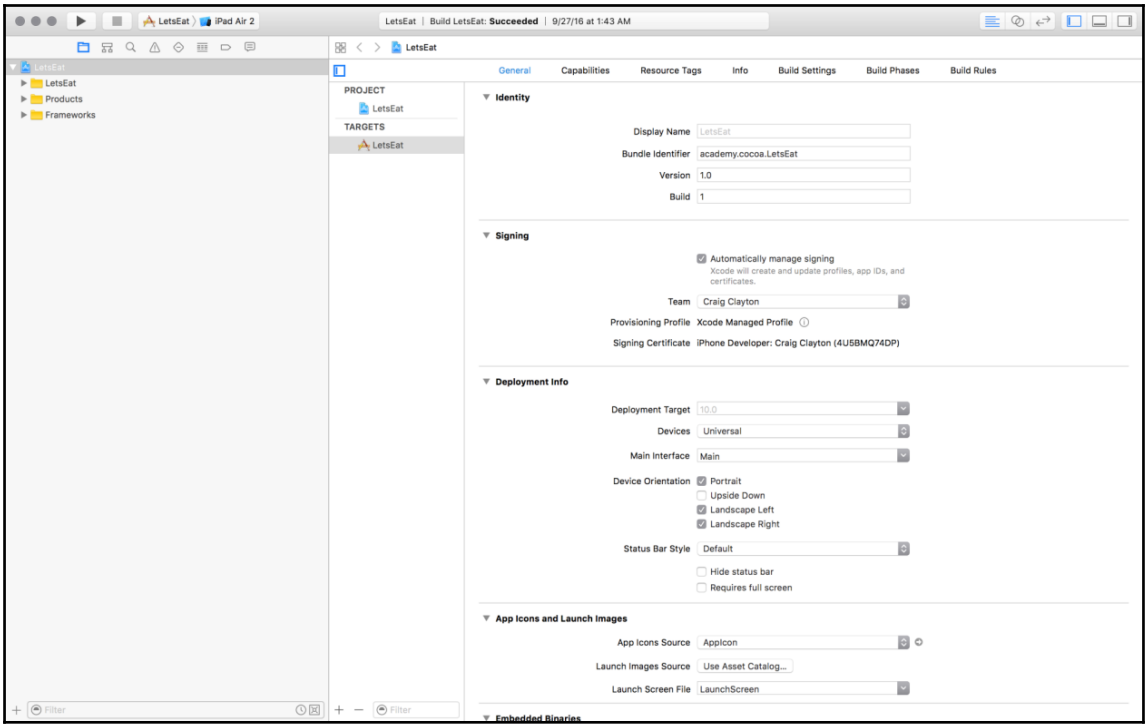

Carrier
5:07 PM
100%
Back
BOSTON, MA
Brewery, Burgers, American
Japanese, Sushi, International
American, Burgers, Brewery
Available Times
7:30pm
7:30pm
7:30pm
Elephant Walk, The - Boston
Panasian, Vietnamese, International
Metropolis Cafe
Mediterranean, International, Tapas
Rustic Kitchen
Prime Rib, Burgers, Steak
Safari
iPad Air 2 - iOS 11.1

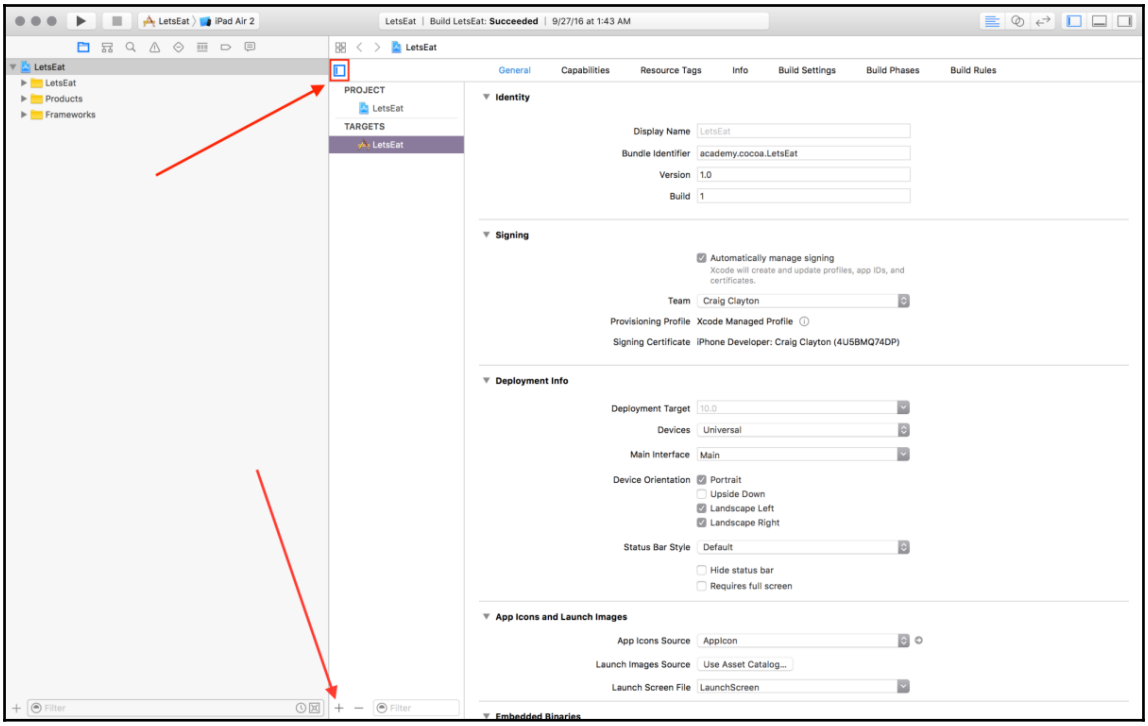

Carrier
5:08 PM
100%
Back
BOSTON, MA
The Tap Trailhouse
Brewery, Burgers, American
o ya
Japanese, Sushi, International
Available Times
7:30pm
7:30pm
7:30pm
Available Times
7:30pm
7:30pm
7:30pm
Skipjack's Boston
American, Burgers, Brewery
Elephant Walk, The - Boston
Panasian, Vietnamese, International
Available Times
7:30pm
7:30pm
7:30pm
Available Times
7:30pm
7:30pm
7:30pm
Explore
Map
Search or enter website name
Apple
Bing
Google
Yahoo
iPad Air 2 - iOS 11.1

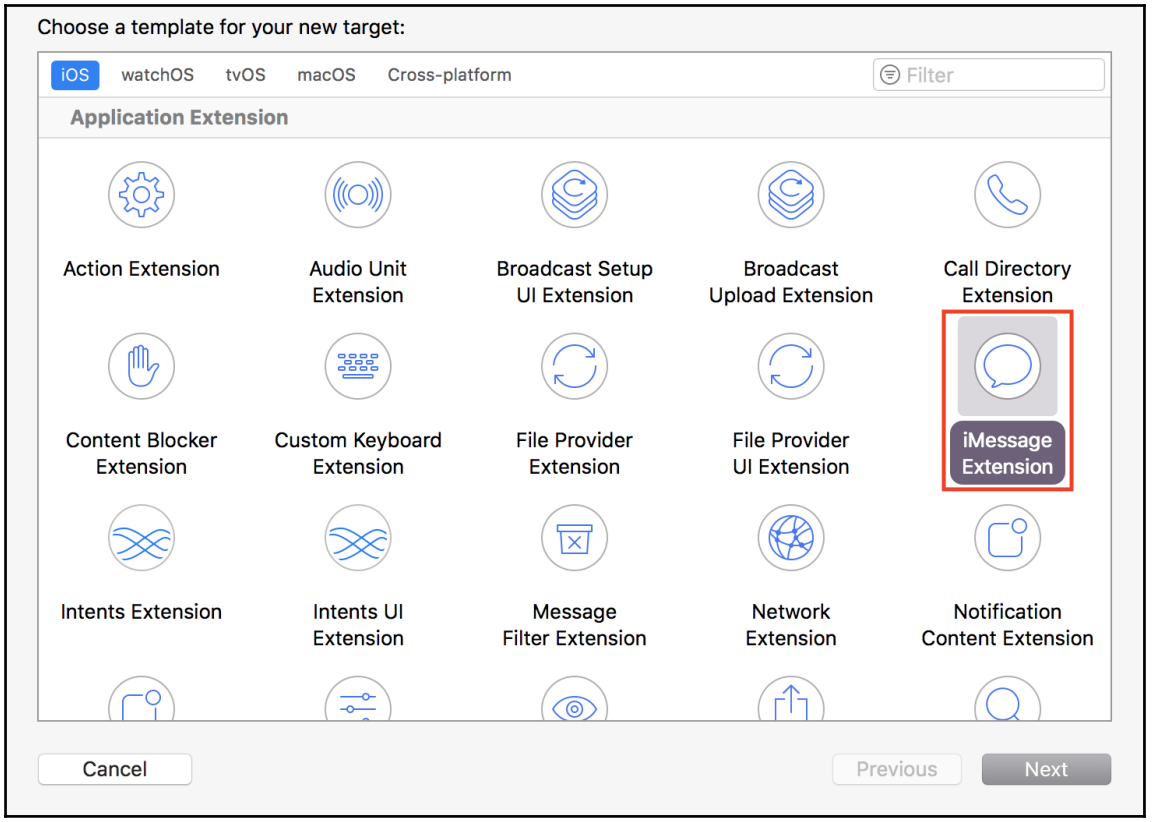

Carrier
6:07 PM
100%
BOSTON, MA
o ya
Reviews
There are currently no reviews
Photos
Photos
There are currently no photos
Address
9 East Street
Essex St
Atlantic
South St
Beach St
LEATHER DISTRICT
Explore
Map
iPad Air - iOS 11.1

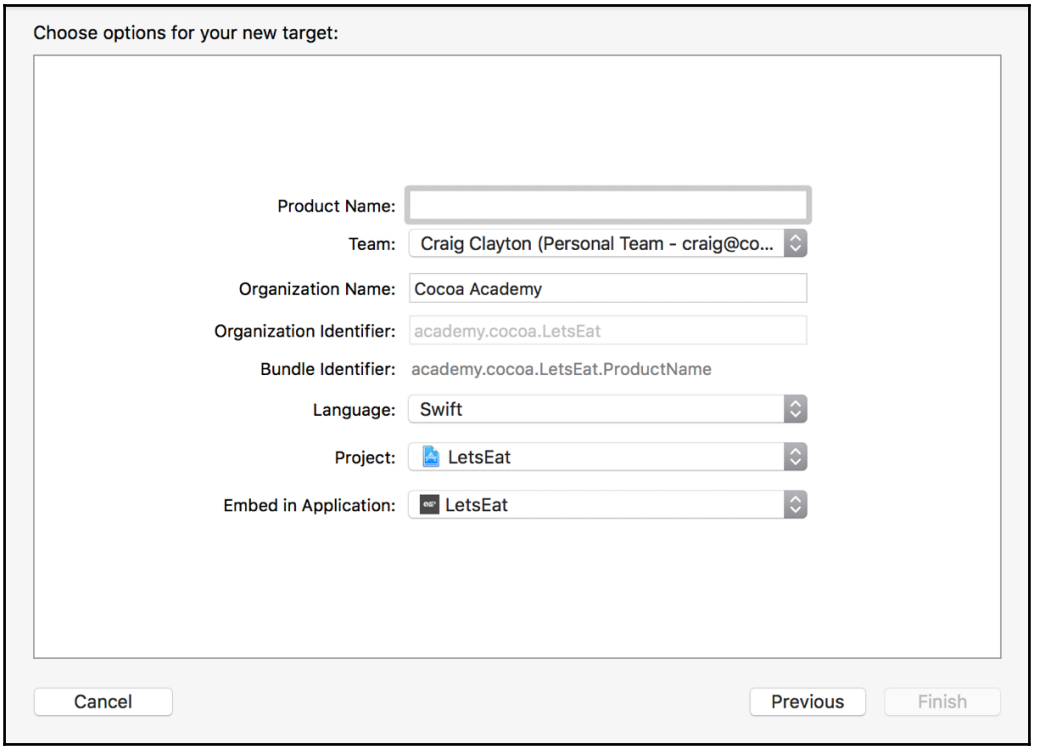
Chapter 24: iMessages

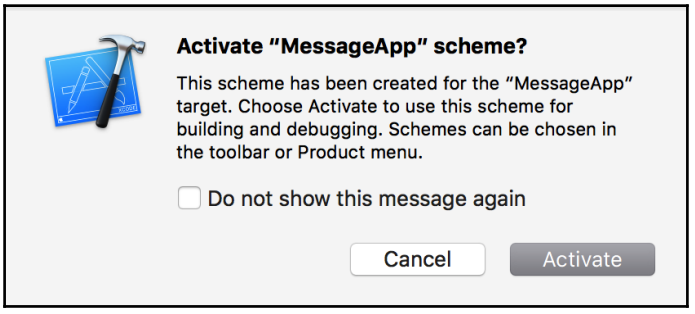


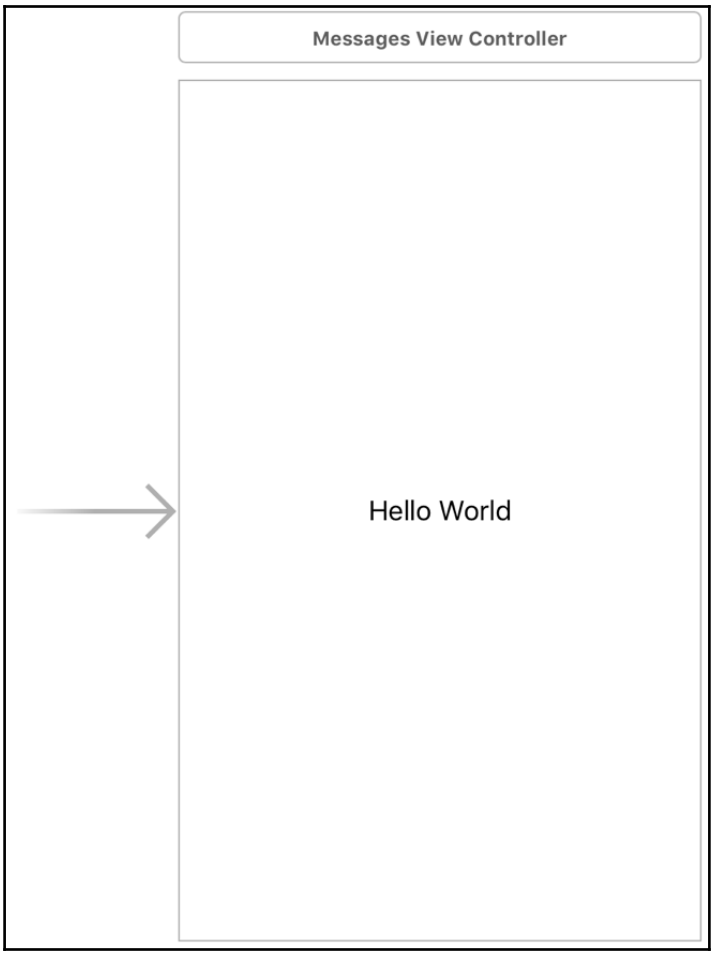

LetsEat
iPad Air 2
LetsEat | Build LetsEat: Succeeded | 9/27/16 at 1:43 AM
LetsEat
Products
Frameworks
PROJECT
TARGETS
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
Identity
Display Name
Bundle Identifier
academy.cocoa.LetsEat
Version
1.0
Build
1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
Craig Clayton
Provisioning Profile
Xcode Managed Profile
Signing Certificate
iPhone Developer: Craig Clayton (4U5BMQ74DP)
Deployment Info
Deployment Target
Devices
Universal
Main Interface
Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
App Icons and Launch Images
App Icons Source
AppIcon
Launch Images Source
Use Asset Catalog...
Launch Screen File
LaunchScreen
Embedded Binaries

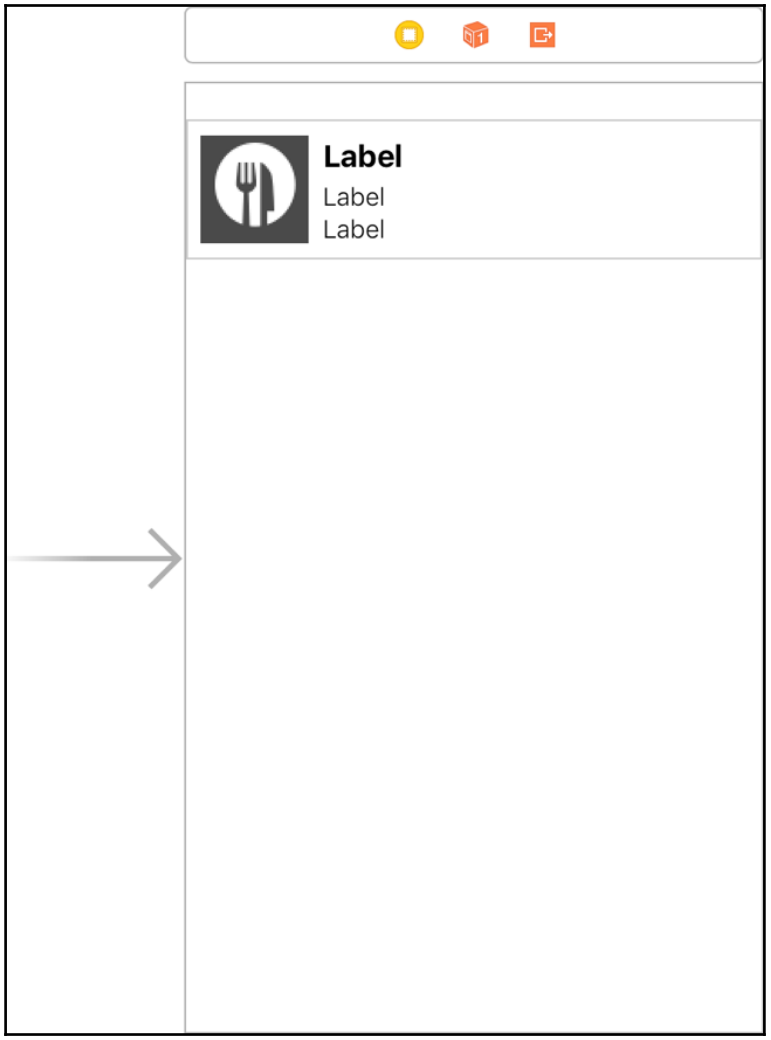

LetsEat
iPad Air 2
LetsEat | Build LetsEat: Succeeded | 9/27/16 at 1:43 AM
LetsEat
Products
Frameworks
PROJECT
TARGETS
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
Identity
Display Name
Bundle Identifier
academy.cocoa.LetsEat
Version
Build
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
Craig Clayton
Provisioning Profile
Xcode Managed Profile
Signing Certificate
iPhone Developer: Craig Clayton (4U5BMQ74DP)
Deployment Info
Deployment Target
Devices
Universal
Main Interface
Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
App Icons and Launch Images
App Icons Source
AppIcon
Launch Images Source
Use Asset Catalog...
Launch Screen File
LaunchScreen
Embedded Binaries

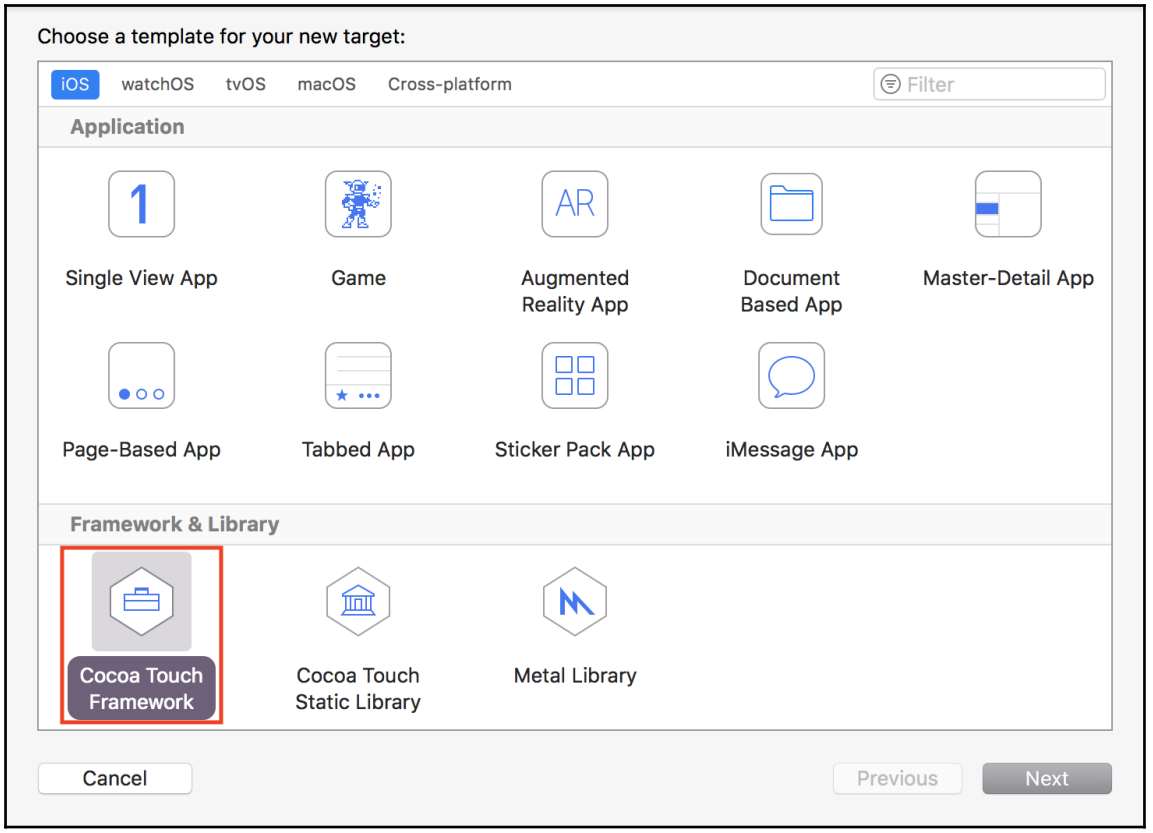

Choose a template for your new target:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application Extension
Action Extension
Audio Unit Extension
Broadcast Setup UI Extension
Broadcast Upload Extension
Call Directory Extension
Content Blocker Extension
Custom Keyboard Extension
File Provider Extension
File Provider UI Extension
iMessage Extension
Intents Extension
Intents UI Extension
Message Filter Extension
Network Extension
Notification Content Extension
Cancel
Previous
Next

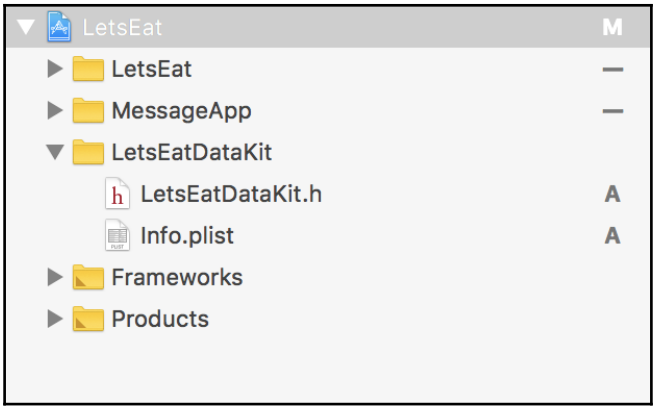

Choose options for your new target:
Product Name:
Team:
Craig Clayton (Personal Team - craig@co...
Organization Name:
Cocoa Academy
Organization Identifier:
academy.cocoa.LetsEat
Bundle Identifier:
academy.cocoa.LetsEat.ProductName
Language:
Swift
Project:
LetsEat
Embed in Application:
LetsEat
Cancel
Previous
Finish

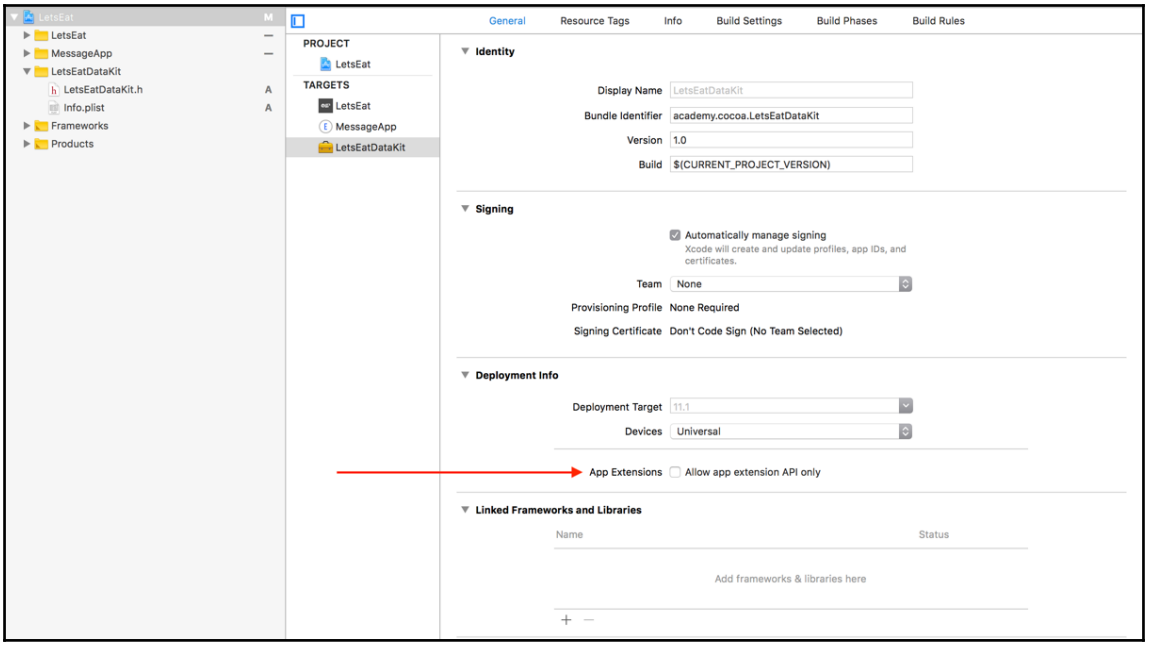

Activate "MessageApp" scheme?
This scheme has been created for the "MessageApp" target. Choose Activate to use this scheme for building and debugging. Schemes can be chosen in the toolbar or Product menu.
Do not show this message again
Cancel
Activate

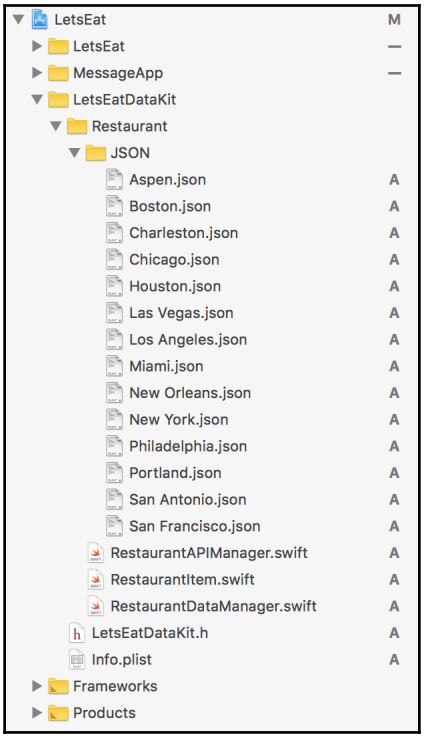

Messages View Controller
Hello World


Label
Label
Label

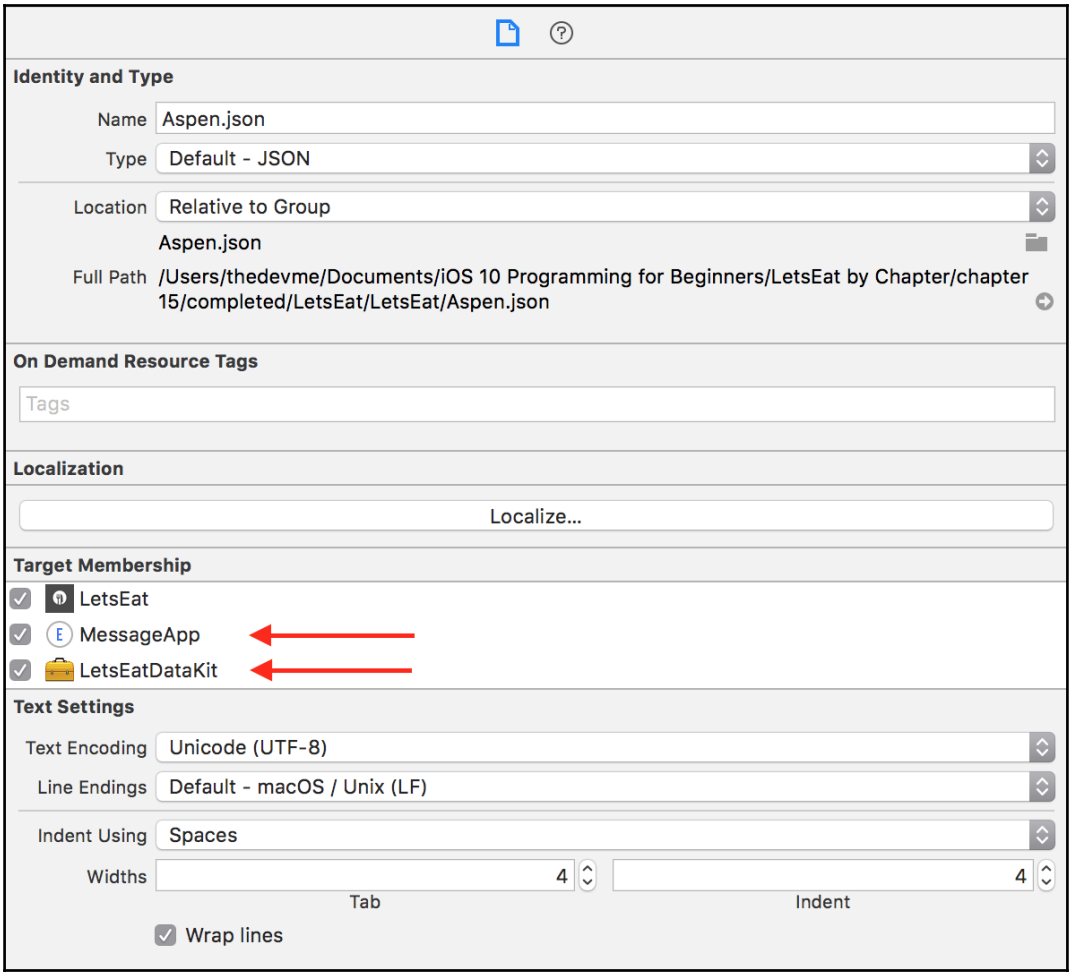

Choose a template for your new target:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application
Single View App
Game
Augmented Reality App
Document Based App
Master-Detail App
Page-Based App
Tabbed App
Sticker Pack App
iMessage App
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Metal Library
Cancel
Previous
Next

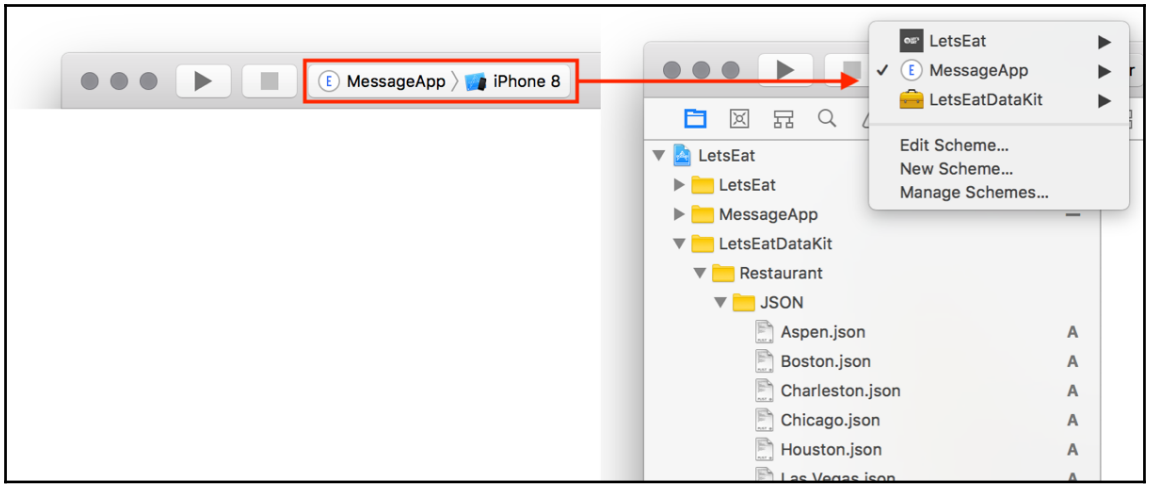

LetsEat
M
LetsEat
MessageApp
LetsEatDataKit
LetsEatDataKit.h
A
Info.plist
A
Frameworks
Products

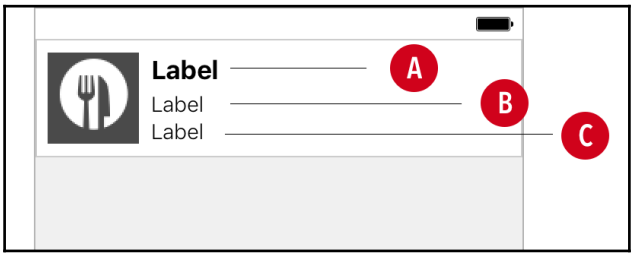

LetsEat
LetsEat
MessageApp
LetsEatDataKit
LetsEatDataKit.h
Info.plist
Frameworks
Products
General
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
LetsEat
TARGETS
LetsEat
MessageApp
LetsEatDataKit
Identity
Display Name
LetsEatDataKit
Bundle Identifier
academy.cocoa.LetsEatDataKit
Version
1.0
Build
$(CURRENT_PROJECT_VERSION)
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
None
Provisioning Profile
None Required
Signing Certificate
Don't Code Sign (No Team Selected)
Deployment Info
Deployment Target
11.1
Devices
Universal
App Extensions
Allow app extension API only
Linked Frameworks and Libraries
Name
Status
Add frameworks & libraries here

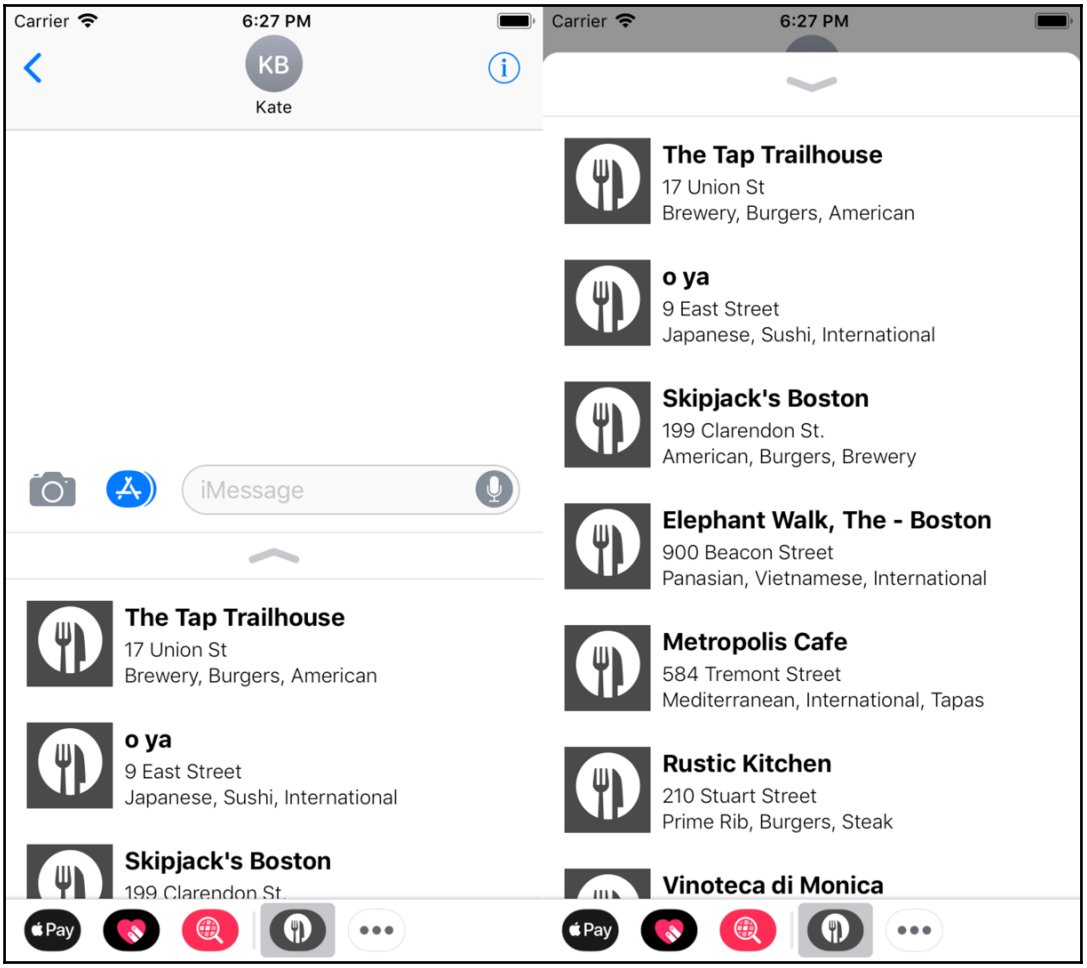

LetsEat M
LetsEat
MessageApp
LetsEatDataKit
Restaurant
JSON
Aspen.json A
Boston.json A
Charleston.json A
Chicago.json A
Houston.json A
Las Vegas.json A
Los Angeles.json A
Miami.json A
New Orleans.json A
New York.json A
Philadelphia.json A
Portland.json A
San Antonio.json A
San Francisco.json A
RestaurantAPIManager.swift A
RestaurantItem.swift A
RestaurantDataManager.swift A
LetsEatDataKit.h A
Info.plist A
Frameworks
Products

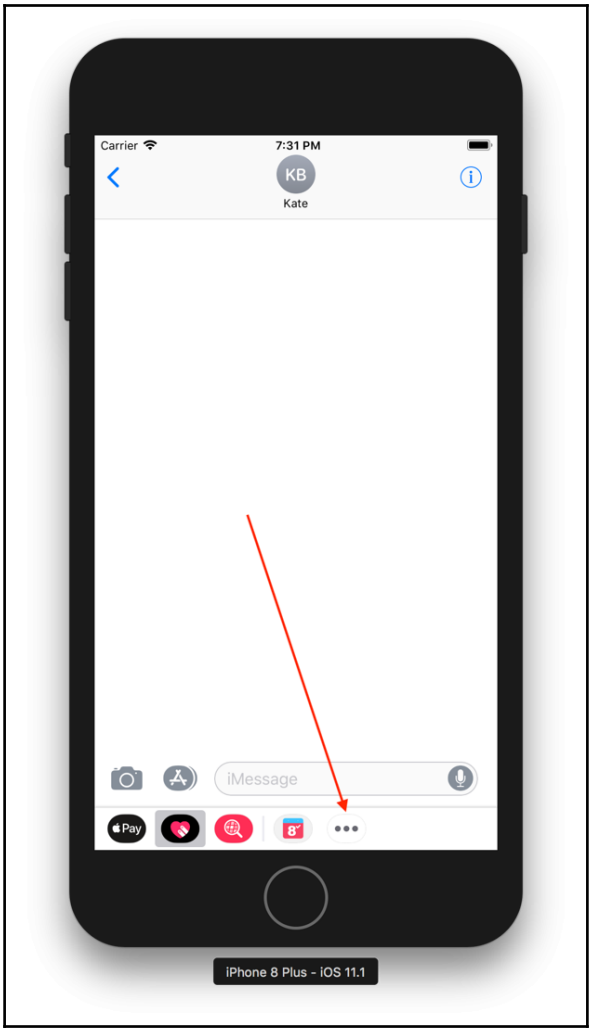

LetsEat › LetsEatDataKit › Restaurant › JSON › Aspen.json › No Selection
[
{
"id": 115903,
"name": "Square Grouper",
"address": "304 E Hopkins Ave",
"city": "Aspen",
"state": "CO",
"area": "Denver / Colorado",
"postal_code": "81611",
"country": "US",
"phone": "9704294968",
"lat": 39.190334,
"long": -106.820307,
"price": 2,
"cuisines":[
"Seafood",
"American"
],
"reserve_url": "http://www.opentable.com/single.aspx?rid=115903",
"mobile_reserve_url": "http://mobile.opentable.com/opentable/?
restId=115903",
"image_url": "https://www.opentable.com/img/restimages/115903.jpg"
},
Identity and Type
Name Aspen.json
Type Default - JSON
Location Relative to Group
Aspen.json
Full Path /Users/craigclayton/My Documents/Cocoa Academy/iOS 11 Projects/project files/chapter 24/completed/LetsEat/LetsEatDataKit/Restaurant/JSON/Aspen.json
On Demand Resource Tags
Localization
Localize...
Target Membership
LetsEat
MessageApp
LetsEatDataKit
Text Settings
Text Encoding Unicode (UTF-8)
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines

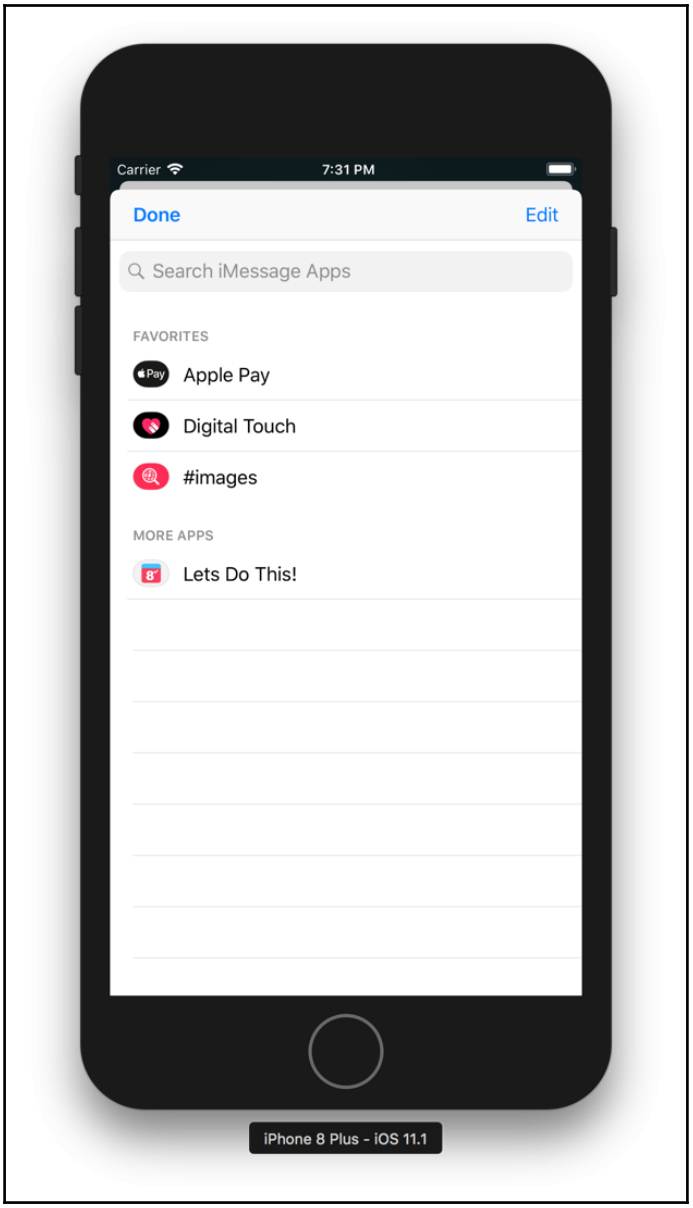

Identity and Type
Name
Aspen.json
Type
Default - JSON
Location
Relative to Group
Aspen.json
Full Path
/Users/thedevme/Documents/iOS 10 Programming for Beginners/LetsEat by Chapter/chapter 15/completed/LetsEat/LetsEat/Aspen.json
On Demand Resource Tags
Tags
Localization
Localize...
Target Membership
LetsEat
MessageApp
LetsEatDataKit
Text Settings
Text Encoding
Unicode (UTF-8)
Line Endings
Default - macOS / Unix (LF)
Indent Using
Spaces
Widths
4
Tab
4
Indent
Wrap lines

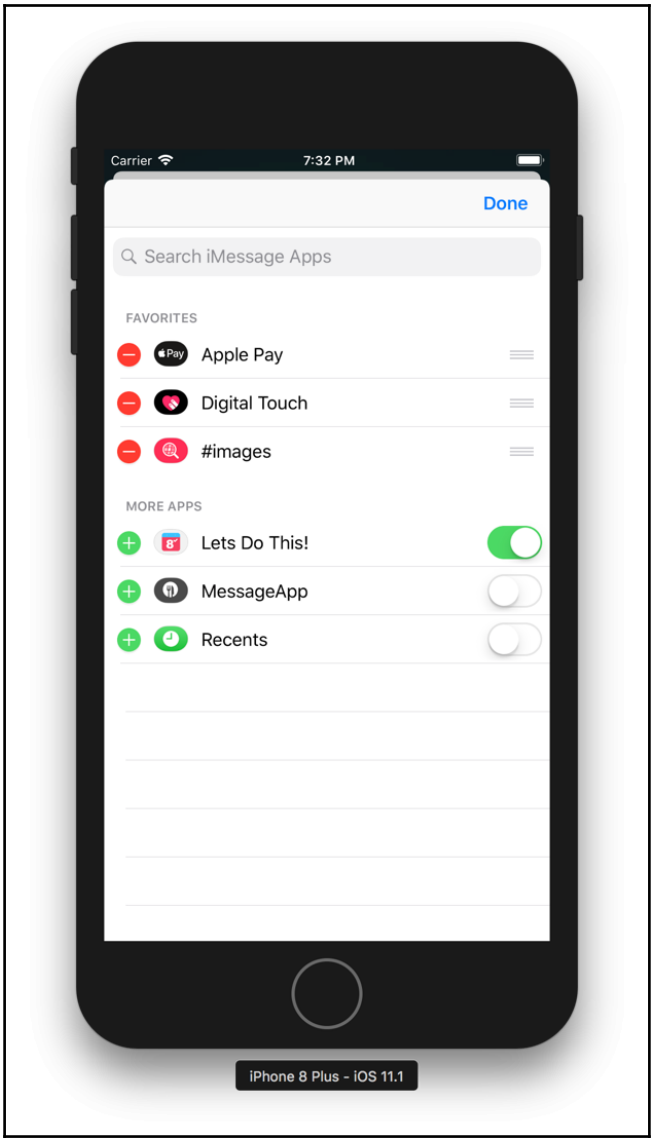

MessageApp
iPhone 8
LetsEat
MessageApp
LetsEatDataKit
Edit Scheme...
New Scheme...
Manage Schemes...
LetsEat
LetsEat
MessageApp
LetsEatDataKit
Restaurant
JSON
Aspen.json
Boston.json
Charleston.json
Chicago.json
Houston.json


Label
Label
Label
A
B
C

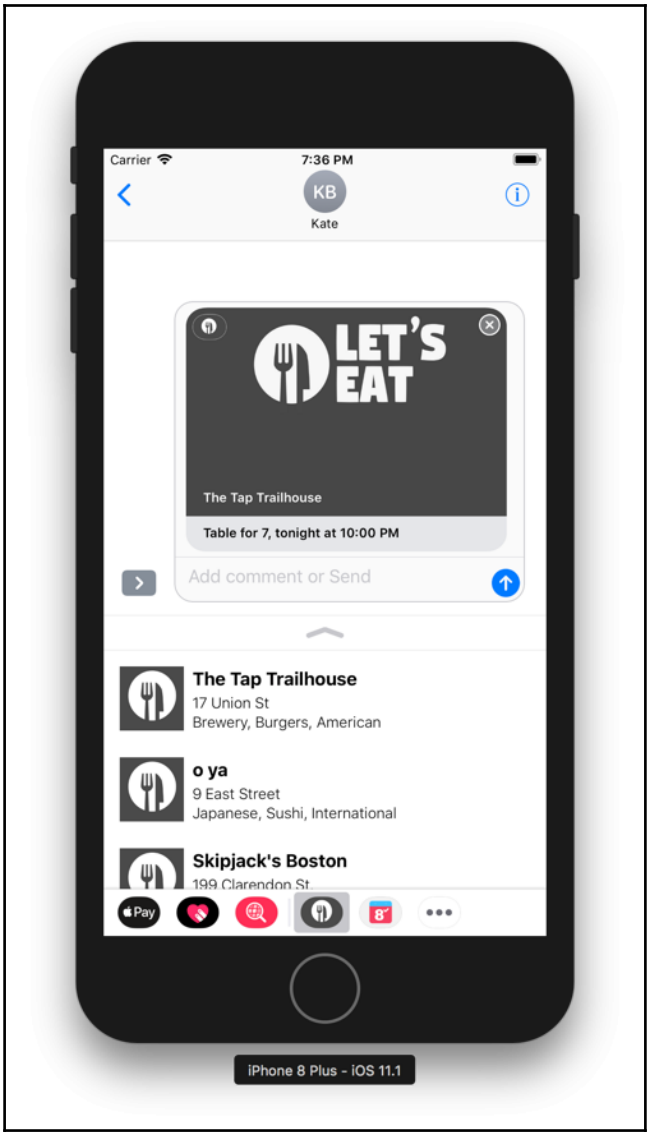

Carrier
6:27 PM
KB
Kate
iMessage
The Tap Trailhouse
17 Union St
Brewery, Burgers, American
o ya
9 East Street
Japanese, Sushi, International
Skipjack's Boston
199 Clarendon St.
American, Burgers, Brewery
Elephant Walk, The - Boston
900 Beacon Street
Panasian, Vietnamese, International
Metropolis Cafe
584 Tremont Street
Mediterranean, International, Tapas
Rustic Kitchen
210 Stuart Street
Prime Rib, Burgers, Steak
Vinoteca di Monica
Pay


Carrier
7:31 PM
KB
Kate
iMessage
iPhone 8 Plus - iOS 11.1

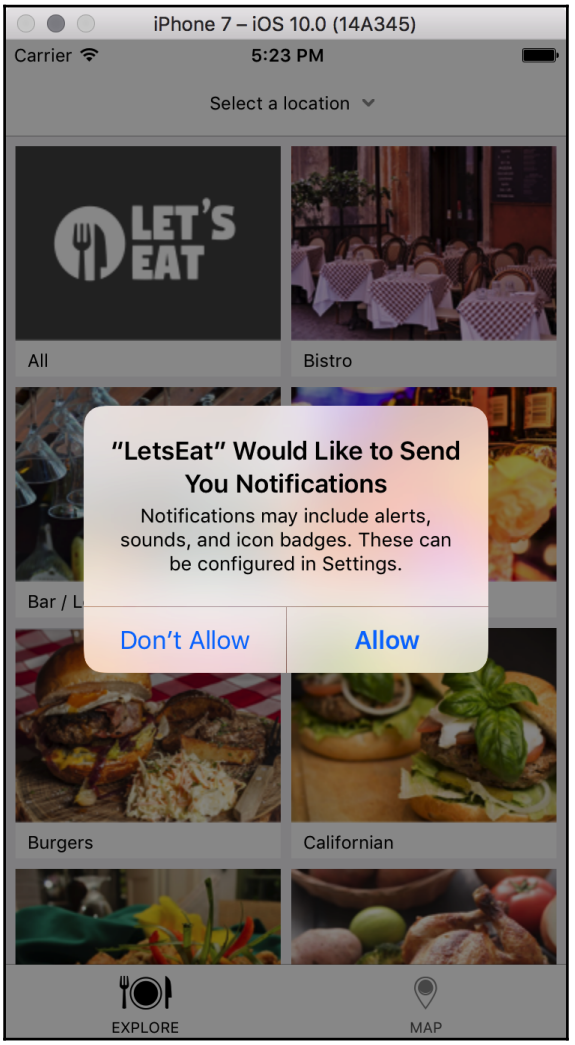

Carrier
7:31 PM
Done
Edit
Search iMessage Apps
FAVORITES
Apple Pay
Digital Touch
#images
MORE APPS
Lets Do This!
iPhone 8 Plus - iOS 11.1

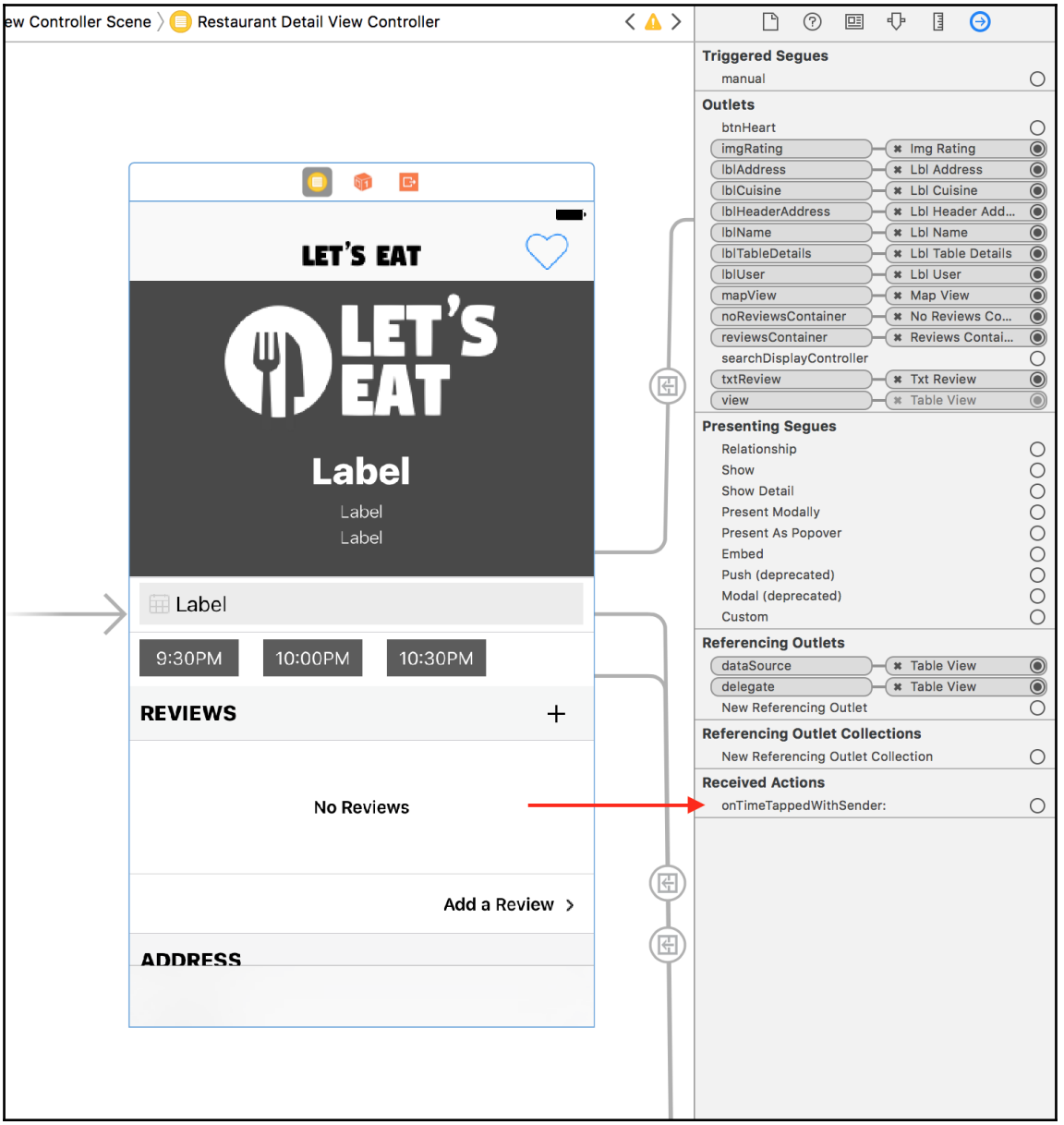

Carrier
7:32 PM
Done
Search iMessage Apps
FAVORITES
Apple Pay
Digital Touch
#images
MORE APPS
Lets Do This!
MessageApp
Recents
iPhone 8 Plus - iOS 11.1


Image, Audio, or Video
Image Title
Image Subtitle
Caption
Subcaption
Trailing Caption
Trailing Subcaption

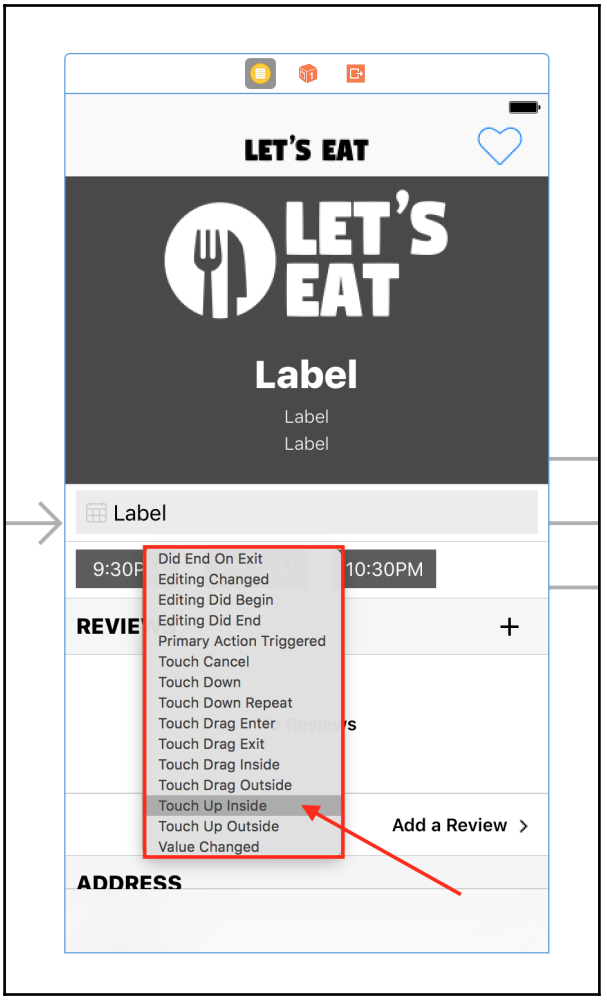

Carrier
7:36 PM
KB
Kate
LET'S EAT
The Tap Trailhouse
Table for 7, tonight at 10:00 PM
Add comment or Send
The Tap Trailhouse
17 Union St
Brewery, Burgers, American
o ya
9 East Street
Japanese, Sushi, International
Skipjack's Boston
199 Clarendon St
iPhone 8 Plus - iOS 11.1

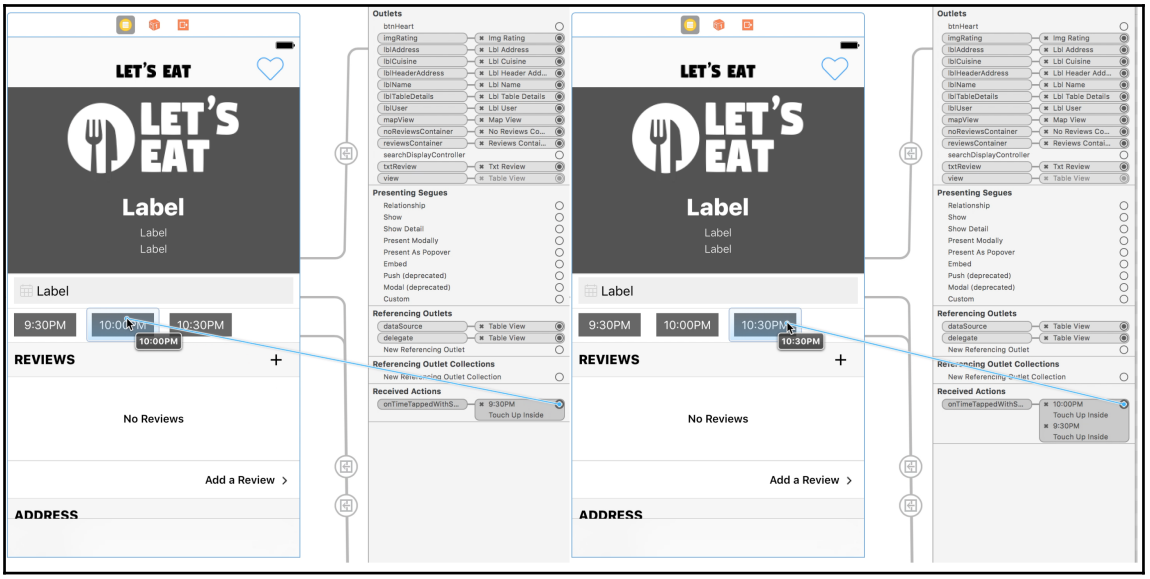
Chapter 25: Notifications



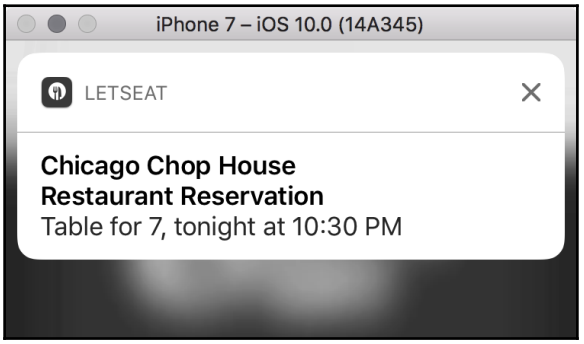

ew Controller Scene
Restaurant Detail View Controller
LET'S EAT
LET'S EAT
Label
Label
Label
Label
9:30PM
10:00PM
10:30PM
REVIEWS
No Reviews
Add a Review
ADDRESS
Triggered Segues
manual
Outlets
btnHeart
imgRating
Img Rating
lblAddress
Lbl Address
lblCuisine
Lbl Cuisine
lblHeaderAddress
Lbl Header Add...
lblName
Lbl Name
lblTableDetails
Lbl Table Details
lblUser
Lbl User
mapView
Map View
noReviewsContainer
No Reviews Co...
reviewsContainer
Reviews Contai...
searchDisplayController
txtReview
Txt Review
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onTimeTappedWithSender:


LET'S EAT
LET'S EAT
Label
Label
Label
Label
9:30PM
10:00PM
10:30PM
9:30PM
REVIEWS
No Reviews
Add a Review
ADDRESS
Outlets
btnHeart
imgRating
Img Rating
lblAddress
Lbl Address
lblCuisine
Lbl Cuisine
lblHeaderAddress
Lbl Header Add...
lblName
Lbl Name
lblTableDetails
Lbl Table Details
lblUser
Lbl User
mapView
Map View
noReviewsContainer
No Reviews Co...
reviewsContainer
Reviews Contai...
searchDisplayController
txtReview
Txt Review
view
Table View
Presenting Segues
Relationship
Show
Show Detail
Present Modally
Present As Popover
Embed
Push (deprecated)
Modal (deprecated)
Custom
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
onTimeTappedWithSender:

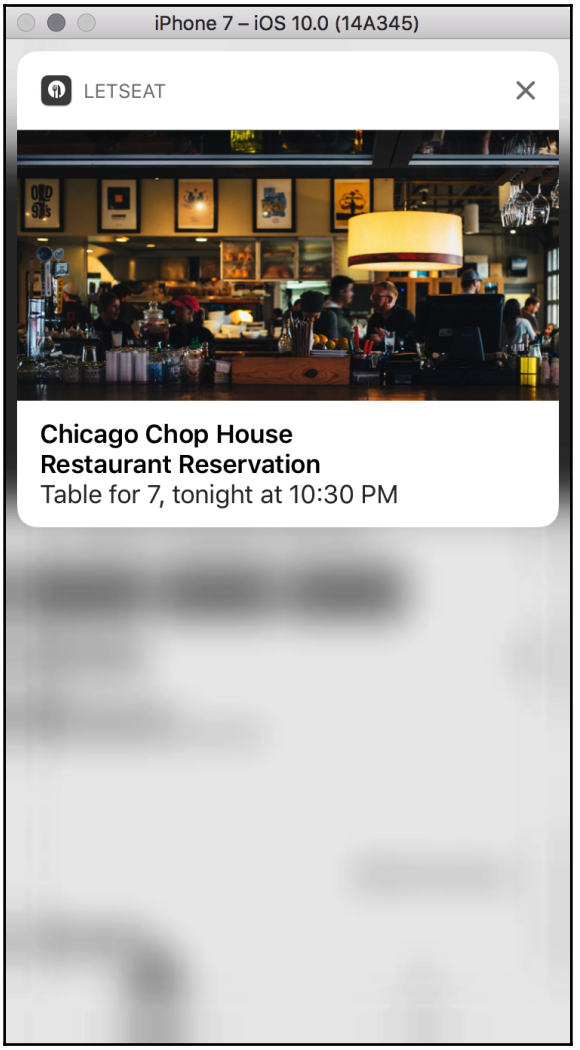

LET'S EAT
LET'S EAT
Label
Label
Label
Label
9:30P
10:30PM
REVIE
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Primary Action Triggered
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up Inside
Touch Up Outside
Value Changed
Add a Review
ADDRESS

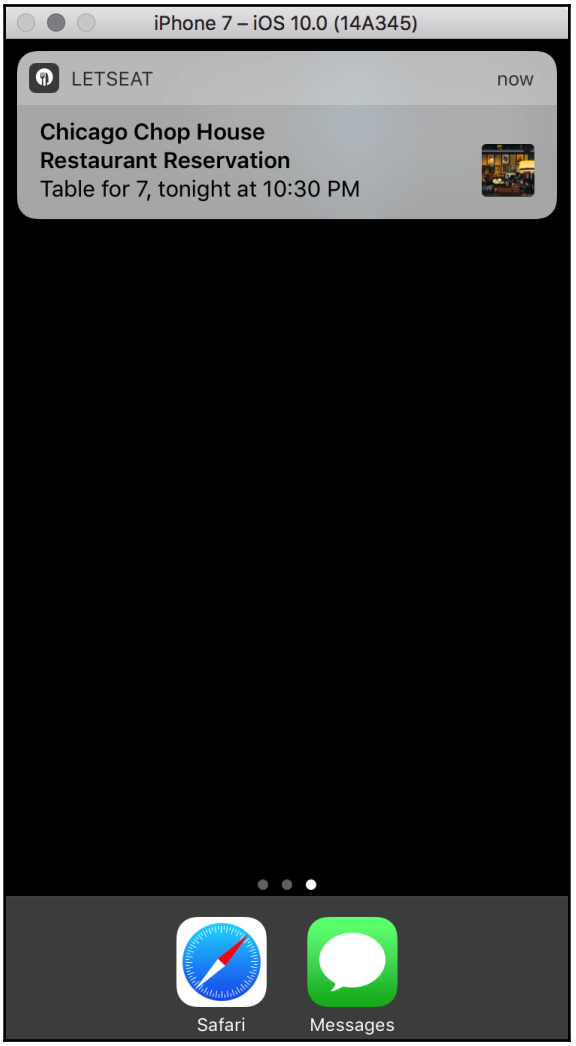

LET'S EAT
LET'S EAT
Label
Label
Label
Label
9:30PM
10:00PM
10:30PM
REVIEWS
No Reviews
Add a Review
ADDRESS
Outlets
Presenting Segues
Referencing Outlets
Referencing Outlet Collections
Received Actions
LET'S EAT
LET'S EAT
Label
Label
Label
Label
9:30PM
10:00PM
10:30PM
REVIEWS
No Reviews
Add a Review
ADDRESS
Outlets
Presenting Segues
Referencing Outlets
Referencing Outlet Collections
Received Actions

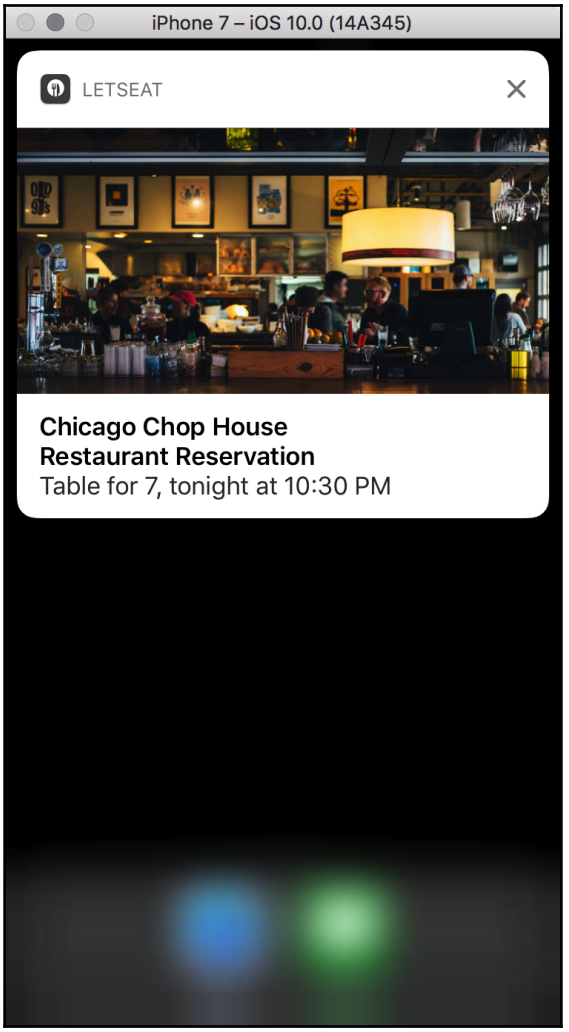

iPhone 7 – iOS 10.0 (14A345)
LETSEAT
now
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:30 PM
EAT
Chicago Chop House
Seafood, Steak
60 W Ontario St
Table for 7, tonight at 10:00 PM
9:30 PM
10:00 PM
10:30 PM
REVIEWS
bigpoppa
Read all reviews
ADDRESS
EXPLORE
MAP


iPhone 7 – iOS 10.0 (14A345)
LETSEAT
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:30 PM

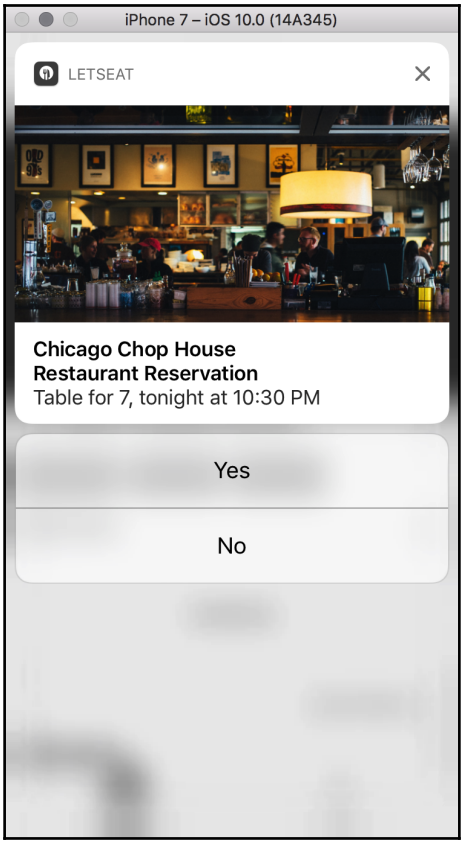

iPhone 7 – iOS 10.0 (14A345)
LETSEAT
now
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:30 PM
EAT
Chicago Chop House
Seafood, Steak
60 W Ontario St
Table for 7, tonight at 10:00 PM
9:30 PM
10:00 PM
10:30 PM
REVIEWS
bigpoppa
Read all reviews
ADDRESS
EXPLORE
MAP

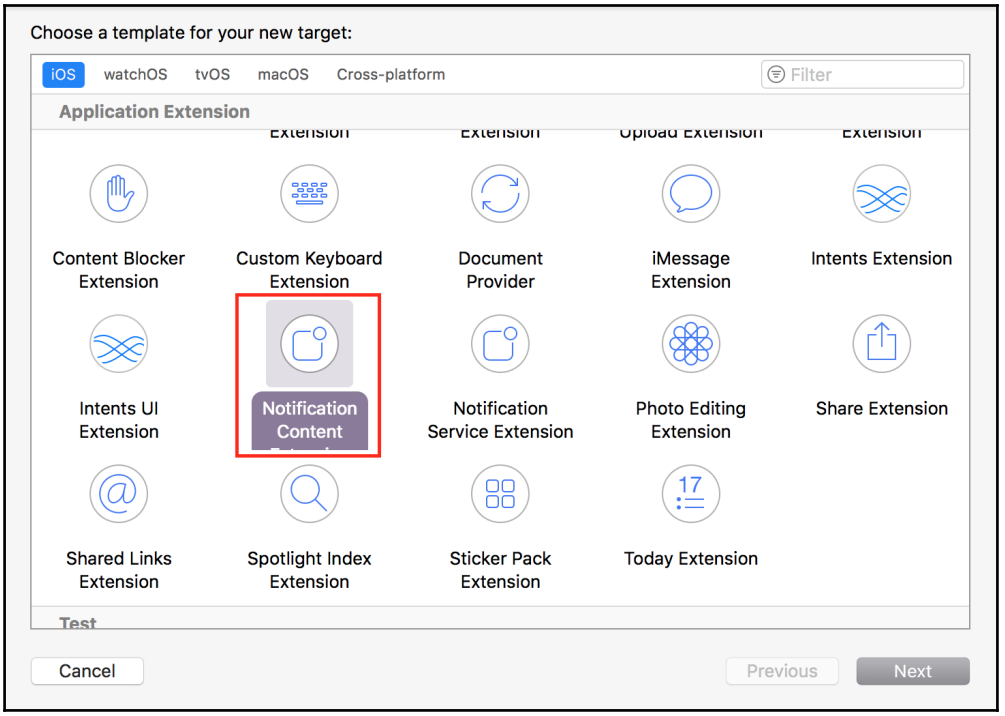

iPhone 7 – iOS 10.0 (14A345)
LETSEAT
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:30 PM

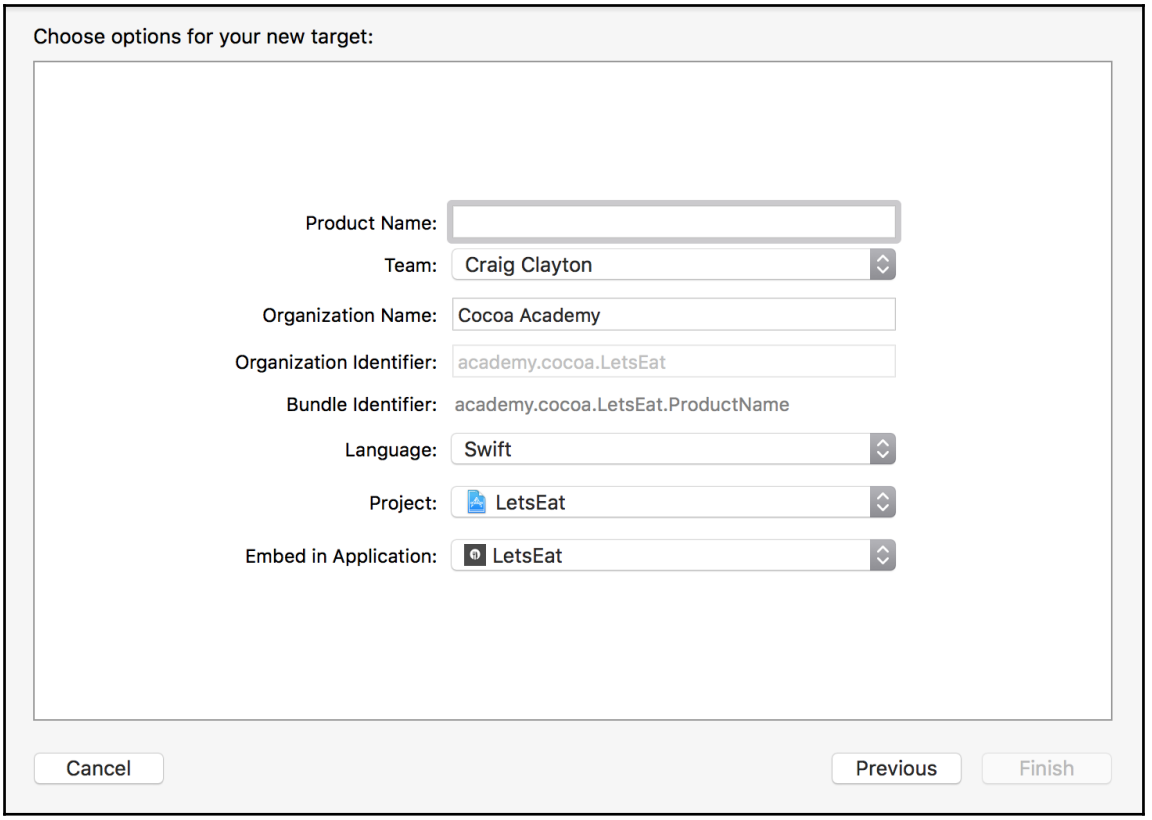

iPhone 7 – iOS 10.0 (14A345)
LETSEAT
now
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:30 PM
Safari
Messages

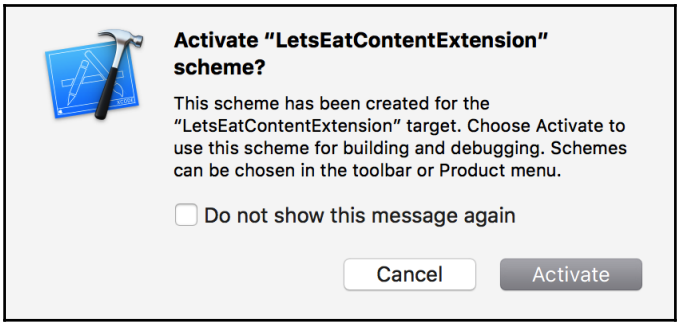


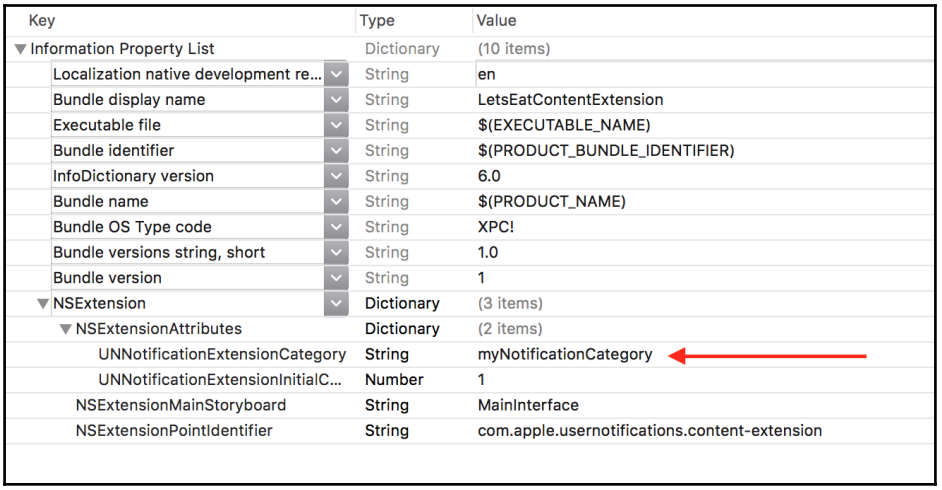

```
func checkNotifications() {
    UNUserNotificationCenter.current().requestAuthorization(options: [.alert, .sound, .badge]) { (isGranted, error) in

        // Add code here

    }
}
```


iPhone 7 – iOS 10.0 (14A345)
LETSEAT
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:30 PM
Yes
No

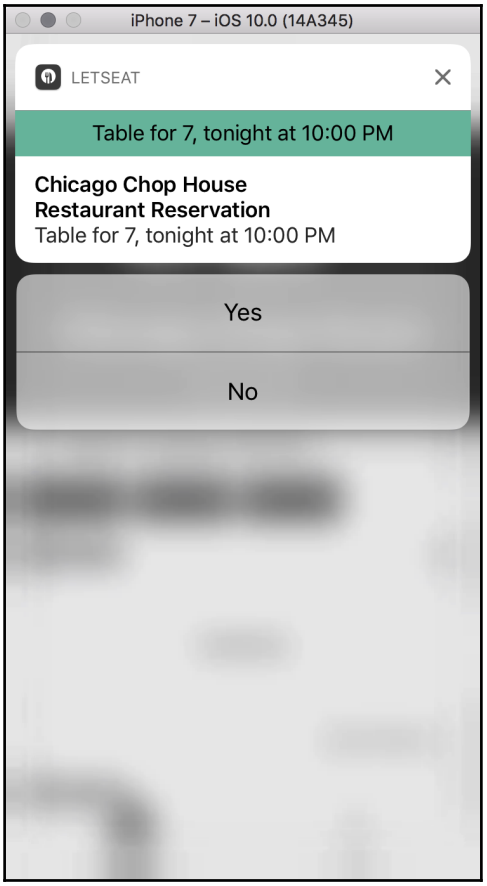

Choose a template for your new target:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application Extension
Content Blocker Extension
Custom Keyboard Extension
Document Provider
iMessage Extension
Intents Extension
Intents UI Extension
Notification Content
Notification Service Extension
Photo Editing Extension
Share Extension
Shared Links Extension
Spotlight Index Extension
Sticker Pack Extension
Today Extension
Test
Cancel
Previous
Next

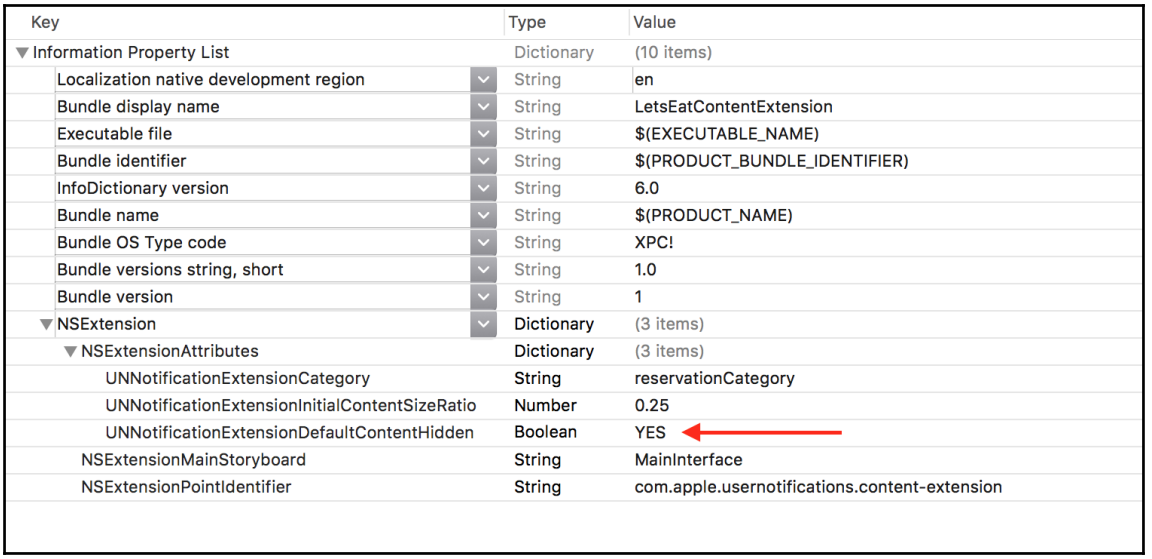

Choose options for your new target:
Product Name:
Team:
Craig Clayton
Organization Name:
Cocoa Academy
Organization Identifier:
academy.cocoa.LetsEat
Bundle Identifier:
academy.cocoa.LetsEat.ProductName
Language:
Swift
Project:
LetsEat
Embed in Application:
LetsEat
Cancel
Previous
Finish


Activate "LetsEatContentExtension" scheme?
This scheme has been created for the "LetsEatContentExtension" target. Choose Activate to use this scheme for building and debugging. Schemes can be chosen in the toolbar or Product menu.
Do not show this message again
Cancel
Activate

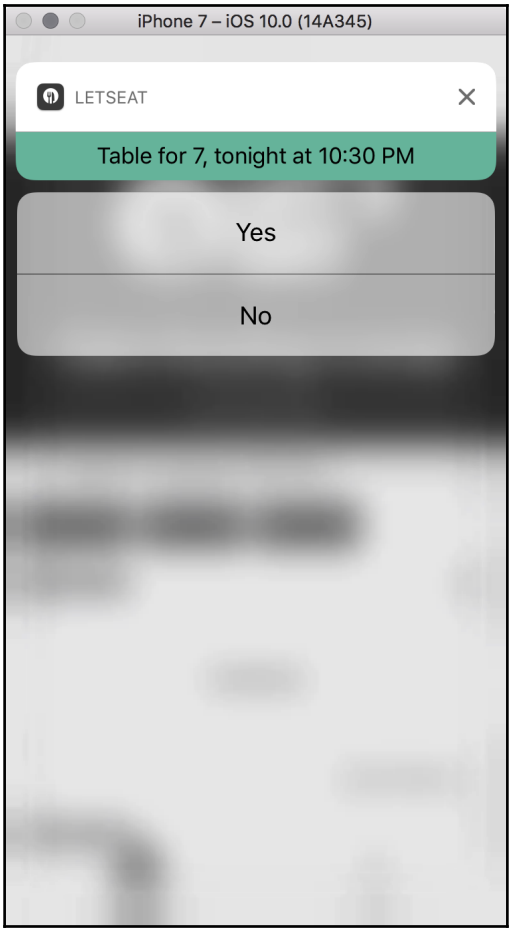


Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development re...	String	en
Bundle display name	String	LetsEatContentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
NSExtension	Dictionary	(3 items)
NSExtensionAttributes	Dictionary	(2 items)
UNNotificationExtensionCategory	String	myNotificationCategory
UNNotificationExtensionInitialC...	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension


iPhone 7 – iOS 10.0 (14A345)
LETSEAT
Table for 7, tonight at 10:00 PM
Chicago Chop House
Restaurant Reservation
Table for 7, tonight at 10:00 PM
Yes
No



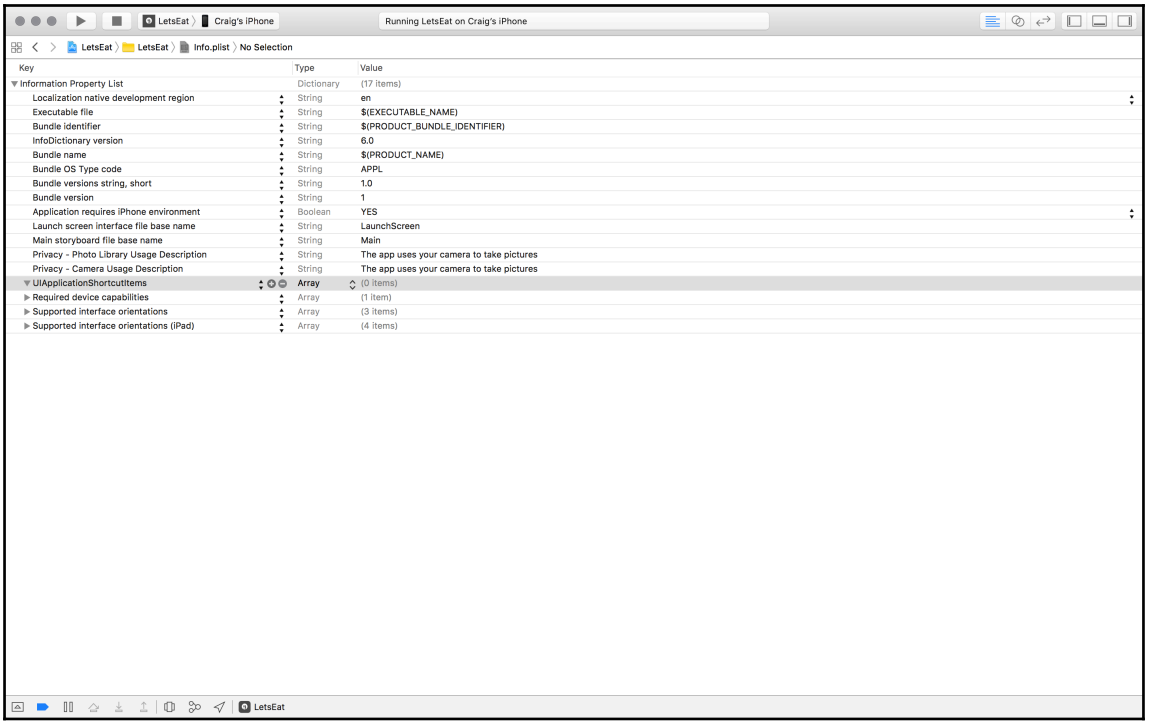
Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development region	String	en
Bundle display name	String	LetsEatContentExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
NSExtension	Dictionary	(3 items)
NSExtensionAttributes	Dictionary	(3 items)
UNNotificationExtensionCategory	String	reservationCategory
UNNotificationExtensionInitialContentSizeRatio	Number	0.25
UNNotificationExtensionDefaultContentHidden	Boolean	YES
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

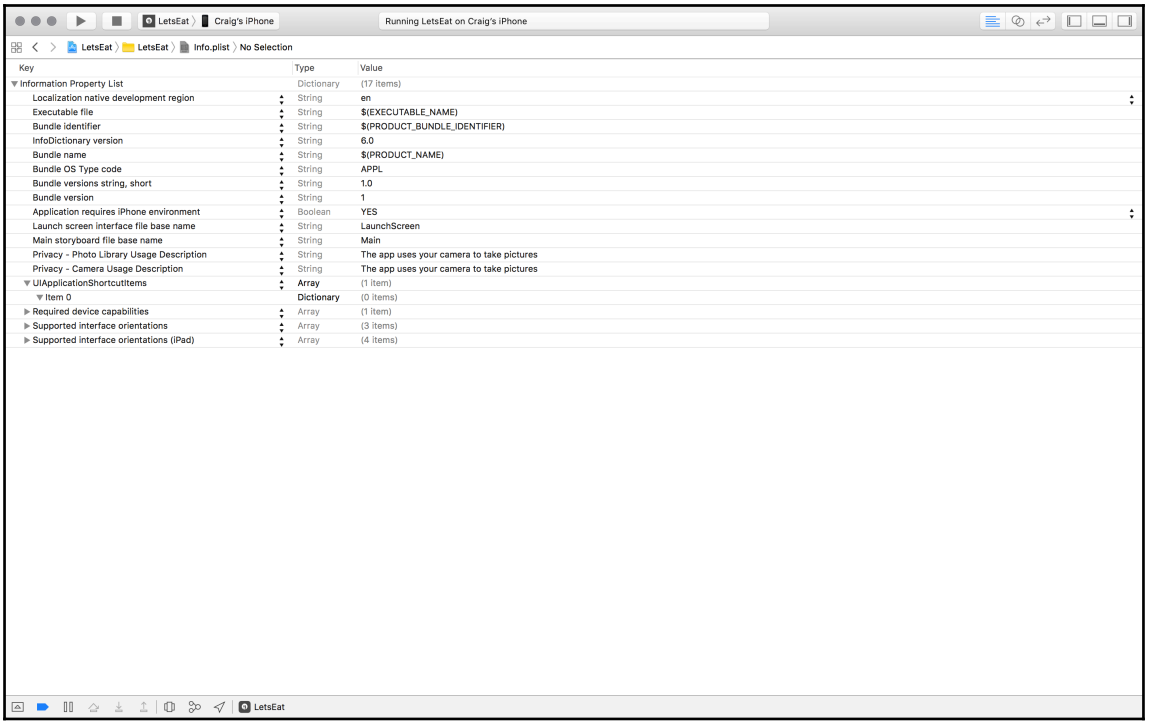

iPhone 7 – iOS 10.0 (14A345)
LETSEAT
Table for 7, tonight at 10:30 PM
Yes
No

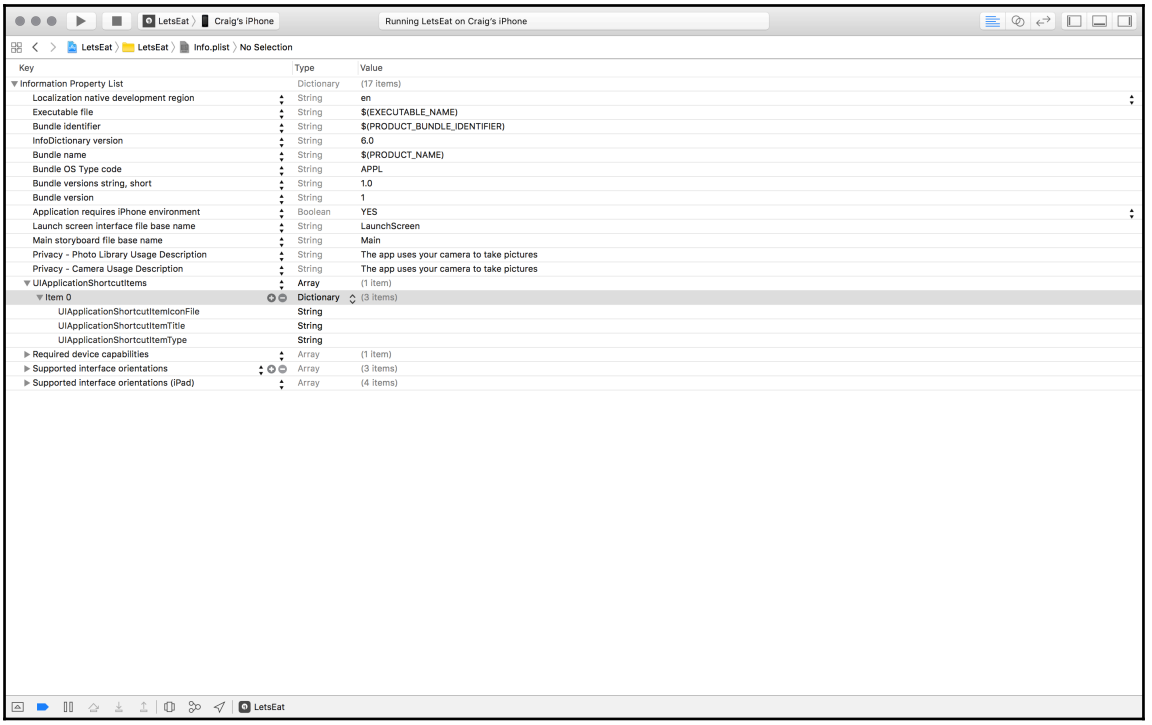
Chapter 26: Just a Peek

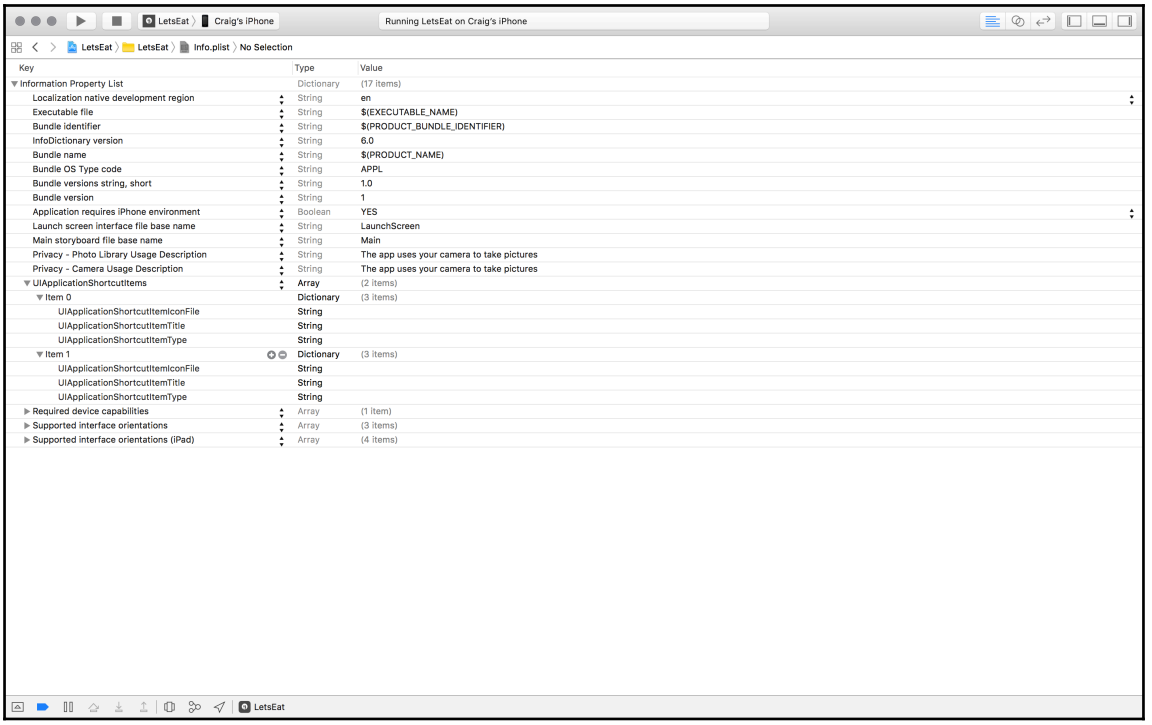
```
func checkShortCut(_ application: UIApplication, launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    var isPerformingAdditionalDelegateHandling = true
                                                                    A
    if let shortcutItem = launchOptions?[UIApplicationLaunchOptionsKey.shortcutItem] as? UIApplicationShortcutItem {

        launchedShortcutItem = shortcutItem
        isPerformingAdditionalDelegateHandling = false
    }

    if let shortcutItems = application.shortcutItems, shortcutItems.isEmpty {
        let laShortcut = UIMutableApplicationShortcutItem(type: Shortcut.openLosAngeles.type, localizedTitle: "Los
B           Angeles", localizedSubtitle: "", icon: UIApplicationShortcutIcon(templateImageName: "shortcut-city"),
            userInfo:nil
        )

C       let lvShortcut = UIMutableApplicationShortcutItem(type: Shortcut.openLasVegas.type, localizedTitle: "Las Vegas",
            localizedSubtitle: "", icon: UIApplicationShortcutIcon(templateImageName: "shortcut-city"), userInfo: nil
        )

        application.shortcutItems = [laShortcut, lvShortcut]
    }
                                                                    D
    return isPerformingAdditionalDelegateHandling
}
```

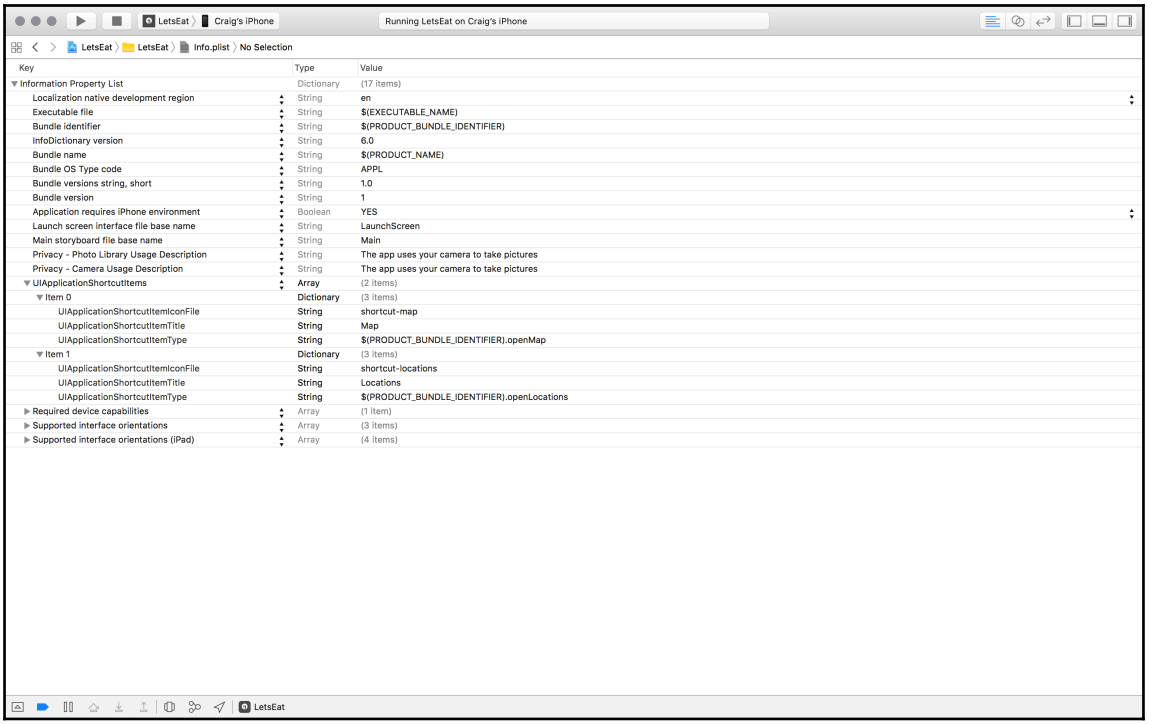
```
func handleShortCut(_ item: UIApplicationShortcutItem) -> Bool {
    var isHandled = false

A   guard Shortcut(with: item.type) != nil, let shortCutType = item.type as String?, let tabBarController = self.window?.
        rootViewController as? UITabBarController else { return false }

    switch (shortCutType) {

    case Shortcut.openLocations.type:

        tabBarController.selectedIndex = 0
B       let navController = self.window?.rootViewController?.childViewControllers.first as! UINavigationController
        let viewController = navController.childViewControllers.first as! ExploreViewController
        viewController.performSegue(withIdentifier: "locationList", sender: self)

        isHandled = true

        break

    case Shortcut.openMap.type:

C       tabBarController.selectedIndex = 1
        isHandled = true

        break

    case Shortcut.openLosAngeles.type:

        let navController = self.window?.rootViewController?.childViewControllers.first as! UINavigationController
        let viewController = navController.childViewControllers.first as! ExploreViewController
        viewController.selectedCity = LocationItem(state: "CA", city: "Los Angeles")

D       tabBarController.selectedIndex = 1
        tabBarController.selectedIndex = 0
        isHandled = true

        break

    case Shortcut.openLasVegas.type:

        let navController = self.window?.rootViewController?.childViewControllers.first as! UINavigationController
        let viewController = navController.childViewControllers.first as! ExploreViewController
E       viewController.selectedCity =  LocationItem(state: "NV", city: "Las Vegas")

        tabBarController.selectedIndex = 1
        tabBarController.selectedIndex = 0
        isHandled = true

        break

    default:
        break
    }

    return isHandled
}
```


LetsEat
Craig's iPhone
Running LetsEat on Craig's iPhone
LetsEat
LetsEat
Info.plist
No Selection
Key
Type
Value
Information Property List
Dictionary
(17 items)
Localization native development region
String
en
Executable file
String
$(EXECUTABLE_NAME)
Bundle identifier
String
$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version
String
6.0
Bundle name
String
$(PRODUCT_NAME)
Bundle OS Type code
String
APPL
Bundle versions string, short
String
1.0
Bundle version
String
1
Application requires iPhone environment
Boolean
YES
Launch screen interface file base name
String
LaunchScreen
Main storyboard file base name
String
Main
Privacy - Photo Library Usage Description
String
The app uses your camera to take pictures
Privacy - Camera Usage Description
String
The app uses your camera to take pictures
UIApplicationShortcutItems
Array
(0 items)
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
LetsEat

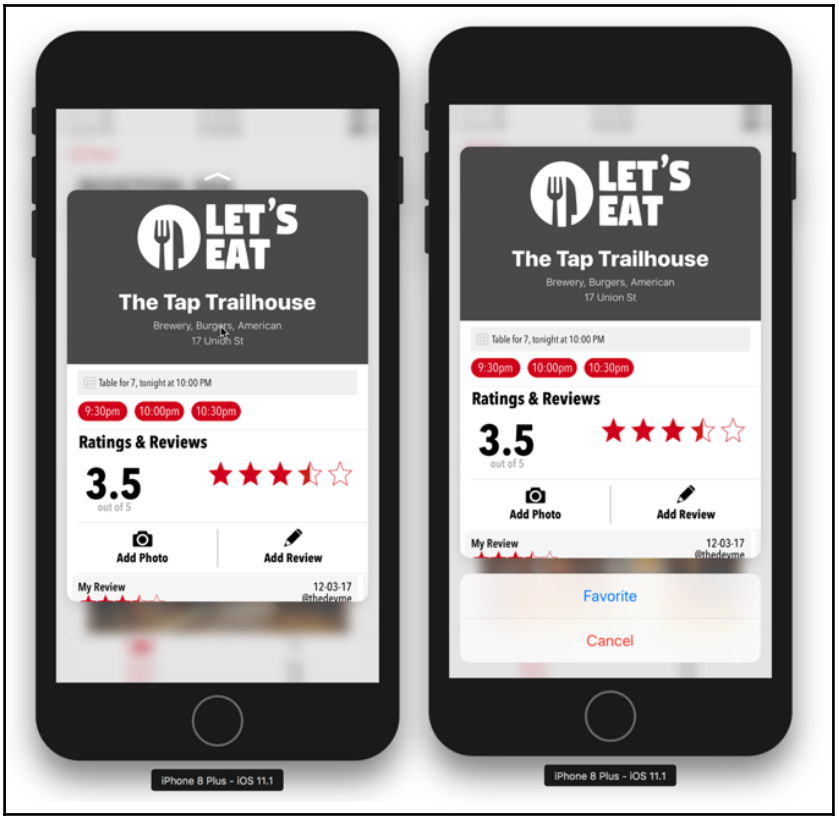

LetsEat
Craig's iPhone
Running LetsEat on Craig's iPhone
LetsEat
LetsEat
Info.plist
No Selection
Key
Type
Value
Information Property List
Dictionary
(17 items)
Localization native development region
String
en
Executable file
String
$(EXECUTABLE_NAME)
Bundle identifier
String
$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version
String
6.0
Bundle name
String
$(PRODUCT_NAME)
Bundle OS Type code
String
APPL
Bundle versions string, short
String
1.0
Bundle version
String
1
Application requires iPhone environment
Boolean
YES
Launch screen interface file base name
String
LaunchScreen
Main storyboard file base name
String
Main
Privacy - Photo Library Usage Description
String
The app uses your camera to take pictures
Privacy - Camera Usage Description
String
The app uses your camera to take pictures
UIApplicationShortcutItems
Array
(1 item)
Item 0
Dictionary
(0 items)
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
LetsEat

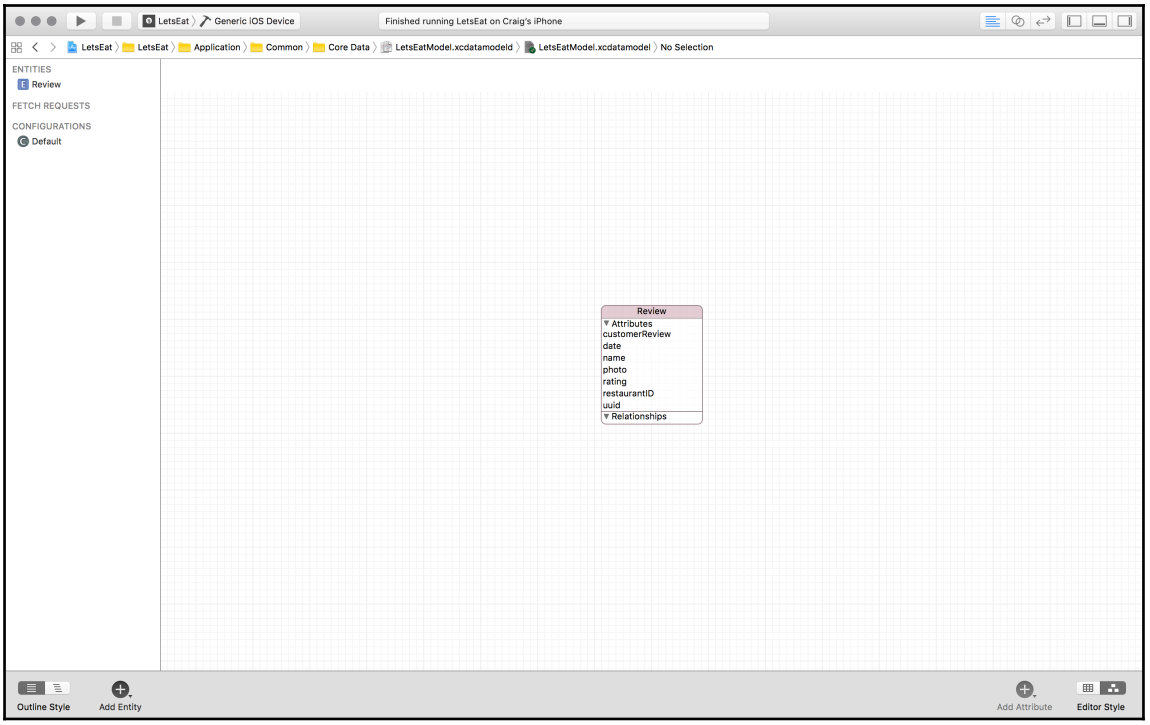

LetsEat
Craig's iPhone
Running LetsEat on Craig's iPhone
LetsEat
LetsEat
Info.plist
No Selection
Key
Type
Value
Information Property List
Dictionary
(17 items)
Localization native development region
String
en
Executable file
String
$(EXECUTABLE_NAME)
Bundle identifier
String
$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version
String
6.0
Bundle name
String
$(PRODUCT_NAME)
Bundle OS Type code
String
APPL
Bundle versions string, short
String
1.0
Bundle version
String
1
Application requires iPhone environment
Boolean
YES
Launch screen interface file base name
String
LaunchScreen
Main storyboard file base name
String
Main
Privacy - Photo Library Usage Description
String
The app uses your camera to take pictures
Privacy - Camera Usage Description
String
The app uses your camera to take pictures
UIApplicationShortcutItems
Array
(1 item)
Item 0
Dictionary
(3 items)
UIApplicationShortcutItemIconFile
String
UIApplicationShortcutItemTitle
String
UIApplicationShortcutItemType
String
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
LetsEat

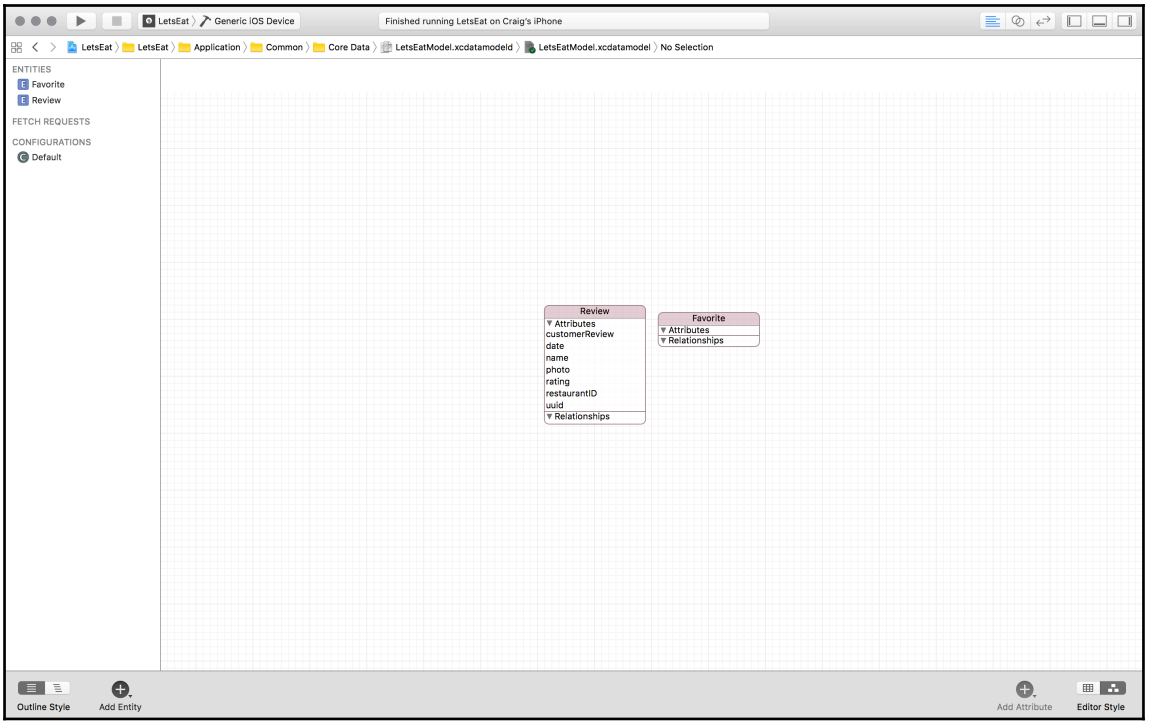

LetsEat 〉 Craig's iPhone
Running LetsEat on Craig's iPhone
LetsEat 〉 LetsEat 〉 Info.plist 〉 No Selection
Key
Type
Value
Information Property List
Dictionary
(17 items)
Localization native development region
String
en
Executable file
String
$(EXECUTABLE_NAME)
Bundle identifier
String
$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version
String
6.0
Bundle name
String
$(PRODUCT_NAME)
Bundle OS Type code
String
APPL
Bundle versions string, short
String
1.0
Bundle version
String
1
Application requires iPhone environment
Boolean
YES
Launch screen interface file base name
String
LaunchScreen
Main storyboard file base name
String
Main
Privacy - Photo Library Usage Description
String
The app uses your camera to take pictures
Privacy - Camera Usage Description
String
The app uses your camera to take pictures
UIApplicationShortcutItems
Array
(2 items)
Item 0
Dictionary
(3 items)
UIApplicationShortcutItemIconFile
String
UIApplicationShortcutItemTitle
String
UIApplicationShortcutItemType
String
Item 1
Dictionary
(3 items)
UIApplicationShortcutItemIconFile
String
UIApplicationShortcutItemTitle
String
UIApplicationShortcutItemType
String
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
LetsEat

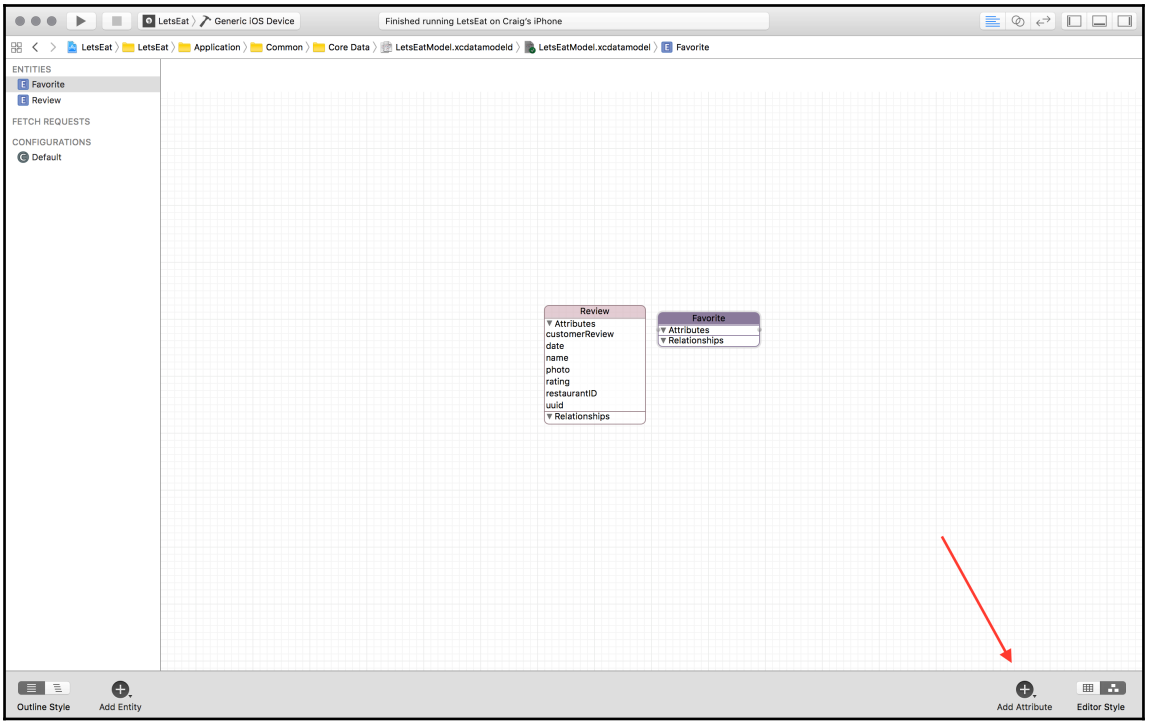

LetsEat 〉 Craig's iPhone
Running LetsEat on Craig's iPhone
LetsEat 〉 LetsEat 〉 Info.plist 〉 No Selection
Key
Type
Value
Information Property List
Dictionary
(17 items)
Localization native development region
String
en
Executable file
String
$(EXECUTABLE_NAME)
Bundle identifier
String
$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version
String
6.0
Bundle name
String
$(PRODUCT_NAME)
Bundle OS Type code
String
APPL
Bundle versions string, short
String
1.0
Bundle version
String
1
Application requires iPhone environment
Boolean
YES
Launch screen interface file base name
String
LaunchScreen
Main storyboard file base name
String
Main
Privacy - Photo Library Usage Description
String
The app uses your camera to take pictures
Privacy - Camera Usage Description
String
The app uses your camera to take pictures
UIApplicationShortcutItems
Array
(2 items)
Item 0
Dictionary
(3 items)
UIApplicationShortcutItemIconFile
String
shortcut-map
UIApplicationShortcutItemTitle
String
Map
UIApplicationShortcutItemType
String
$(PRODUCT_BUNDLE_IDENTIFIER).openMap
Item 1
Dictionary
(3 items)
UIApplicationShortcutItemIconFile
String
shortcut-locations
UIApplicationShortcutItemTitle
String
Locations
UIApplicationShortcutItemType
String
$(PRODUCT_BUNDLE_IDENTIFIER).openLocations
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(3 items)
Supported interface orientations (iPad)
Array
(4 items)
LetsEat

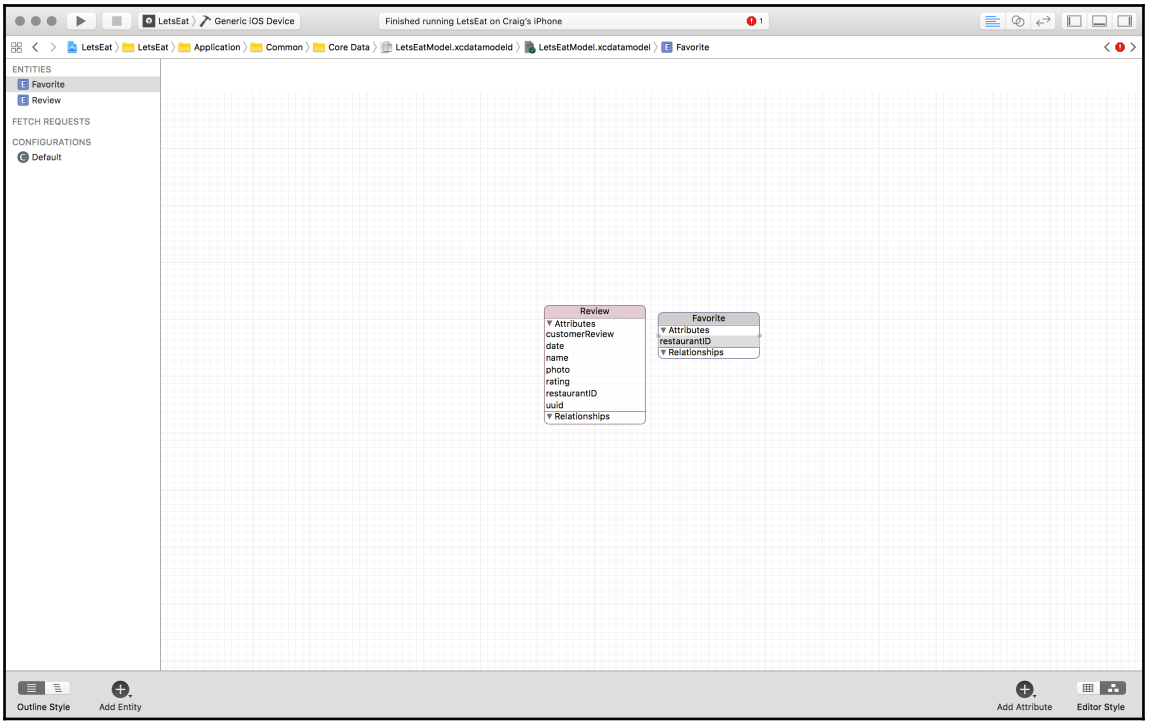

1
Map
Locations
Los Angeles
Las Vegas
iPhone 8 Plus - iOS 11.1

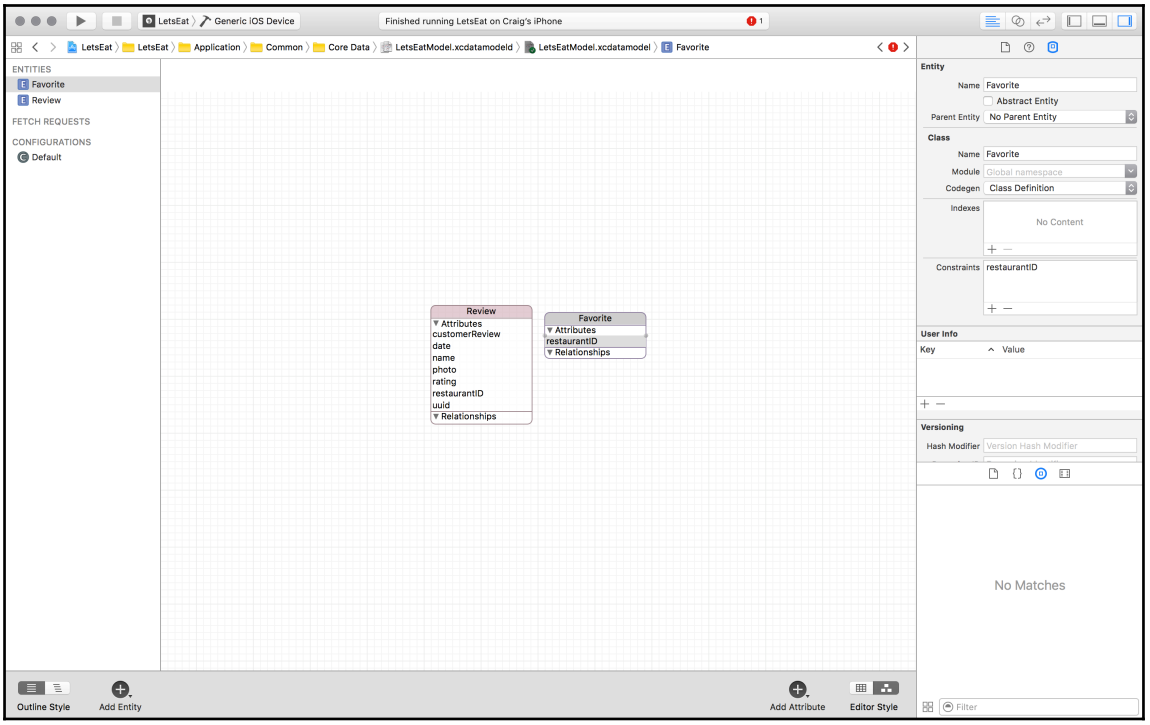

LET'S EAT
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
9:30pm
10:00pm
10:30pm
Ratings & Reviews
3.5
out of 5
Add Photo
Add Review
My Review
12-03-17
iPhone 8 Plus - iOS 11.1
LET'S EAT
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
9:30pm
10:00pm
10:30pm
Ratings & Reviews
3.5
out of 5
Add Photo
Add Review
My Review
12-03-17
Favorite
Cancel
iPhone 8 Plus - iOS 11.1

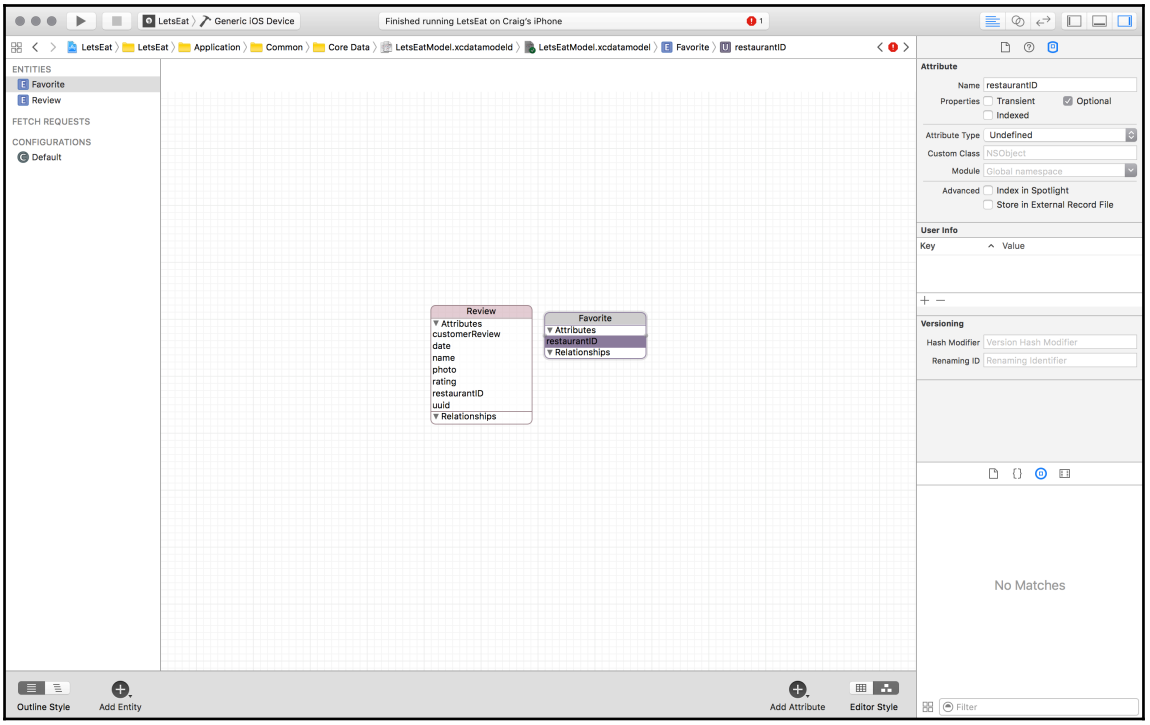

LetsEat 〉 Generic iOS Device
Finished running LetsEat on Craig's iPhone
LetsEat 〉 LetsEat 〉 Application 〉 Common 〉 Core Data 〉 LetsEatModel.xcdatamodeld 〉 LetsEatModel.xcdatamodel 〉 No Selection
ENTITIES
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
customerReview
date
name
photo
rating
restaurantID
uuid
Relationships
Outline Style
Add Entity
Add Attribute
Editor Style


LetsEat 〉 Generic iOS Device
Finished running LetsEat on Craig's iPhone
LetsEat 〉 LetsEat 〉 Application 〉 Common 〉 Core Data 〉 LetsEatModel.xcdatamodeld 〉 LetsEatModel.xcdatamodel 〉 No Selection
ENTITIES
Favorite
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
customerReview
date
name
photo
rating
restaurantID
uuid
Relationships
Favorite
Attributes
Relationships
Outline Style
Add Entity
Add Attribute
Editor Style

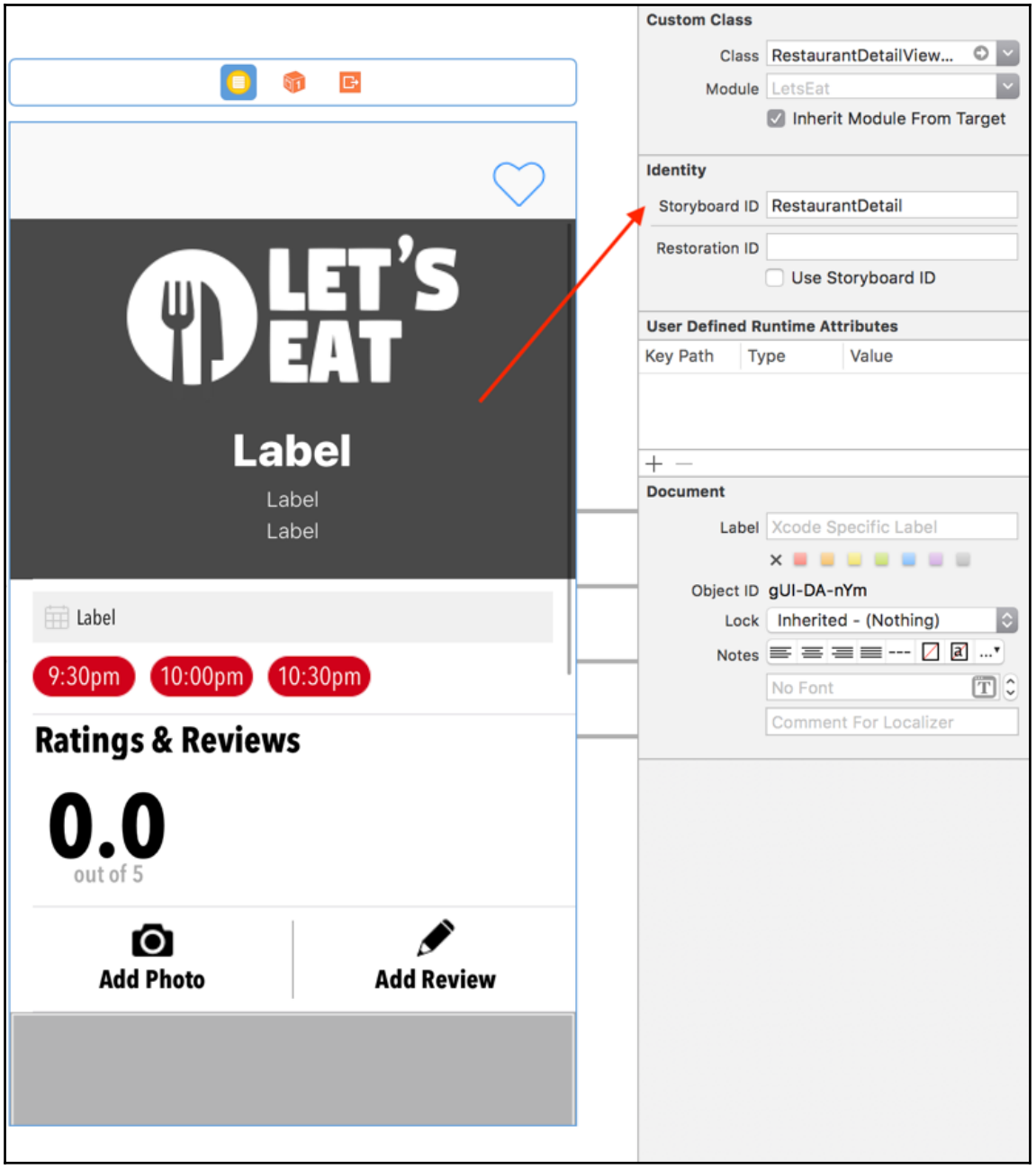

LetsEat 〉 Generic iOS Device
Finished running LetsEat on Craig's iPhone
LetsEat 〉 LetsEat 〉 Application 〉 Common 〉 Core Data 〉 LetsEatModel.xcdatamodeld 〉 LetsEatModel.xcdatamodel 〉 Favorite
ENTITIES
Favorite
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
customerReview
date
name
photo
rating
restaurantID
uuid
Relationships
Favorite
Attributes
Relationships
Outline Style
Add Entity
Add Attribute
Editor Style

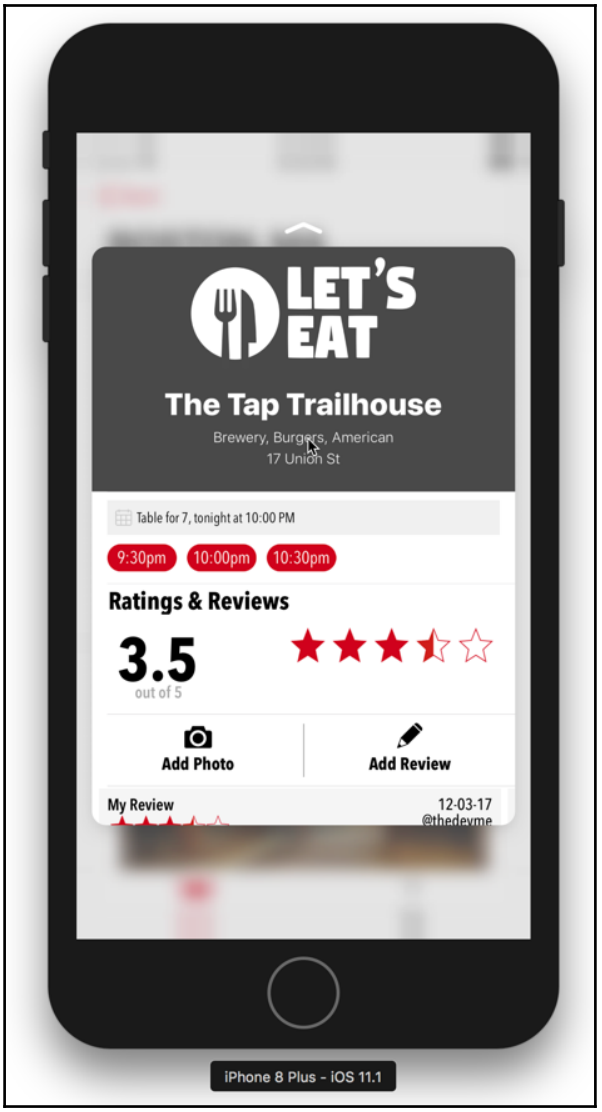

LetsEat 〉 Generic iOS Device
Finished running LetsEat on Craig's iPhone
LetsEat 〉 LetsEat 〉 Application 〉 Common 〉 Core Data 〉 LetsEatModel.xcdatamodeld 〉 LetsEatModel.xcdatamodel 〉 Favorite
ENTITIES
Favorite
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
customerReview
date
name
photo
rating
restaurantID
uuid
Relationships
Favorite
Attributes
restaurantID
Relationships
Outline Style
Add Entity
Add Attribute
Editor Style

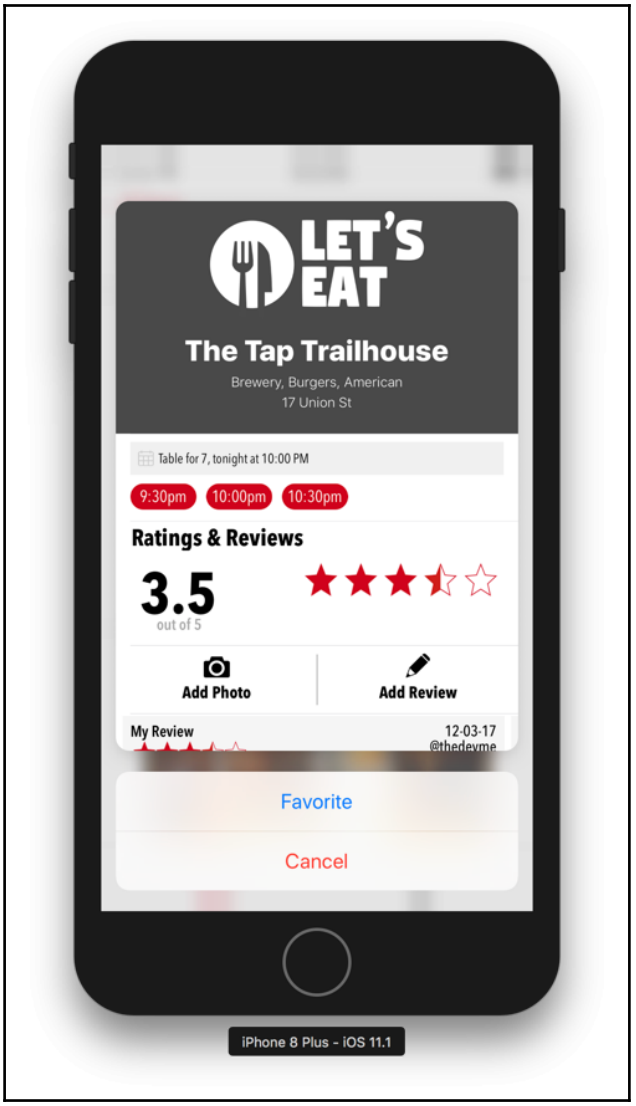

LetsEat
Generic iOS Device
Finished running LetsEat on Craig's iPhone
LetsEat
LetsEat
Application
Common
Core Data
LetsEatModel.xcdatamodeld
LetsEatModel.xcdatamodel
Favorite
ENTITIES
Favorite
Review
FETCH REQUESTS
CONFIGURATIONS
Default
Review
Attributes
customerReview
date
name
photo
rating
restaurantID
uuid
Relationships
Favorite
Attributes
restaurantID
Relationships
Entity
Name
Favorite
Abstract Entity
Parent Entity
No Parent Entity
Class
Name
Favorite
Module
Global namespace
Codegen
Class Definition
Indexes
No Content
Constraints
restaurantID
User Info
Key
Value
Versioning
Hash Modifier
Version Hash Modifier
No Matches
Filter
Outline Style
Add Entity
Add Attribute
Editor Style



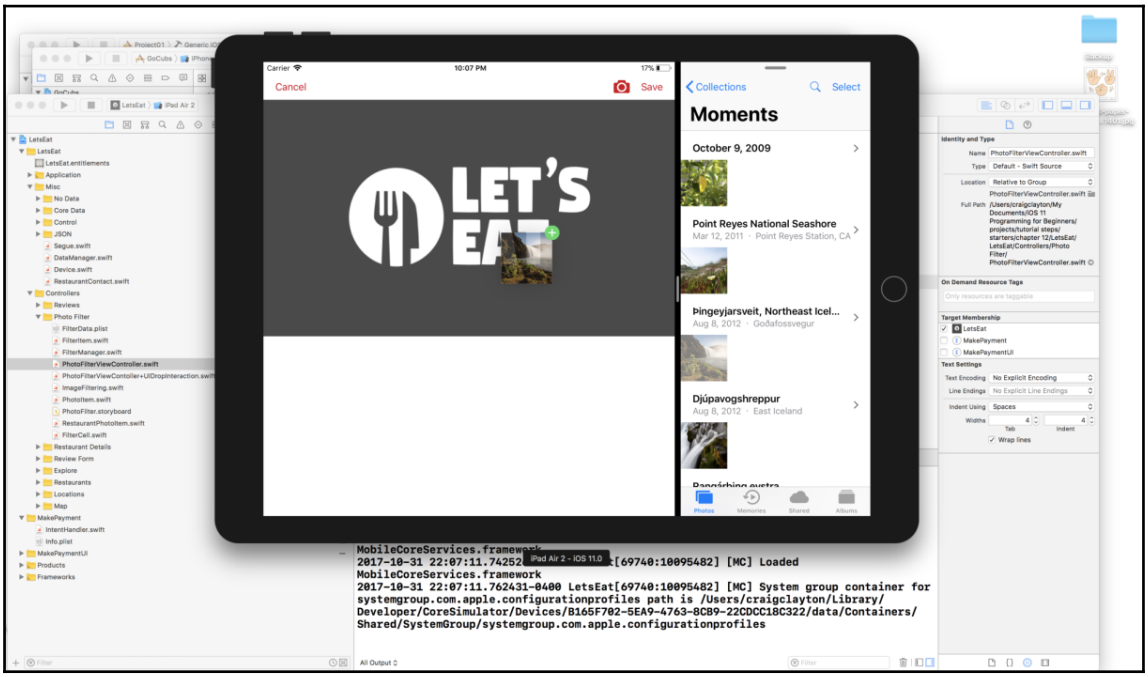

```
extension RestaurantViewController: UIViewControllerPreviewingDelegate {

    func previewingContext(_ previewingContext: UIViewControllerPreviewing, viewControllerForLocation location: CGPoint) ->
        UIViewController? {
        let restaurantDetail : UIStoryboard = UIStoryboard(name: "RestaurantDetail", bundle: nil)
A
        guard let indexPath = collectionView?.indexPathForItem(at: location), let cell = collectionView?.cellForItem(at:
            indexPath), let detailVC = restaurantDetail.instantiateViewController(withIdentifier: "RestaurantDetail") as?
            RestaurantDetailViewController else { return nil }

        selectedRestaurant = manager.restaurantItem(at: indexPath)
        detailVC.selectedRestaurant = selectedRestaurant
B       detailVC.preferredContentSize = CGSize(width: 0.0, height: 528)
        previewingContext.sourceRect = cell.frame

        return detailVC
    }

C   func previewingContext(_ previewingContext: UIViewControllerPreviewing, commit viewControllerToCommit: UIViewController)
        {

        show(viewControllerToCommit, sender: self)
    }
}
```

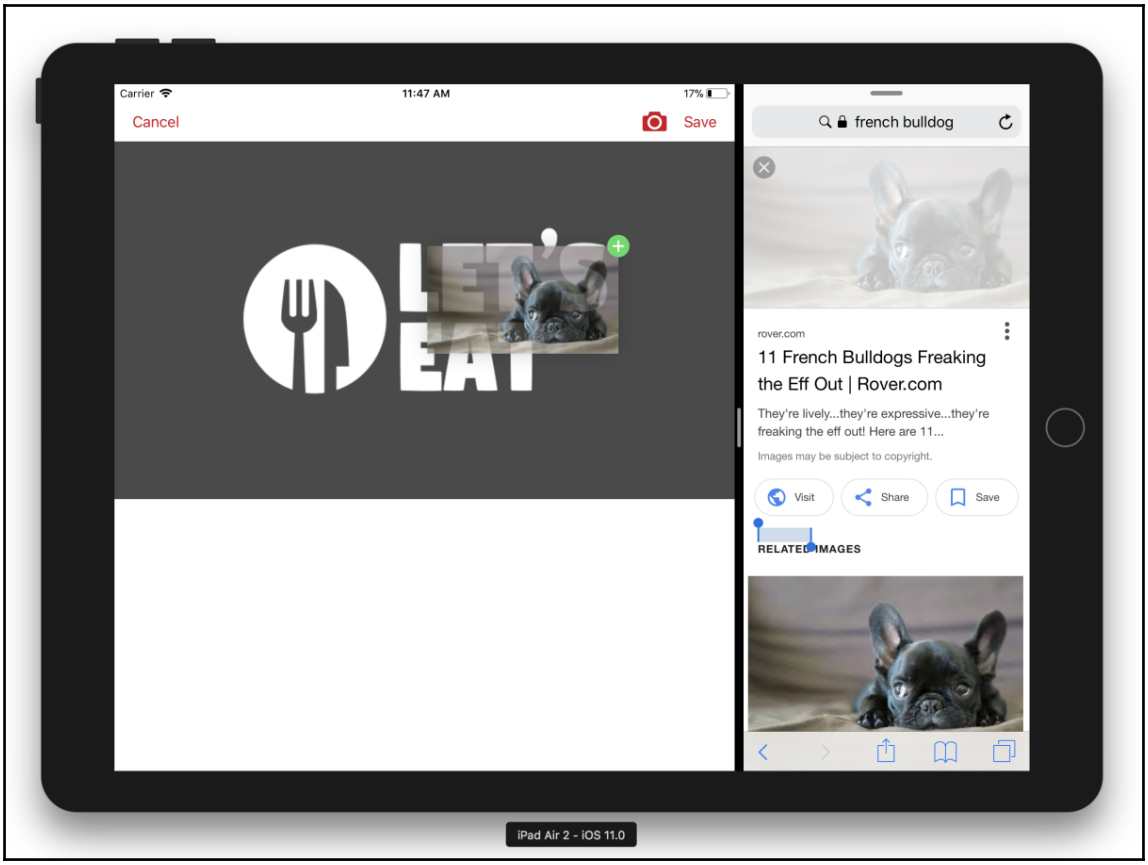

Custom Class
Class RestaurantDetailView...
Module LetsEat
Inherit Module From Target
Identity
Storyboard ID RestaurantDetail
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label Xcode Specific Label
Object ID gUI-DA-nYm
Lock Inherited - (Nothing)
Notes
No Font
Comment For Localizer
LET'S EAT
Label
Label
Label
Label
9:30pm
10:00pm
10:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review

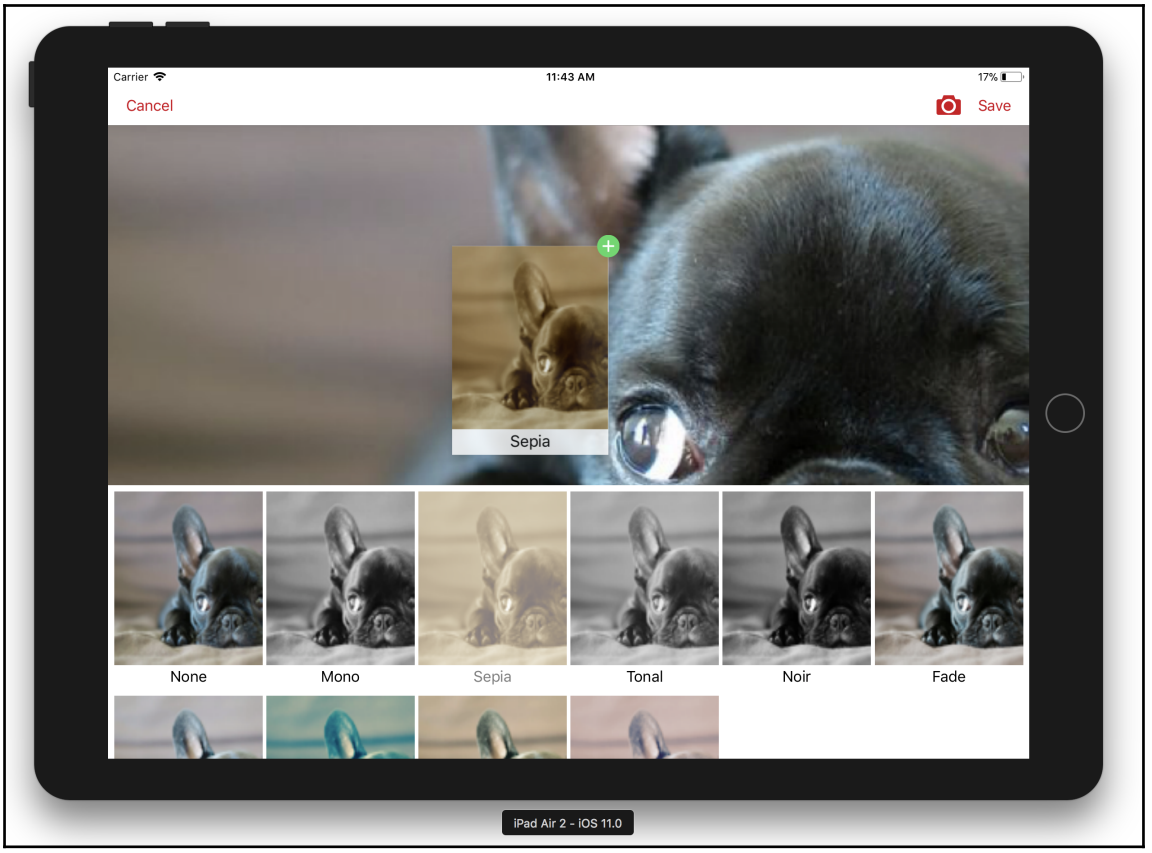

LET'S EAT
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
9:30pm
10:00pm
10:30pm
Ratings & Reviews
3.5
out of 5
Add Photo
Add Review
My Review
12-03-17
iPhone 8 Plus - iOS 11.1

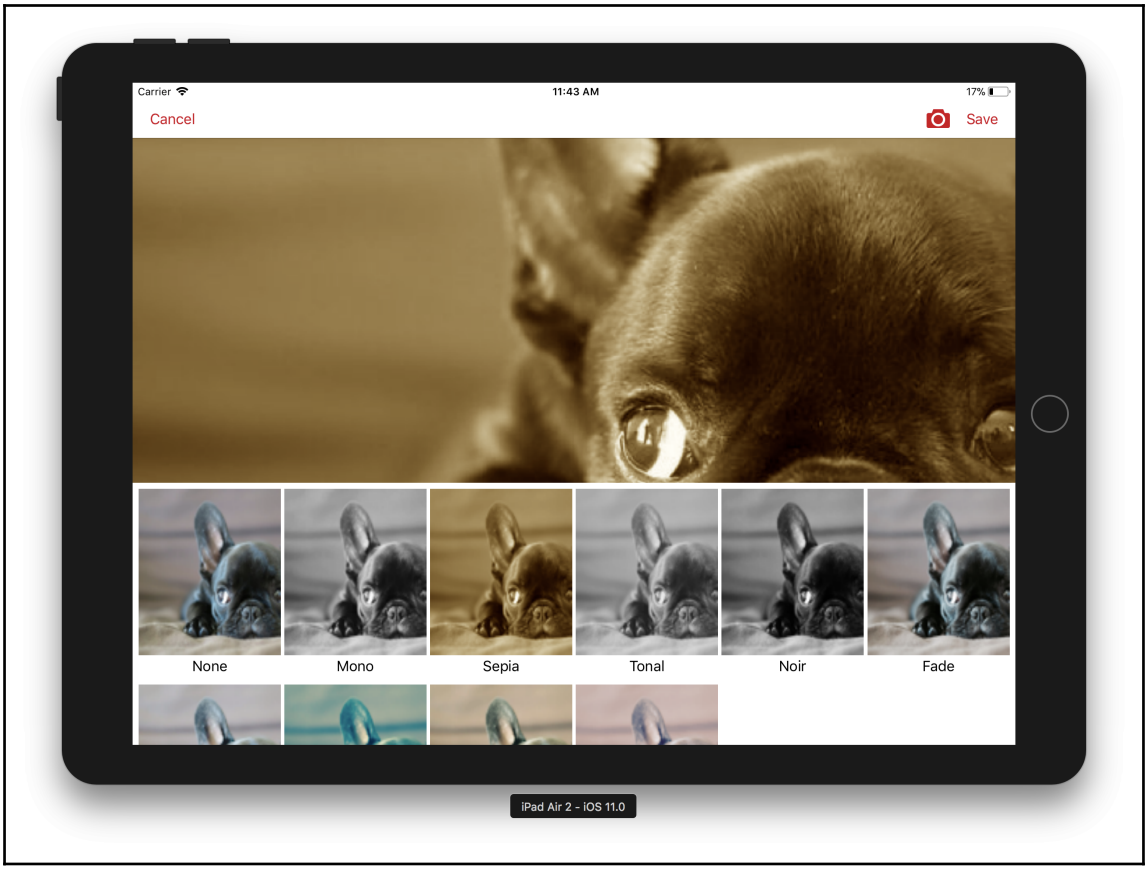

LET'S EAT
The Tap Trailhouse
Brewery, Burgers, American
17 Union St
Table for 7, tonight at 10:00 PM
9:30pm
10:00pm
10:30pm
Ratings & Reviews
3.5
out of 5
Add Photo
Add Review
My Review
12-03-17
Favorite
Cancel
iPhone 8 Plus - iOS 11.1

Chapter 27: Drag and Drop

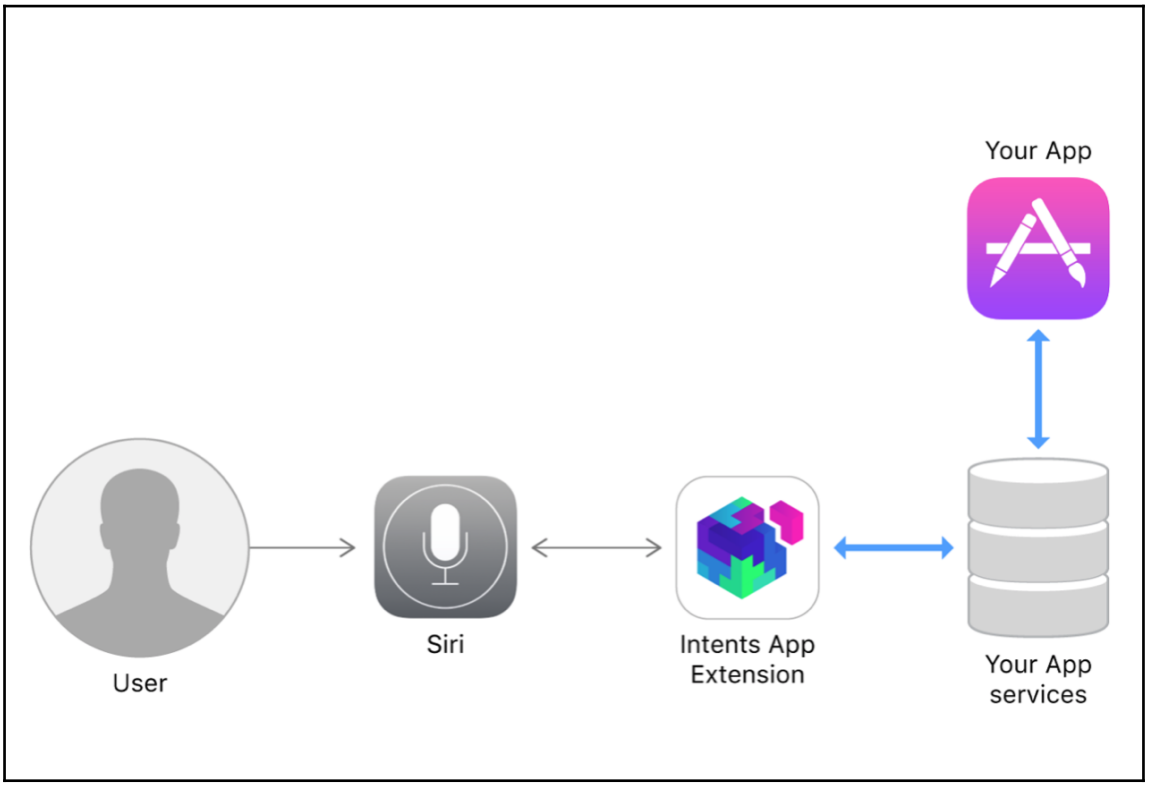


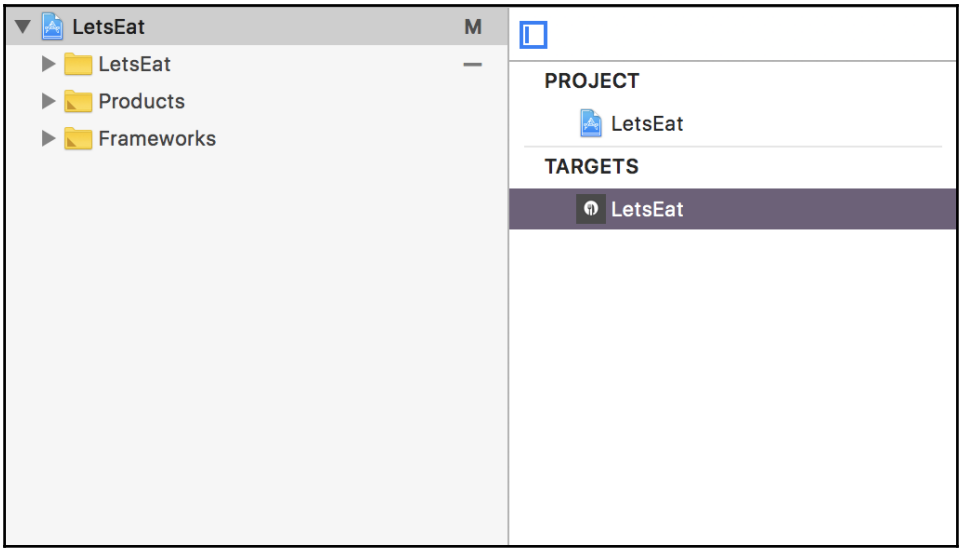

Carrier
11:47 AM
17%
Cancel
Save
LET'S EAT
french bulldog
rover.com
11 French Bulldogs Freaking the Eff Out | Rover.com
They're lively...they're expressive...they're freaking the eff out! Here are 11...
Images may be subject to copyright.
Visit
Share
Save
RELATED IMAGES
iPad Air 2 - iOS 11.0


Carrier
11:43 AM
17%
Cancel
Save
Sepia
None
Mono
Sepia
Tonal
Noir
Fade
iPad Air 2 - iOS 11.0

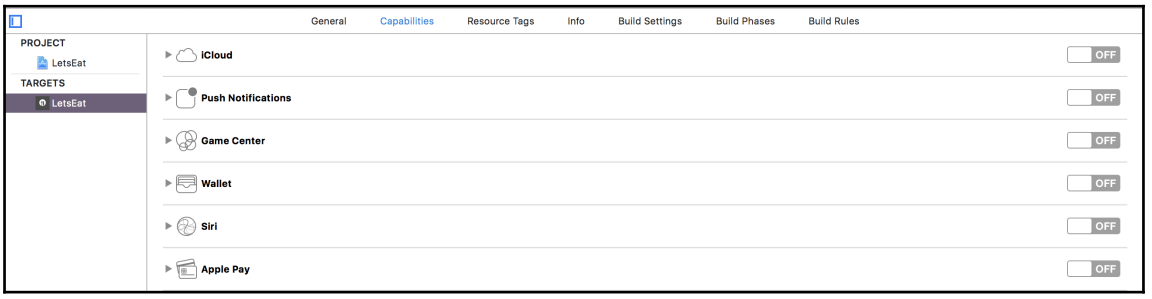

Carrier
11:43 AM
17%
Cancel
Save
None
Mono
Sepia
Tonal
Noir
Fade
iPad Air 2 - iOS 11.0

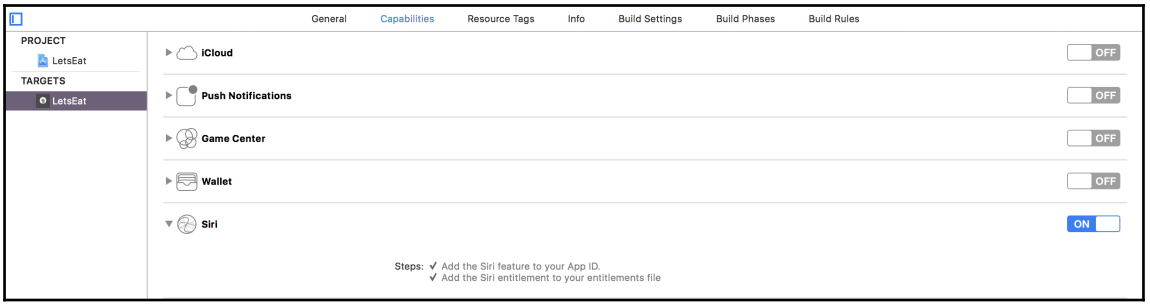
Chapter 28: SiriKit

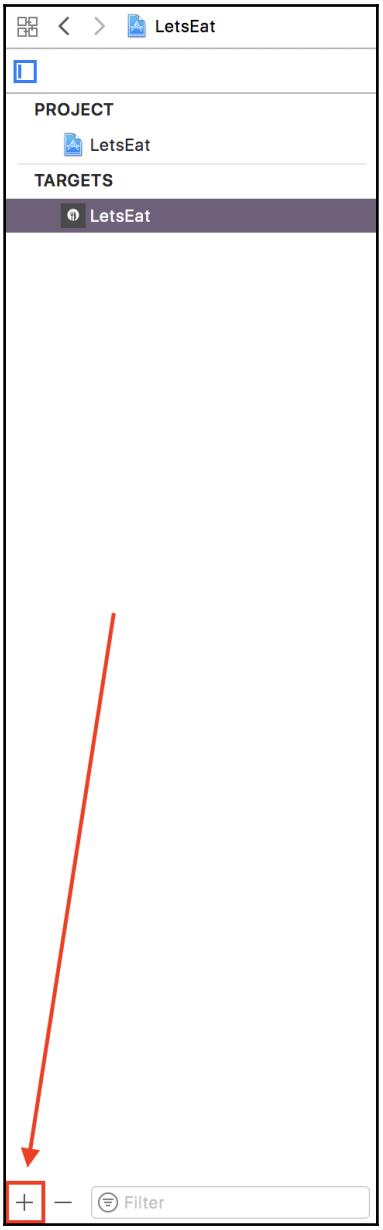



LetsEat
M
LetsEat
Products
Frameworks
PROJECT
LetsEat
TARGETS
LetsEat

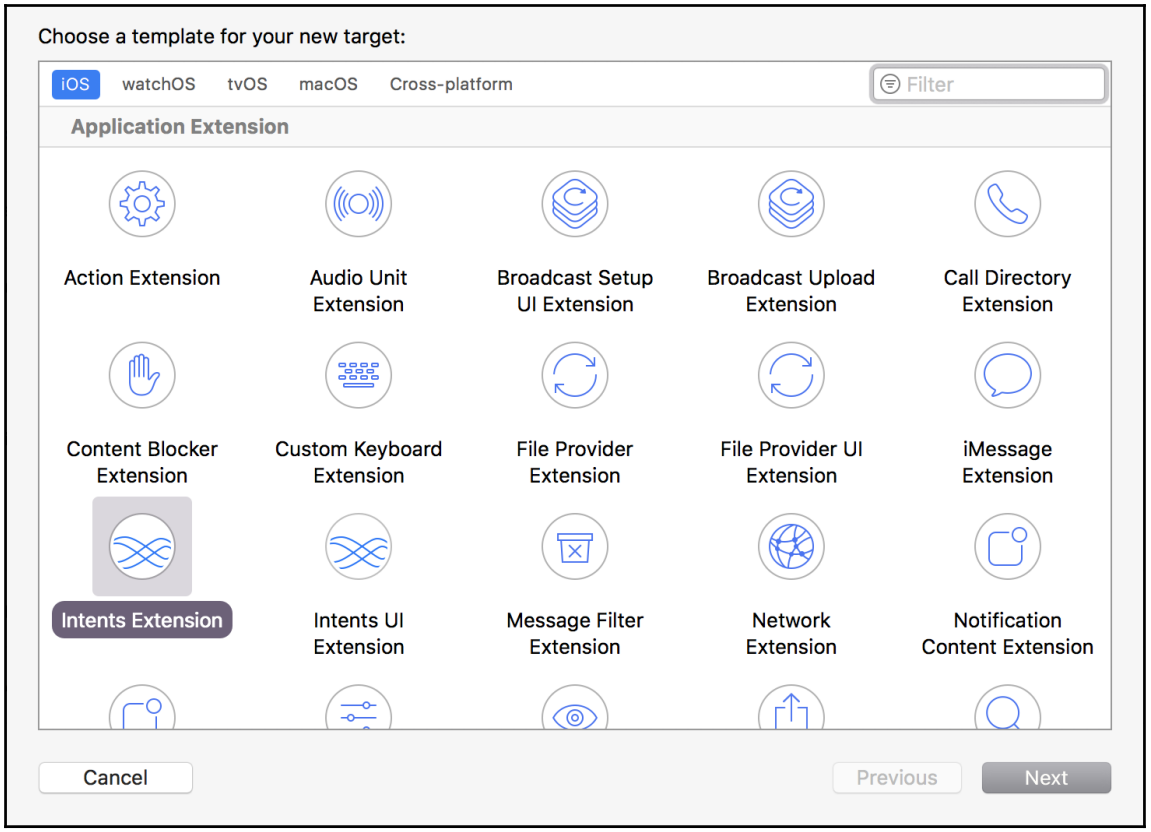

LetsEat
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
LetsEat
TARGETS
LetsEat
Identity
Display Name
LetsEat

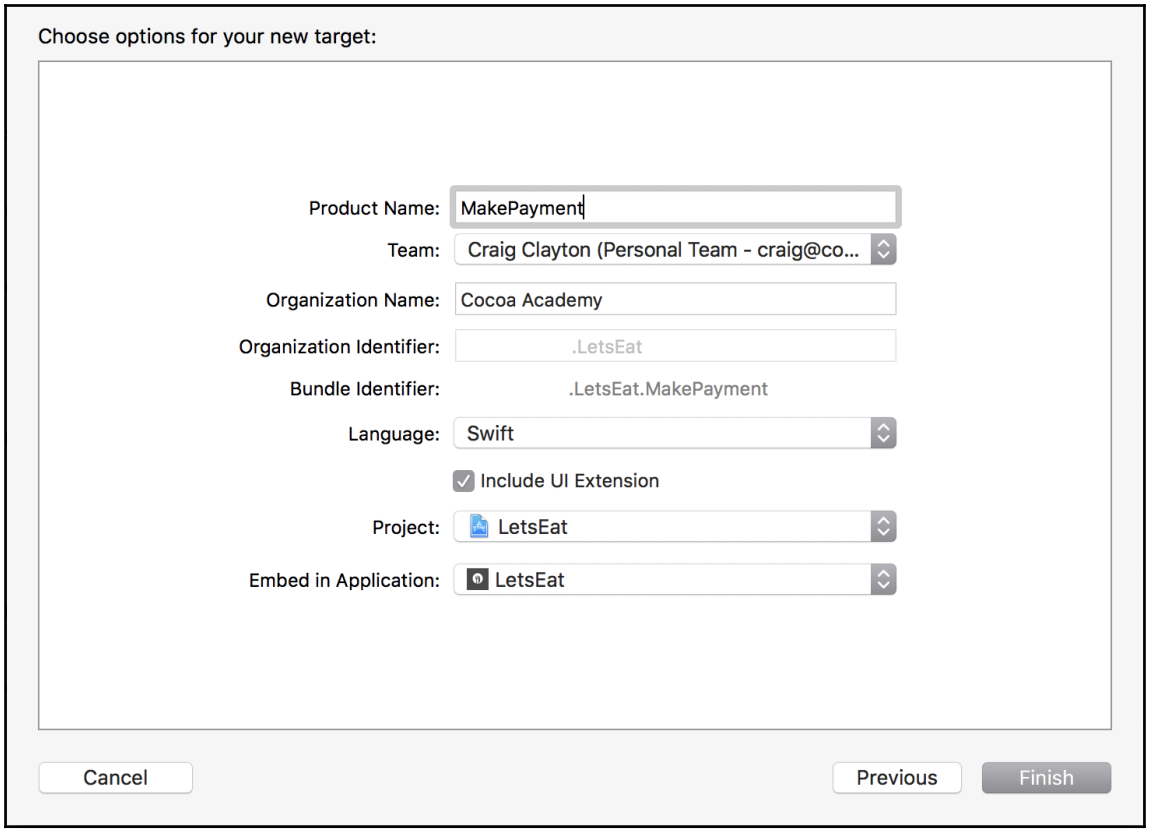

General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
LetsEat
TARGETS
LetsEat
iCloud
Push Notifications
Game Center
Wallet
Siri
Apple Pay
OFF


General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
LetsEat
TARGETS
LetsEat
iCloud
OFF
Push Notifications
OFF
Game Center
OFF
Wallet
OFF
Siri
ON
Steps: ✓ Add the Siri feature to your App ID.
✓ Add the Siri entitlement to your entitlements file


LetsEat
PROJECT
LetsEat
TARGETS
LetsEat
Filter

>


Choose a template for your new target:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application Extension
Action Extension
Audio Unit Extension
Broadcast Setup UI Extension
Broadcast Upload Extension
Call Directory Extension
Content Blocker Extension
Custom Keyboard Extension
File Provider Extension
File Provider UI Extension
iMessage Extension
Intents Extension
Intents UI Extension
Message Filter Extension
Network Extension
Notification Content Extension
Cancel
Previous
Next



Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(2 items)
▼ IntentsRestrictedWhileLocked	Array	(0 items)
▼ IntentsSupported	Array	(3 items)
Item 0	String	INSendMessageIntent
Item 1	String	INSearchForMessagesIntent
Item 2	String	INSetMessageAttributeIntent
NSExtensionPointIdentifier	String	com.apple.intents-service
NSExtensionPrincipalClass	String	$(PRODUCT_MODULE_NAME).IntentHandler

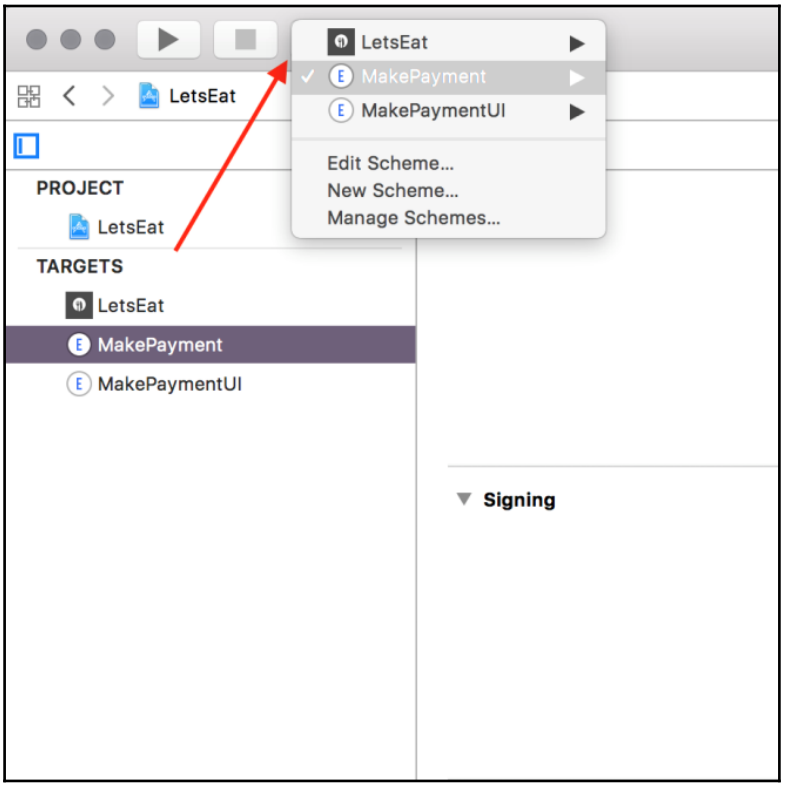
Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
NSExtension	Dictionary	(3 items)
NSExtensionAttributes	Dictionary	(2 items)
IntentsRestrictedWhileLocked	Array	(0 items)
IntentsSupported	Array	(3 items)
Item 0	String	INSendMessageIntent
Item 1	String	INSearchForMessagesIntent
Item 2	String	INSetMessageAttributeIntent
NSExtensionPointIdentifier	String	com.apple.intents-service
NSExtensionPrincipalClass	String	$(PRODUCT_MODULE_NAME).IntentHandler

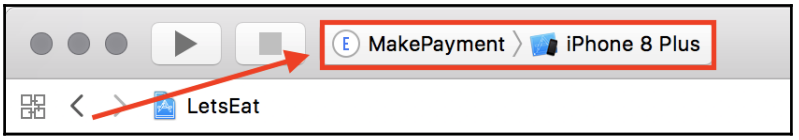
Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
NSExtension	Dictionary	(3 items)
NSExtensionAttributes	Dictionary	(2 items)
IntentsRestrictedWhileLo...	Array	(0 items)
IntentsSupported	Array	(1 item)
Item 0	String	INSendPaymentIntent
NSExtensionPointIdentifier	String	com.apple.intents-service
NSExtensionPrincipalClass	String	$(PRODUCT_MODULE_NAME).IntentHandler

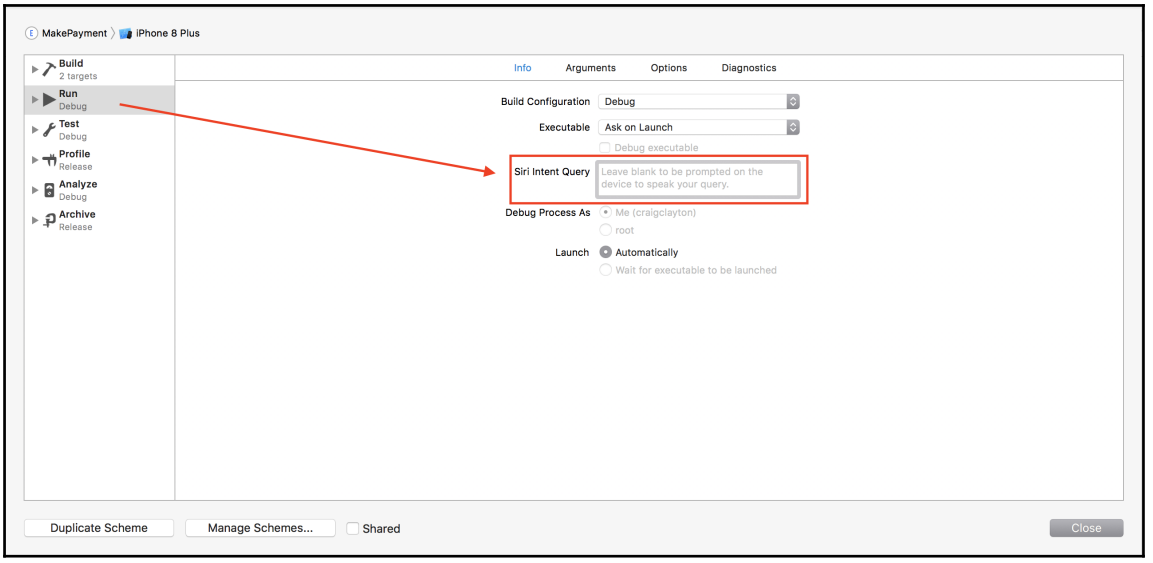
Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(2 items)
▼ IntentsRestrictedWhileLocked	Array	(1 item)
Item 0	String	INSendPaymentIntent
▼ IntentsSupported	Array	(1 item)
Item 0	String	INSendPaymentIntent
NSExtensionPointIdentifier	String	com.apple.intents-service
NSExtensionPrincipalClass	String	$(PRODUCT_MODULE_NAME).IntentHandler

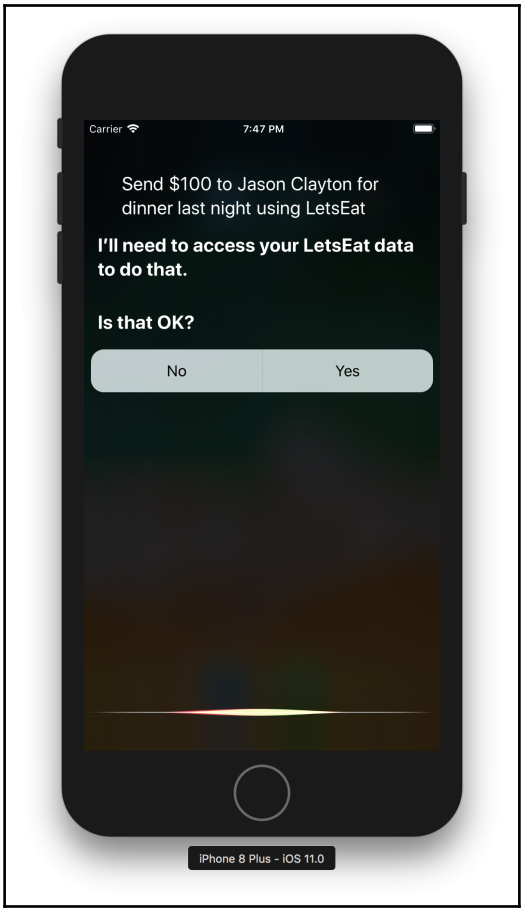
Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePaymentUI
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ IntentsSupported	Array	(1 item)
Item 0	String	INSendPaymentIntent
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.intents-ui-service

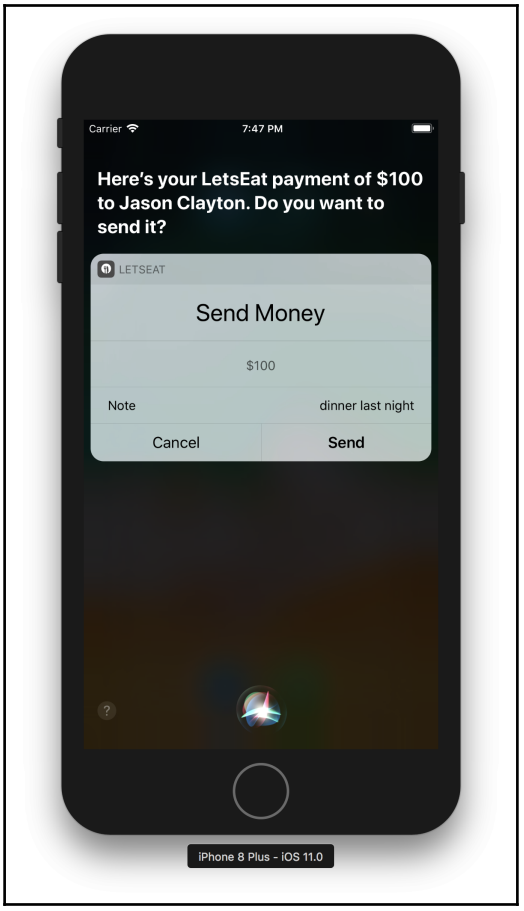
Key	Type	Value
▼ Information Property List	Dictionary	(18 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
Privacy - Camera Usage Description	String	The app uses your camera to take pictures
Privacy - Location When In Use U...	String	The app uses location
Privacy - Photo Library Usage Des...	String	The app uses your camera to take pictures
Privacy - Siri Usage Description	String	This app uses Siri to send payments.
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (i...	Array	(4 items)

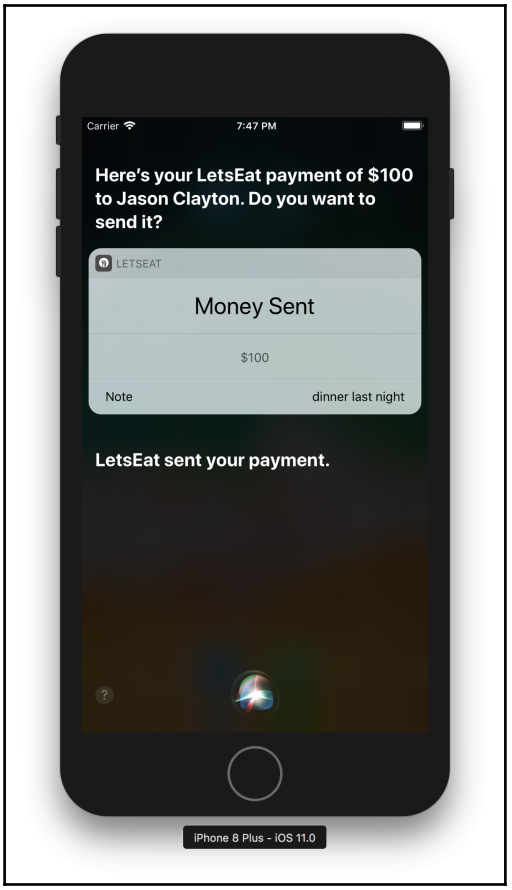


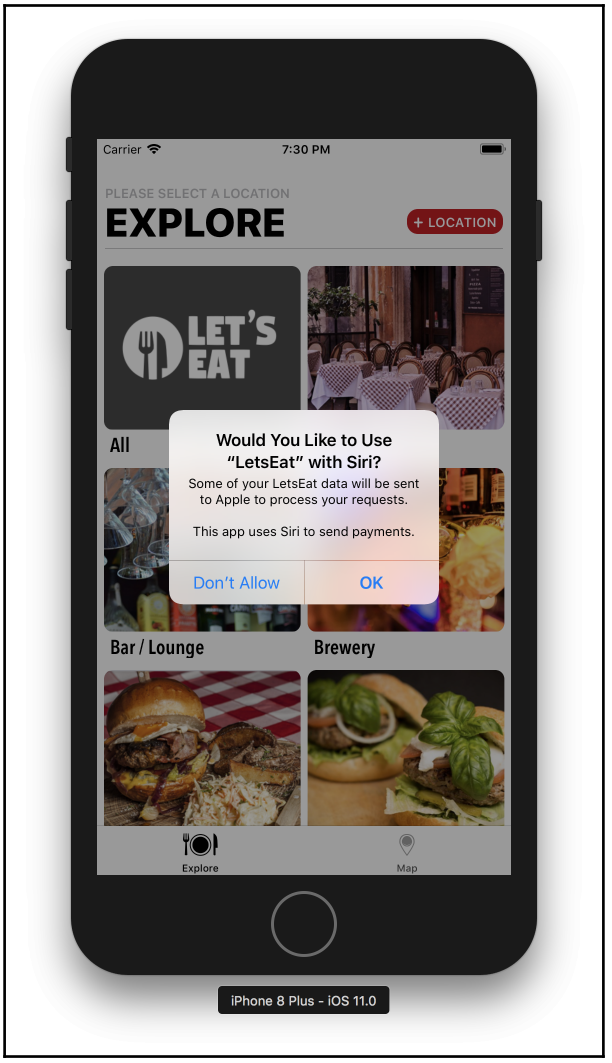

MakePayment
iPhone 8 Plus
LetsEat


MakePayment
iPhone 8 Plus
Build
2 targets
Run
Debug
Test
Debug
Profile
Release
Analyze
Debug
Archive
Release
Info
Arguments
Options
Diagnostics
Build Configuration
Debug
Executable
Ask on Launch
Debug executable
Siri Intent Query
Leave blank to be prompted on the device to speak your query.
Debug Process As
Me (craigclayton)
root
Launch
Automatically
Wait for executable to be launched
Duplicate Scheme
Manage Schemes...
Shared
Close

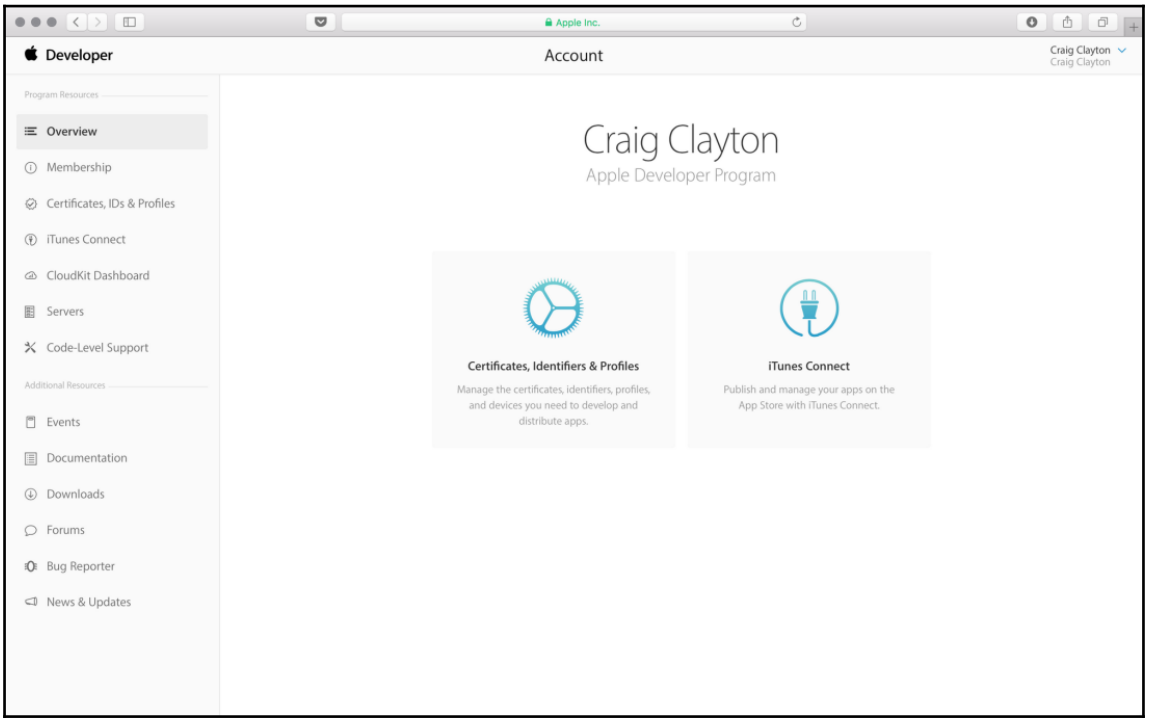

Carrier
7:47 PM
Send $100 to Jason Clayton for dinner last night using LetsEat
I'll need to access your LetsEat data to do that.
Is that OK?
No
Yes
iPhone 8 Plus - iOS 11.0

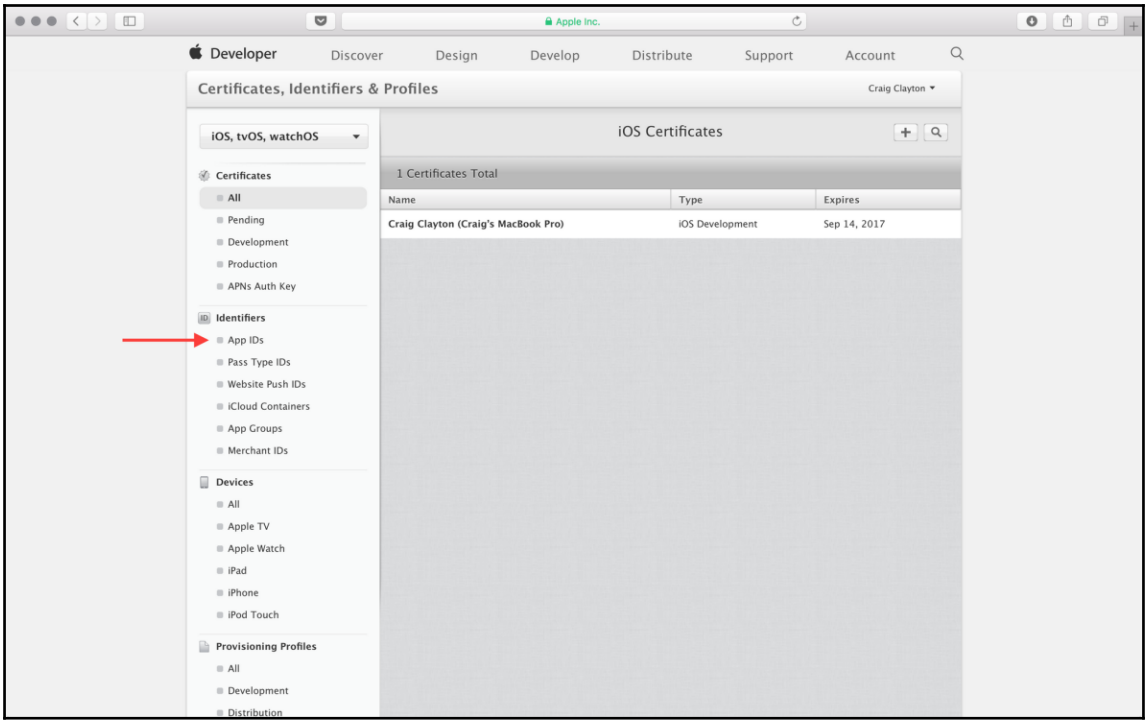

Carrier
7:47 PM
Here's your LetsEat payment of $100 to Jason Clayton. Do you want to send it?
LETSEAT
Send Money
$100
Note
dinner last night
Cancel
Send
iPhone 8 Plus - iOS 11.0

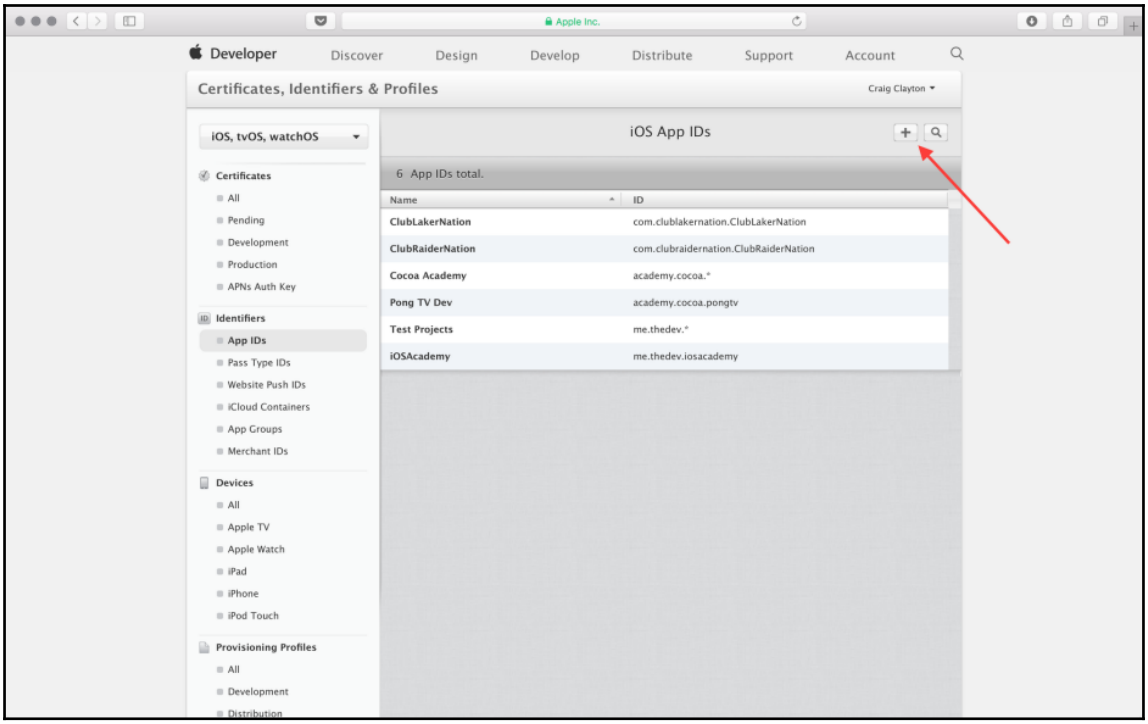

Carrier
7:47 PM
Here's your LetsEat payment of $100 to Jason Clayton. Do you want to send it?
LETSEAT
Money Sent
$100
Note
dinner last night
LetsEat sent your payment.
iPhone 8 Plus - iOS 11.0

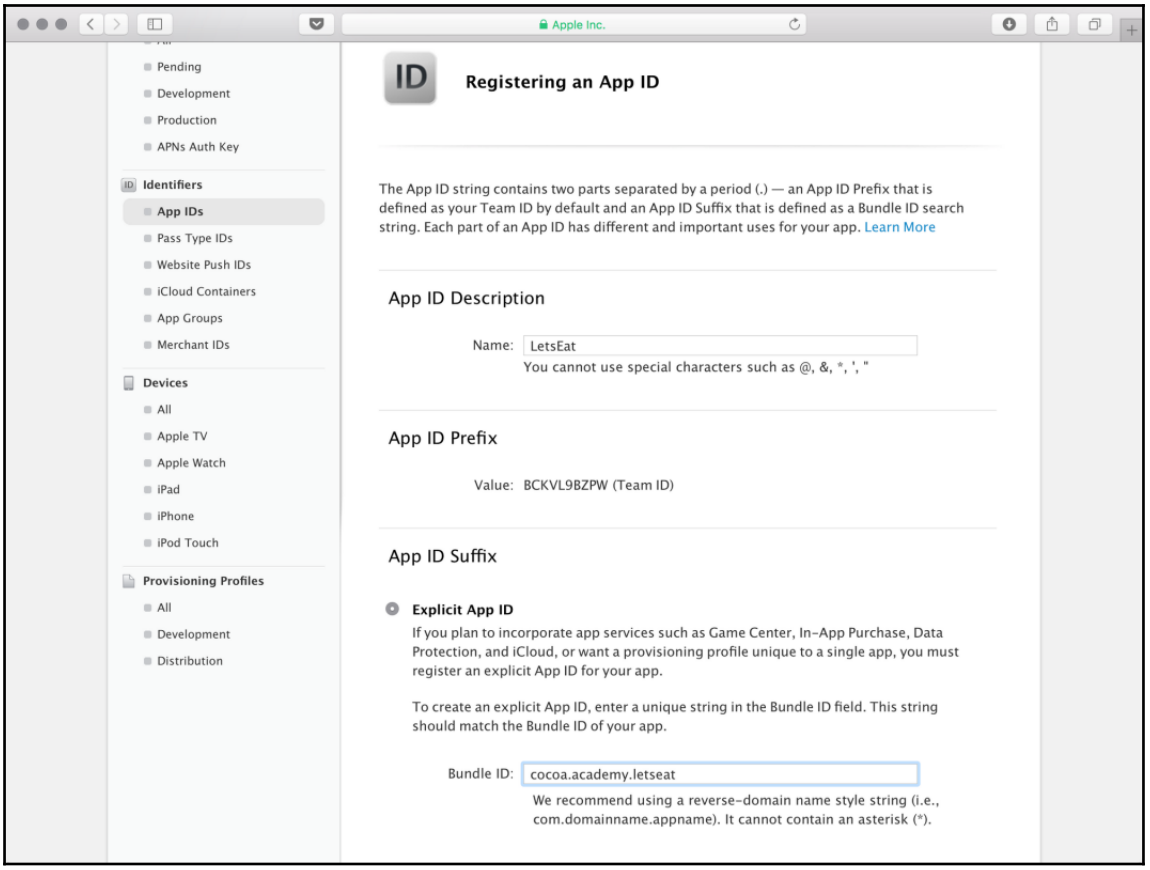

Carrier
7:30 PM
PLEASE SELECT A LOCATION
EXPLORE
+ LOCATION
LET'S EAT
All
Would You Like to Use "LetsEat" with Siri?
Some of your LetsEat data will be sent to Apple to process your requests.
This app uses Siri to send payments.
Don't Allow
OK
Bar / Lounge
Brewery
Explore
Map
iPhone 8 Plus - iOS 11.0

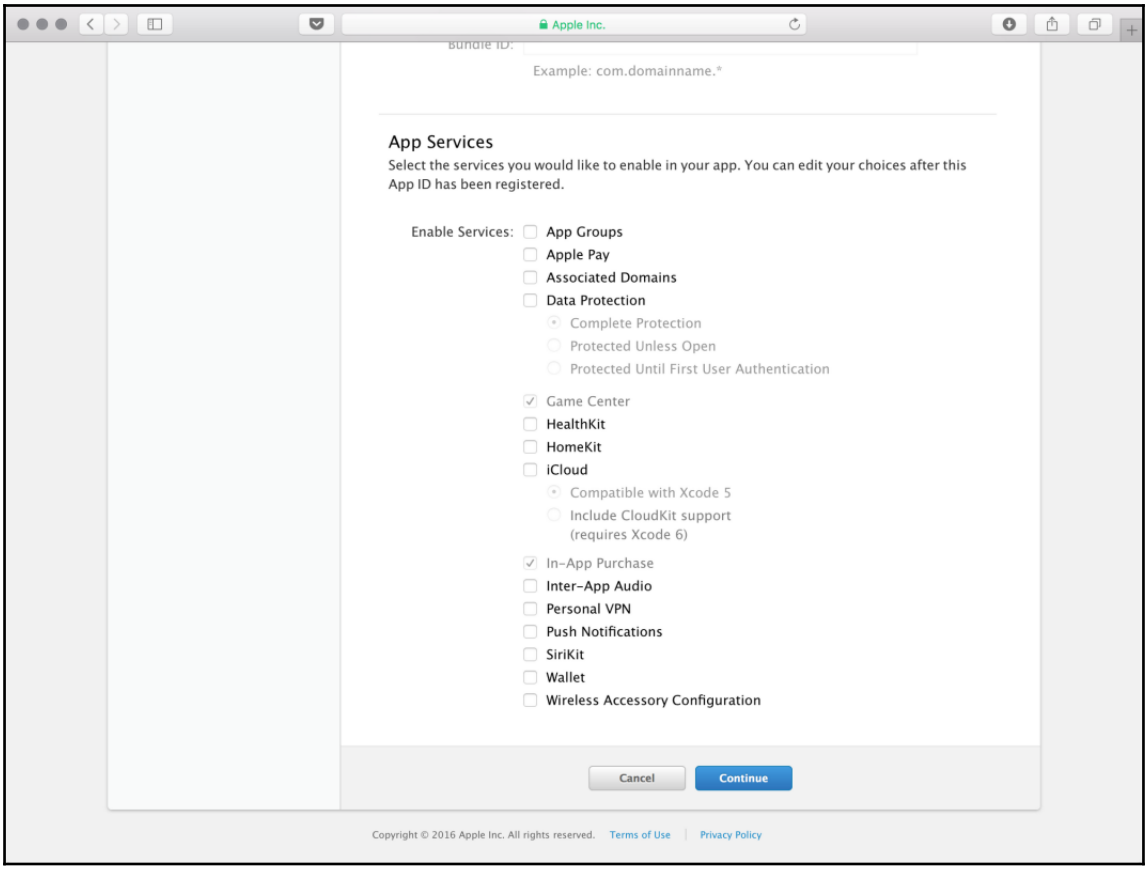
Chapter 29: Beta and Store Submission

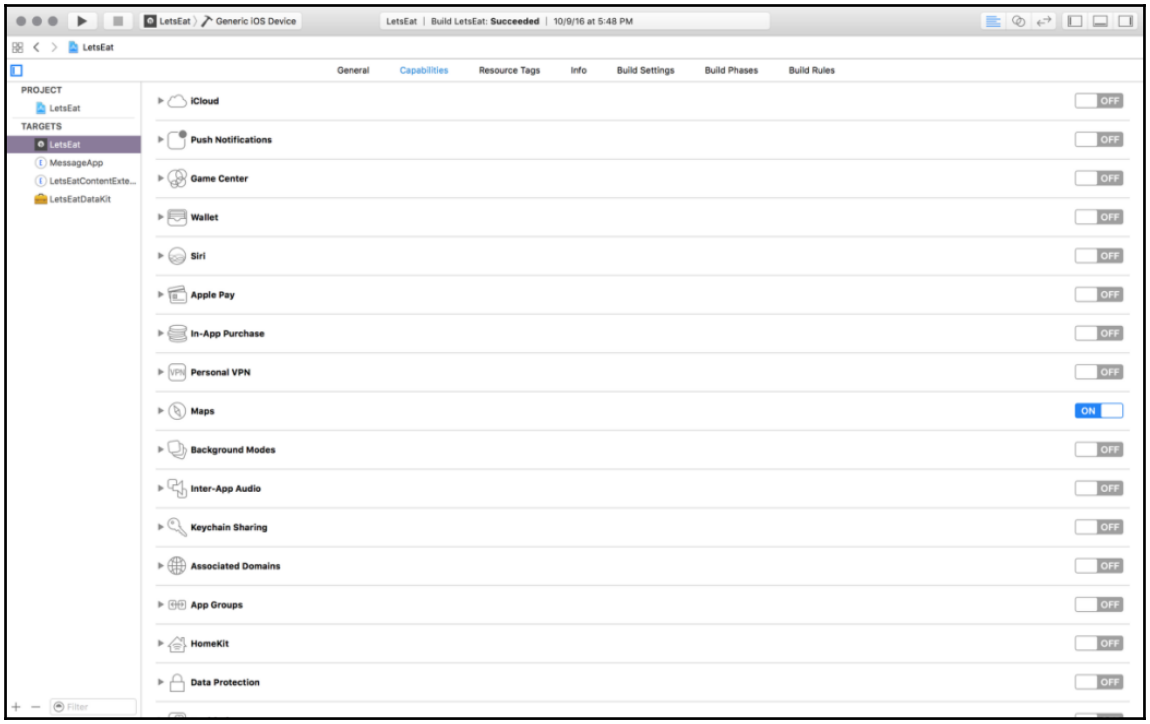


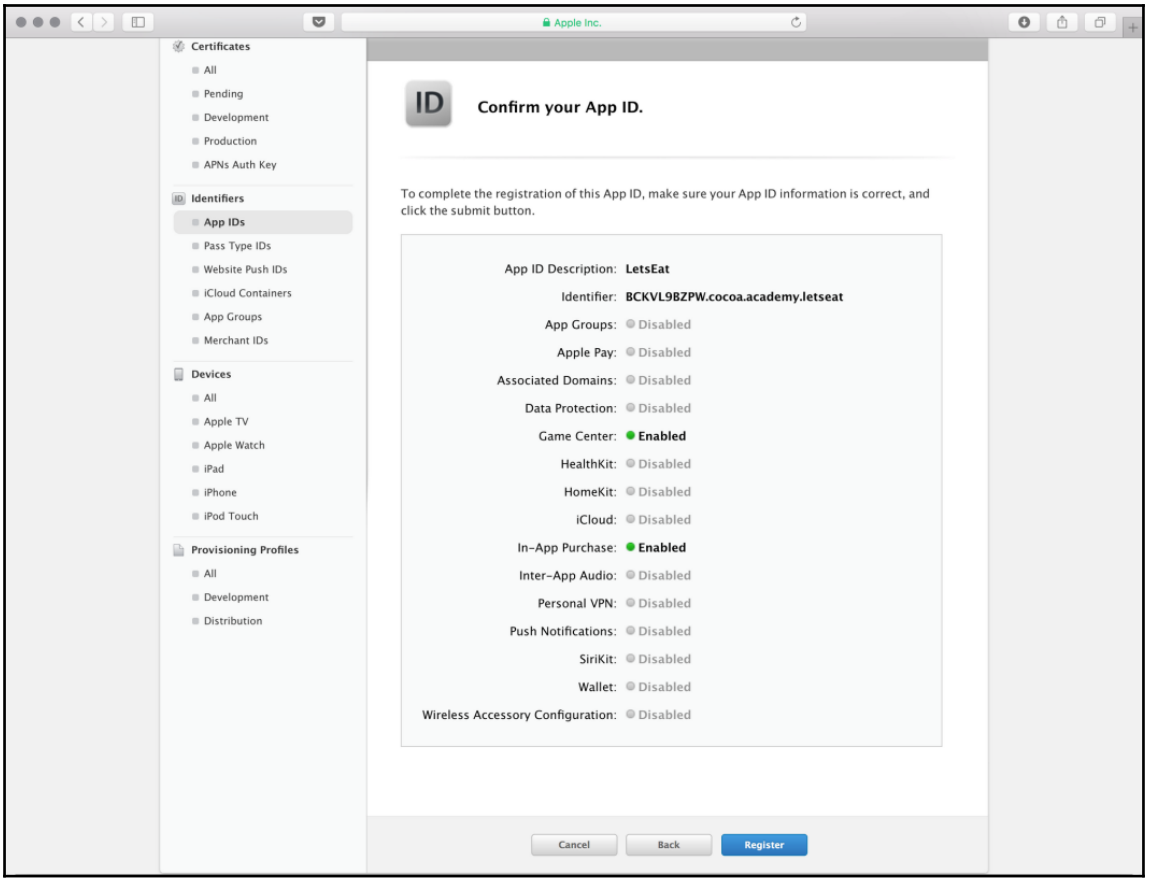

Apple Inc.
Developer
Discover
Design
Develop
Distribute
Support
Account
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
iOS Certificates
1 Certificates Total
Name
Type
Expires
Craig Clayton (Craig's MacBook Pro)
iOS Development
Sep 14, 2017

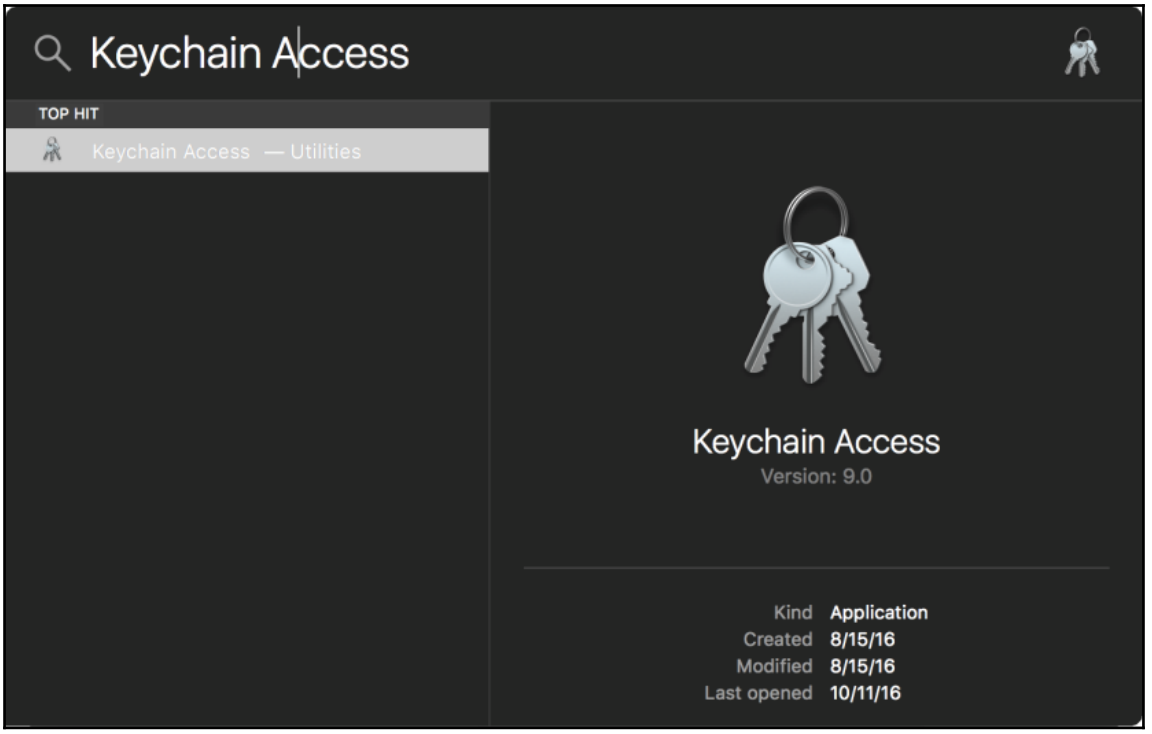

Apple Inc.
Developer
Discover
Design
Develop
Distribute
Support
Account
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
iOS App IDs
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
6 App IDs total.
Name
ID
ClubLakerNation
com.clublakernation.ClubLakerNation
ClubRaiderNation
com.clubraidernation.ClubRaiderNation
Cocoa Academy
academy.cocoa.*
Pong TV Dev
academy.cocoa.pongtv
Test Projects
me.thedev.*
iOSAcademy
me.thedev.iosacademy

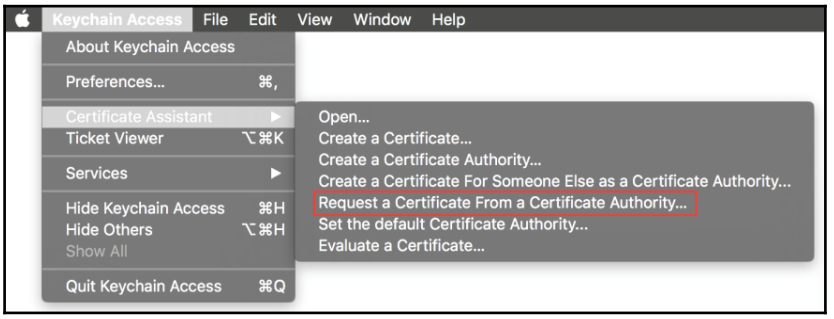

Apple Inc.
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Registering an App ID
The App ID string contains two parts separated by a period (.) — an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. Learn More
App ID Description
Name: LetsEat
You cannot use special characters such as @, &, *, ', "
App ID Prefix
Value: BCKVL9BZPW (Team ID)
App ID Suffix
Explicit App ID
If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.
To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.
Bundle ID: cocoa.academy.letseat
We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

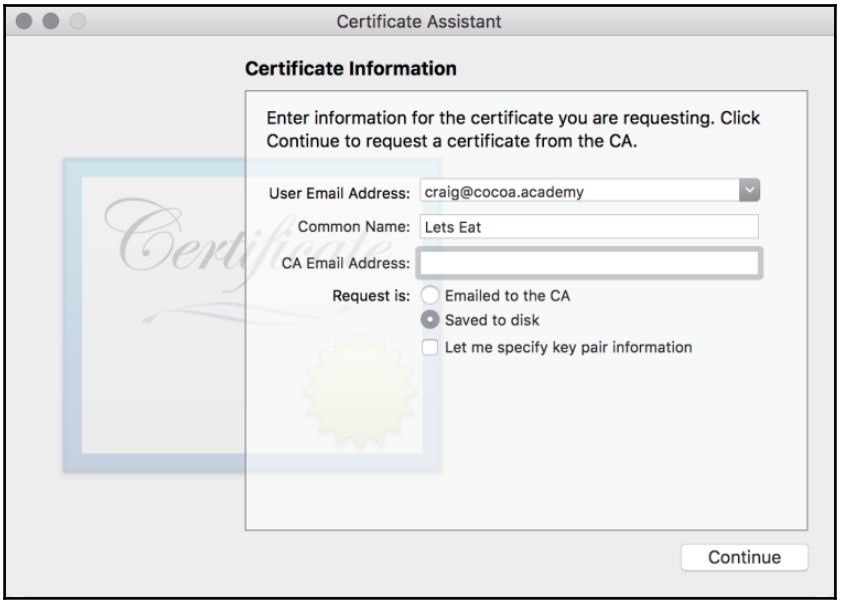

Apple Inc.
Bundle ID:
Example: com.domainname.*
App Services
Select the services you would like to enable in your app. You can edit your choices after this App ID has been registered.
Enable Services:
App Groups
Apple Pay
Associated Domains
Data Protection
Complete Protection
Protected Unless Open
Protected Until First User Authentication
Game Center
HealthKit
HomeKit
iCloud
Compatible with Xcode 5
Include CloudKit support (requires Xcode 6)
In-App Purchase
Inter-App Audio
Personal VPN
Push Notifications
SiriKit
Wallet
Wireless Accessory Configuration
Cancel
Continue
Copyright © 2016 Apple Inc. All rights reserved.
Terms of Use
Privacy Policy

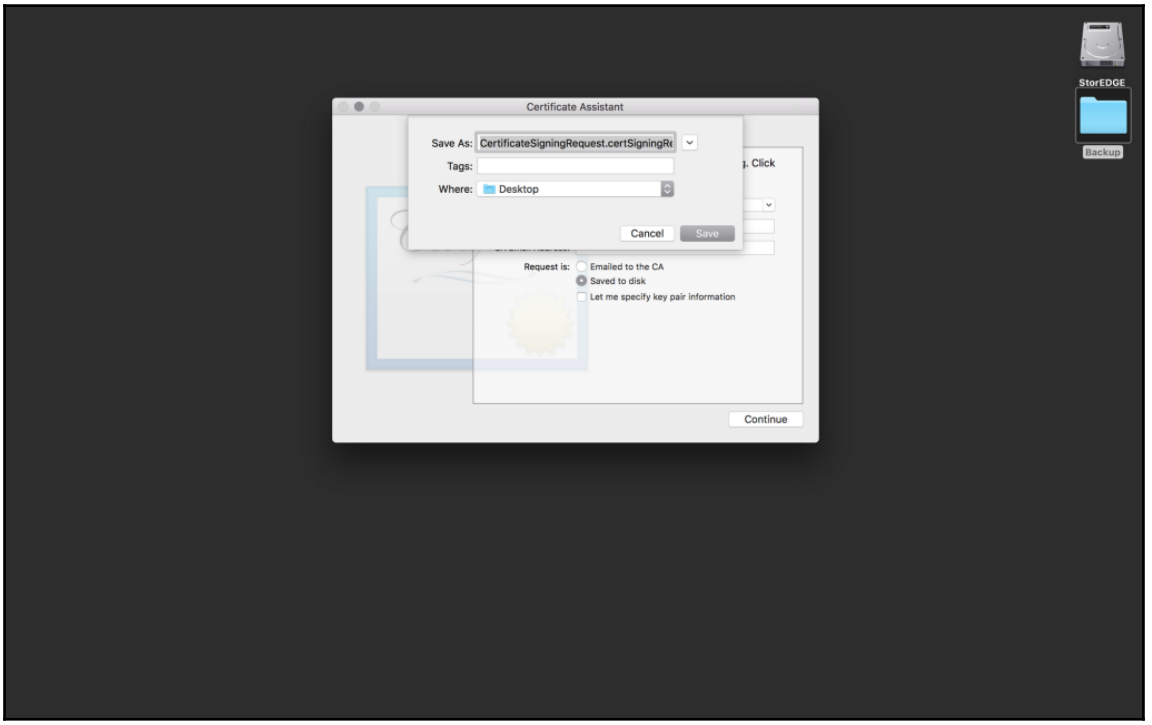

LetsEat
Generic iOS Device
LetsEat | Build LetsEat: Succeeded | 10/9/16 at 5:48 PM
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
LetsEat
TARGETS
LetsEat
MessageApp
LetsEatContentExte...
LetsEatDataKit
iCloud
Push Notifications
Game Center
Wallet
Siri
Apple Pay
In-App Purchase
Personal VPN
Maps
Background Modes
Inter-App Audio
Keychain Sharing
Associated Domains
App Groups
HomeKit
Data Protection
OFF
ON

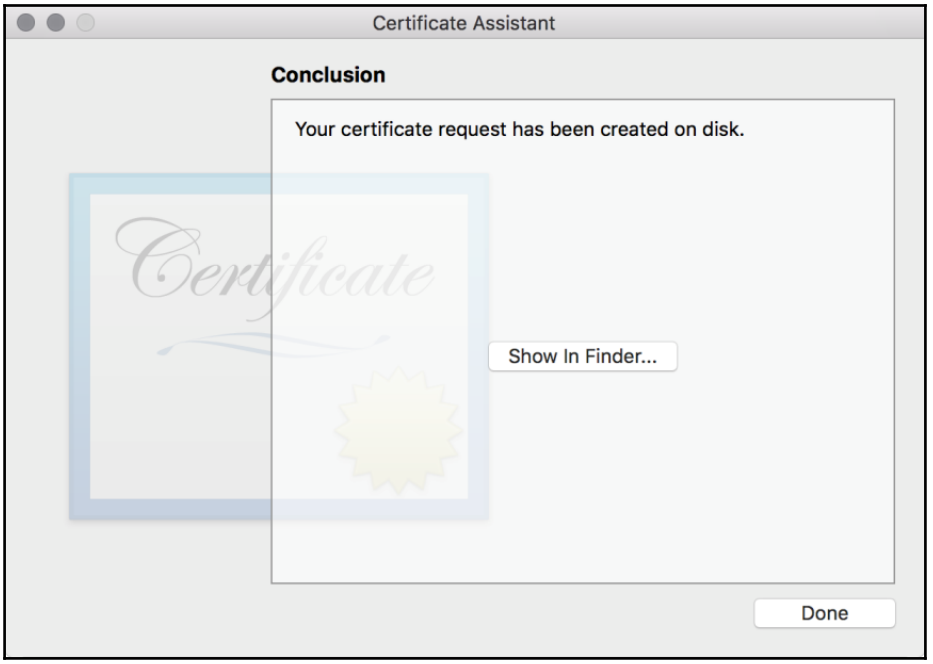

Apple Inc.
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
ID
Confirm your App ID.
To complete the registration of this App ID, make sure your App ID information is correct, and click the submit button.
App ID Description: LetsEat
Identifier: BCKVL9BZPW.cocoa.academy.letseat
App Groups: Disabled
Apple Pay: Disabled
Associated Domains: Disabled
Data Protection: Disabled
Game Center: Enabled
HealthKit: Disabled
HomeKit: Disabled
iCloud: Disabled
In-App Purchase: Enabled
Inter-App Audio: Disabled
Personal VPN: Disabled
Push Notifications: Disabled
SiriKit: Disabled
Wallet: Disabled
Wireless Accessory Configuration: Disabled
Cancel
Back
Register

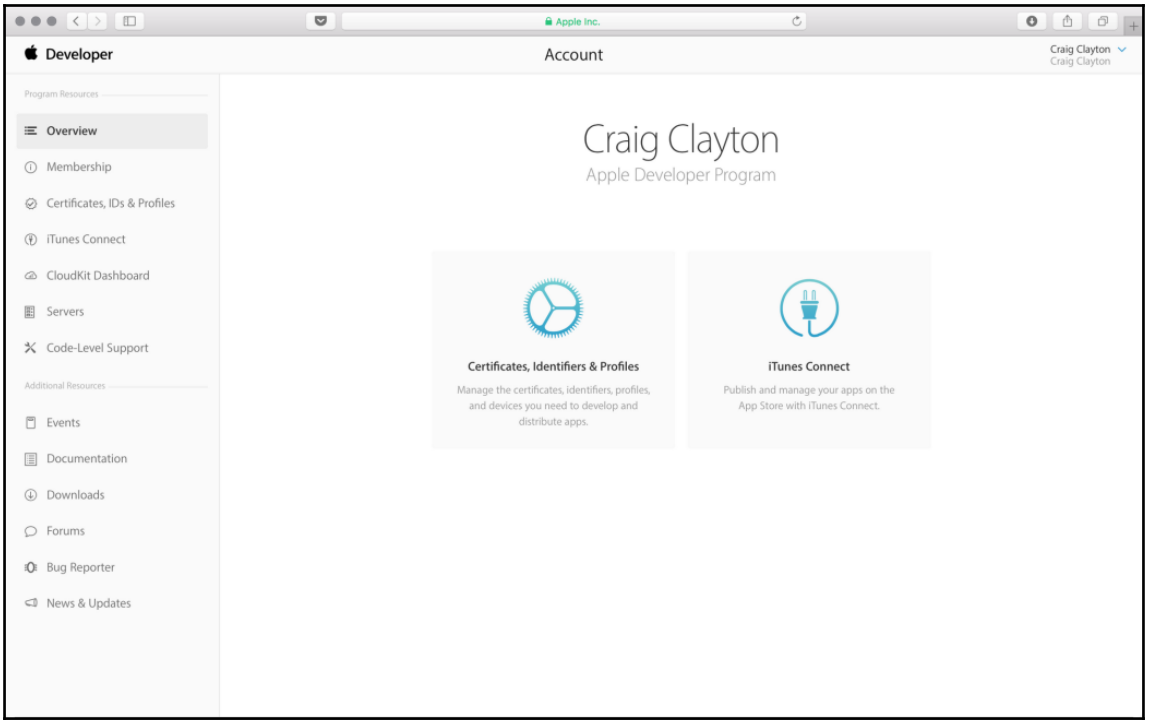

Keychain Access
TOP HIT
Keychain Access — Utilities
Keychain Access
Version: 9.0
Kind Application
Created 8/15/16
Modified 8/15/16
Last opened 10/11/16

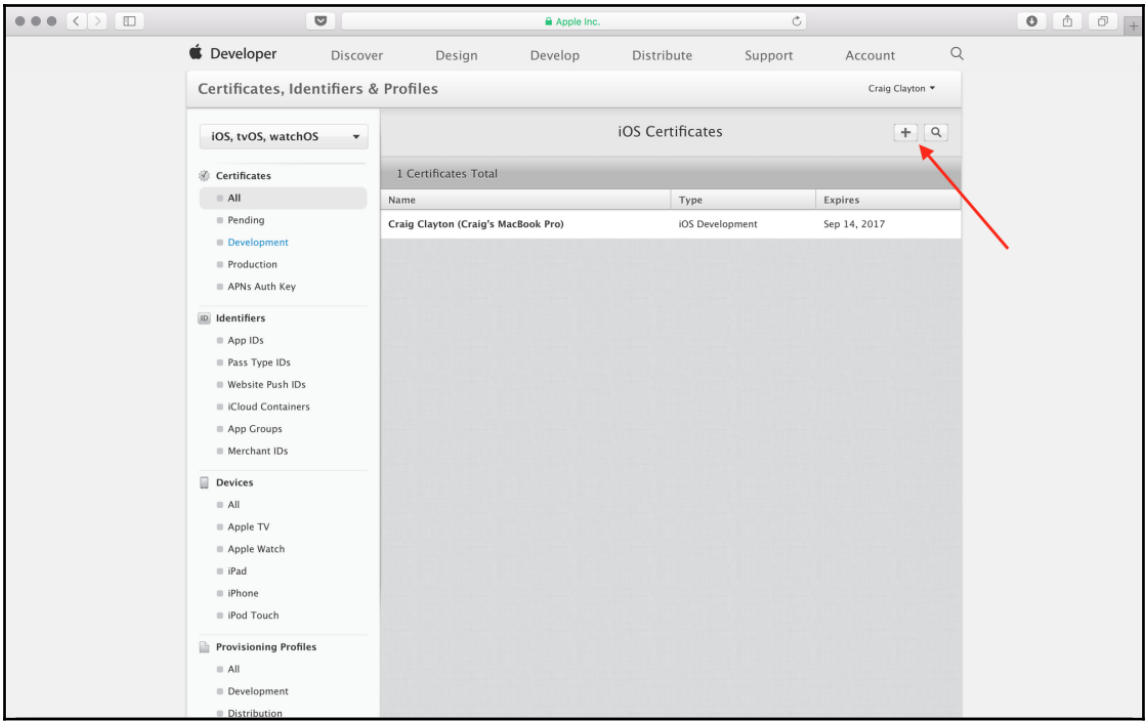

Keychain Access
File
Edit
View
Window
Help
About Keychain Access
Preferences... ⌘,
Certificate Assistant
Ticket Viewer ⌥⌘K
Services
Hide Keychain Access ⌘H
Hide Others ⌥⌘H
Show All
Quit Keychain Access ⌘Q
Open...
Create a Certificate...
Create a Certificate Authority...
Create a Certificate For Someone Else as a Certificate Authority...
Request a Certificate From a Certificate Authority...
Set the default Certificate Authority...
Evaluate a Certificate...

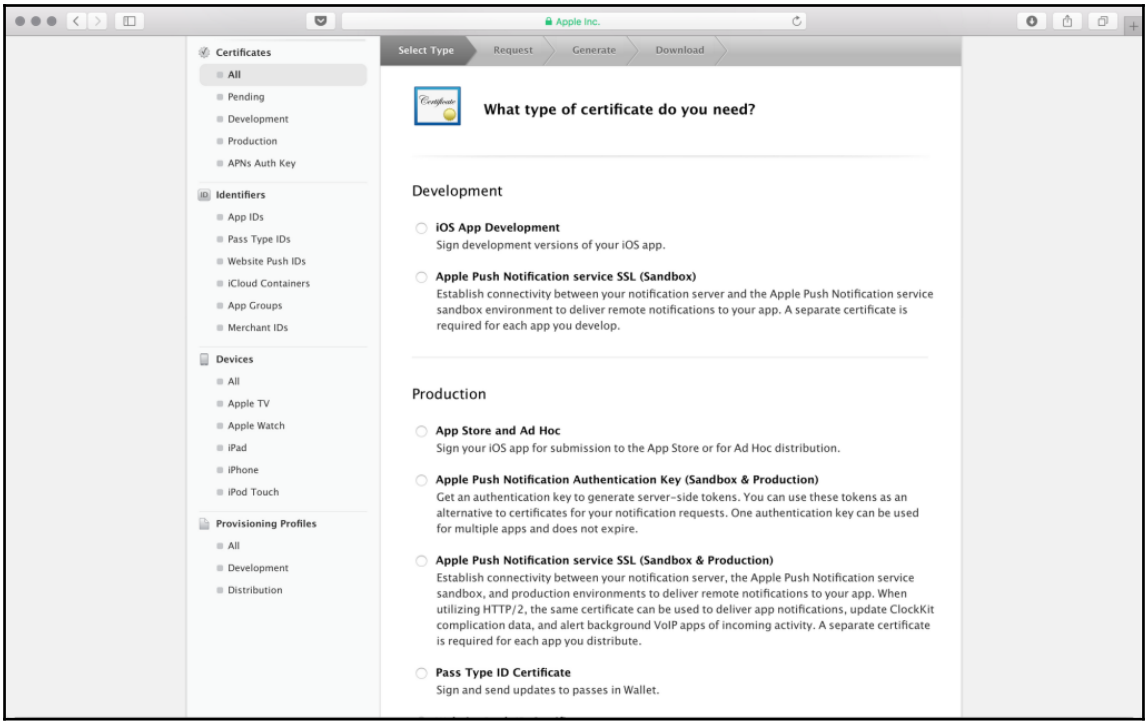

Certificate Assistant
Certificate Information
Enter information for the certificate you are requesting. Click Continue to request a certificate from the CA.
User Email Address: craig@cocoa.academy
Common Name: Lets Eat
CA Email Address:
Request is: Emailed to the CA
Saved to disk
Let me specify key pair information
Continue

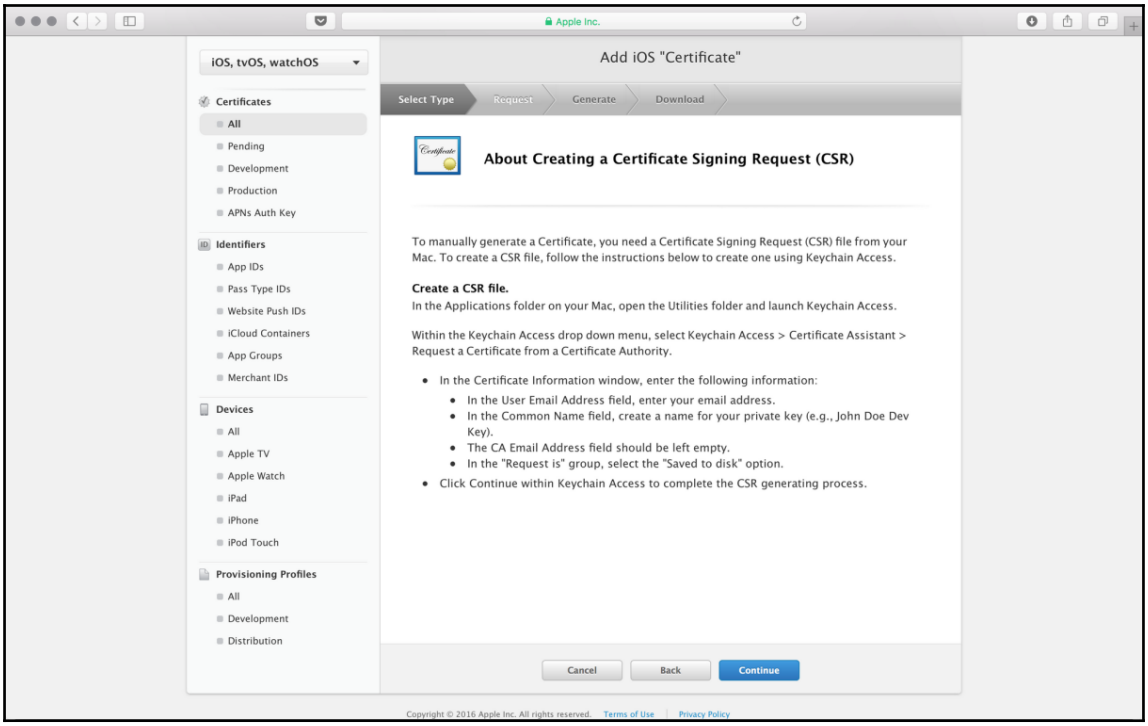

StorEDGE
Backup
Certificate Assistant
Save As: CertificateSigningRequest.certSigningRe
Tags:
Where: Desktop
Cancel
Save
Request is: Emailed to the CA
Saved to disk
Let me specify key pair information
Continue

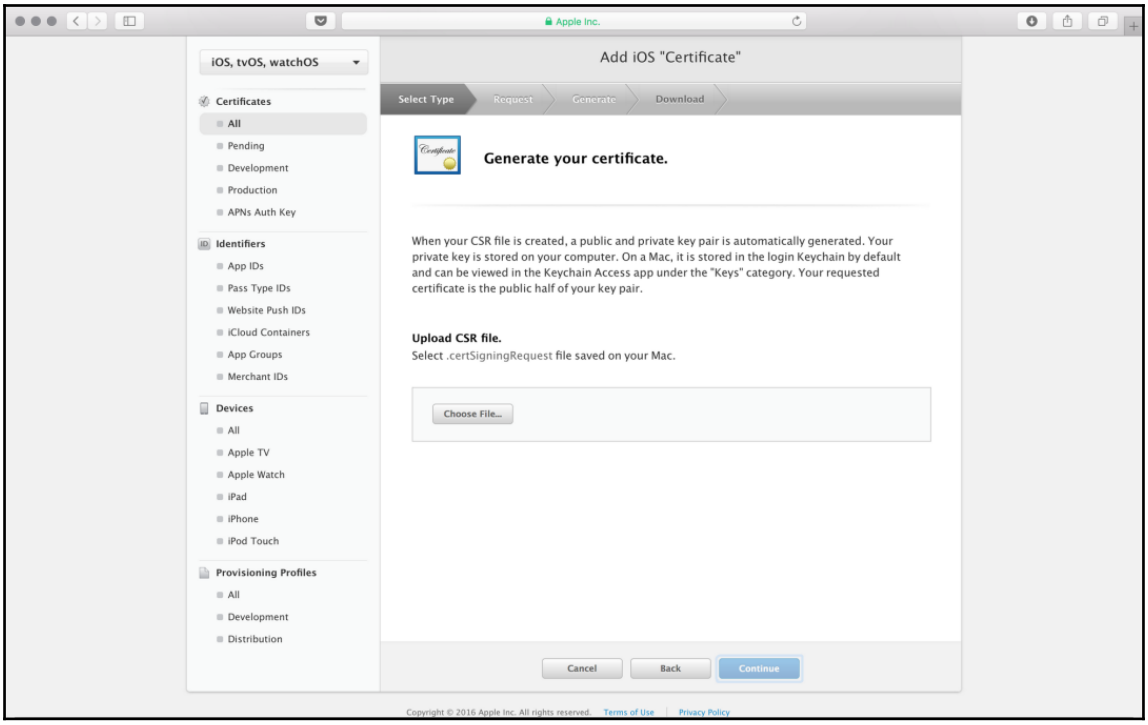

Certificate Assistant
Conclusion
Your certificate request has been created on disk.
Show In Finder...
Done

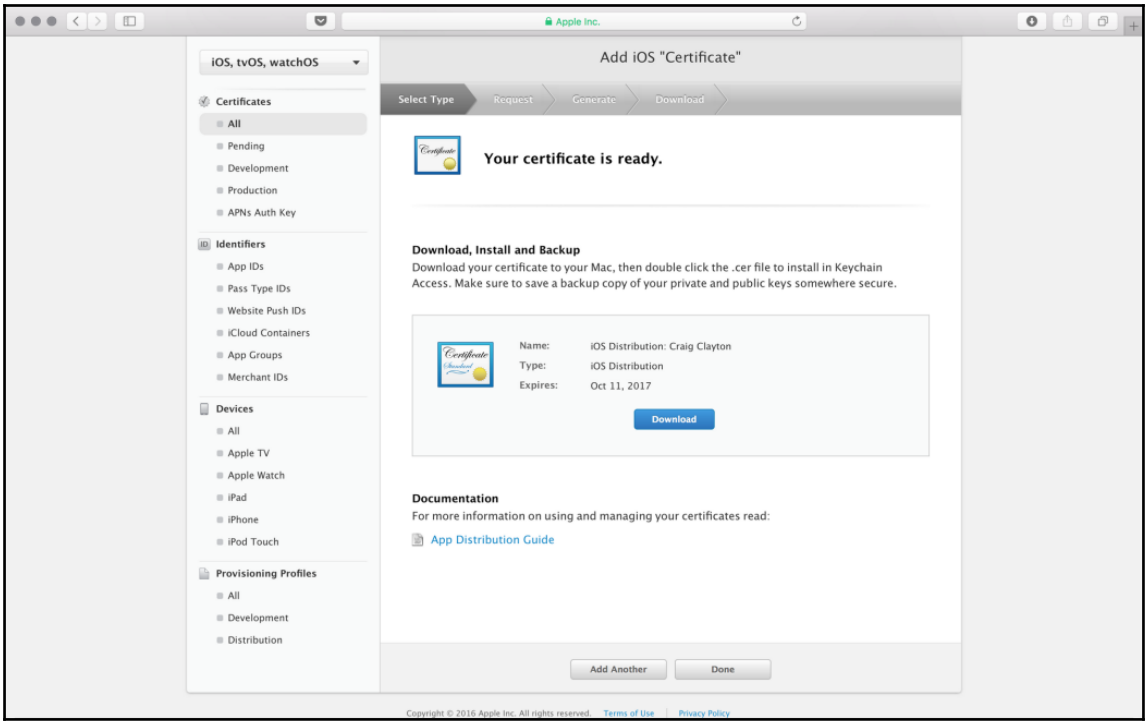

Apple Inc.
Developer
Account
Craig Clayton
Craig Clayton
Program Resources
Overview
Membership
Certificates, IDs & Profiles
iTunes Connect
CloudKit Dashboard
Servers
Code-Level Support
Additional Resources
Events
Documentation
Downloads
Forums
Bug Reporter
News & Updates
Craig Clayton
Apple Developer Program
Certificates, Identifiers & Profiles
Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.
iTunes Connect
Publish and manage your apps on the App Store with iTunes Connect.

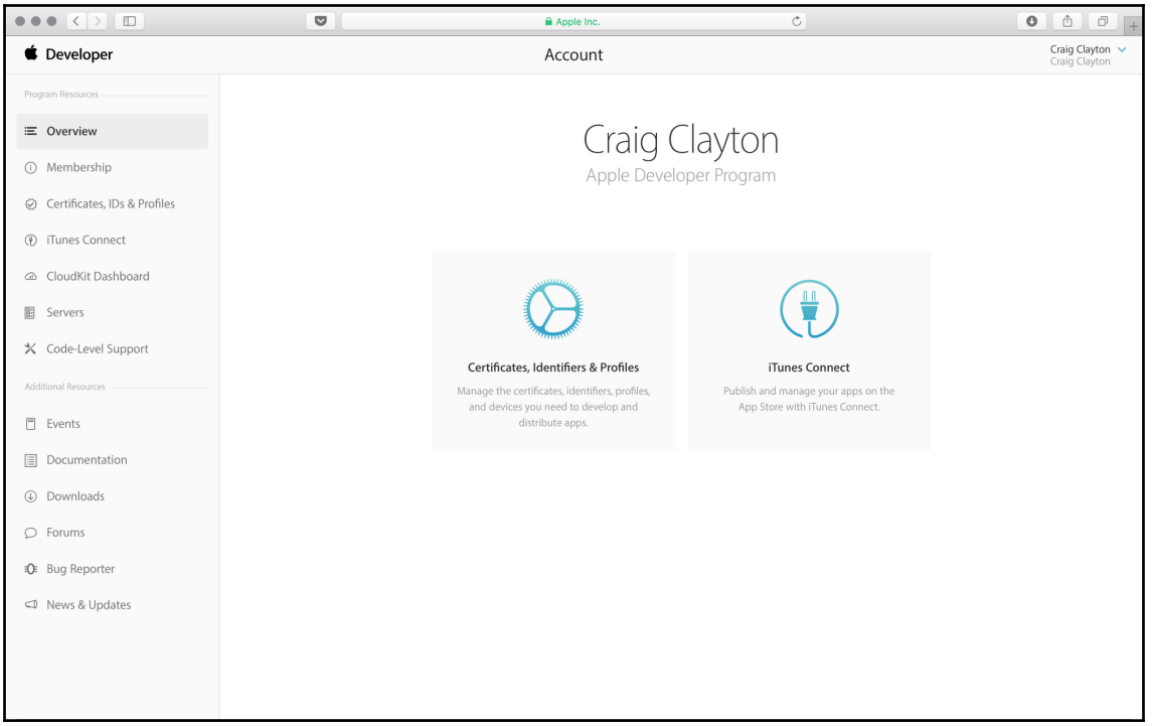

Apple Inc.
Developer
Discover
Design
Develop
Distribute
Support
Account
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
iOS Certificates
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
1 Certificates Total
Name
Type
Expires
Craig Clayton (Craig's MacBook Pro)
iOS Development
Sep 14, 2017

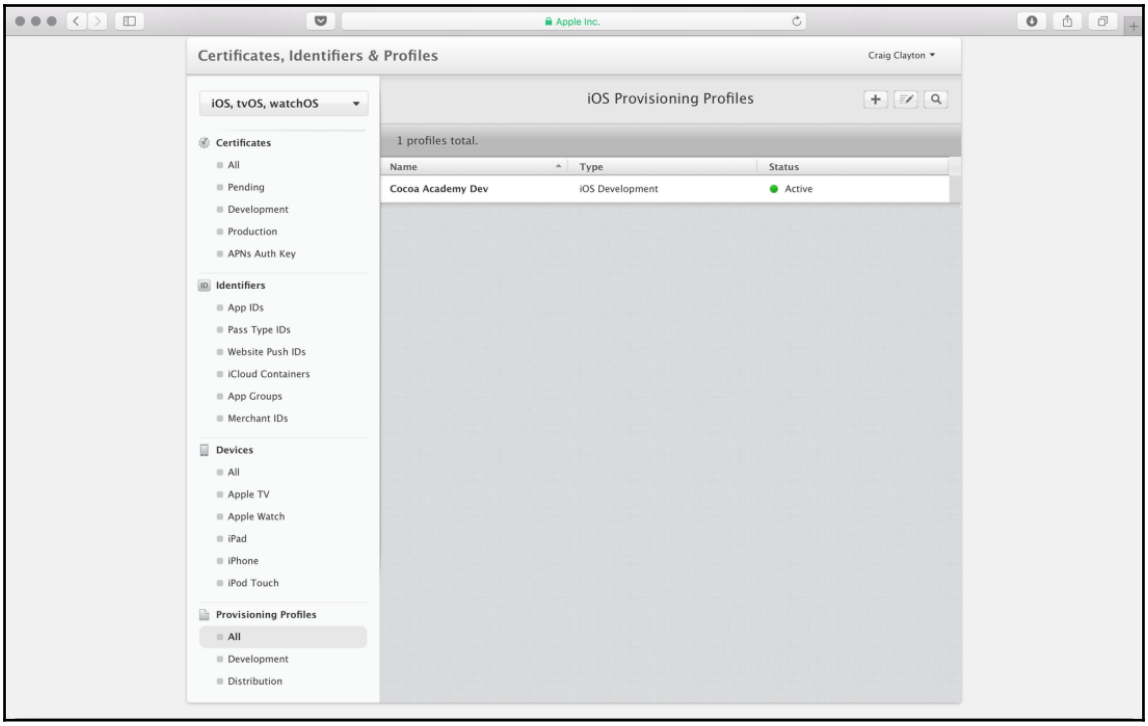

Apple Inc.
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Select Type
Request
Generate
Download
What type of certificate do you need?
Development
iOS App Development
Sign development versions of your iOS app.
Apple Push Notification service SSL (Sandbox)
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment to deliver remote notifications to your app. A separate certificate is required for each app you develop.
Production
App Store and Ad Hoc
Sign your iOS app for submission to the App Store or for Ad Hoc distribution.
Apple Push Notification Authentication Key (Sandbox & Production)
Get an authentication key to generate server-side tokens. You can use these tokens as an alternative to certificates for your notification requests. One authentication key can be used for multiple apps and does not expire.
Apple Push Notification service SSL (Sandbox & Production)
Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environments to deliver remote notifications to your app. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.
Pass Type ID Certificate
Sign and send updates to passes in Wallet.

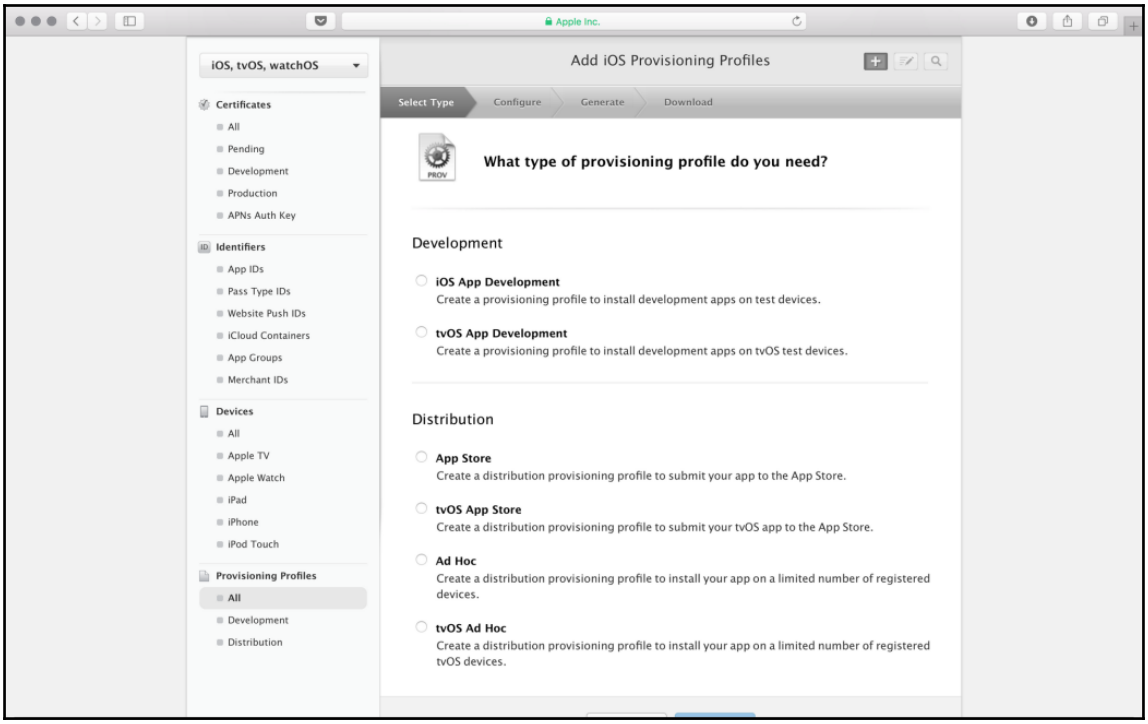

Apple Inc.
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS "Certificate"
Select Type
Request
Generate
Download
About Creating a Certificate Signing Request (CSR)
To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.
Create a CSR file.
In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.
Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.
In the Certificate Information window, enter the following information:
In the User Email Address field, enter your email address.
In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
The CA Email Address field should be left empty.
In the "Request is" group, select the "Saved to disk" option.
Click Continue within Keychain Access to complete the CSR generating process.
Cancel
Back
Continue
Copyright © 2016 Apple Inc. All rights reserved.
Terms of Use
Privacy Policy

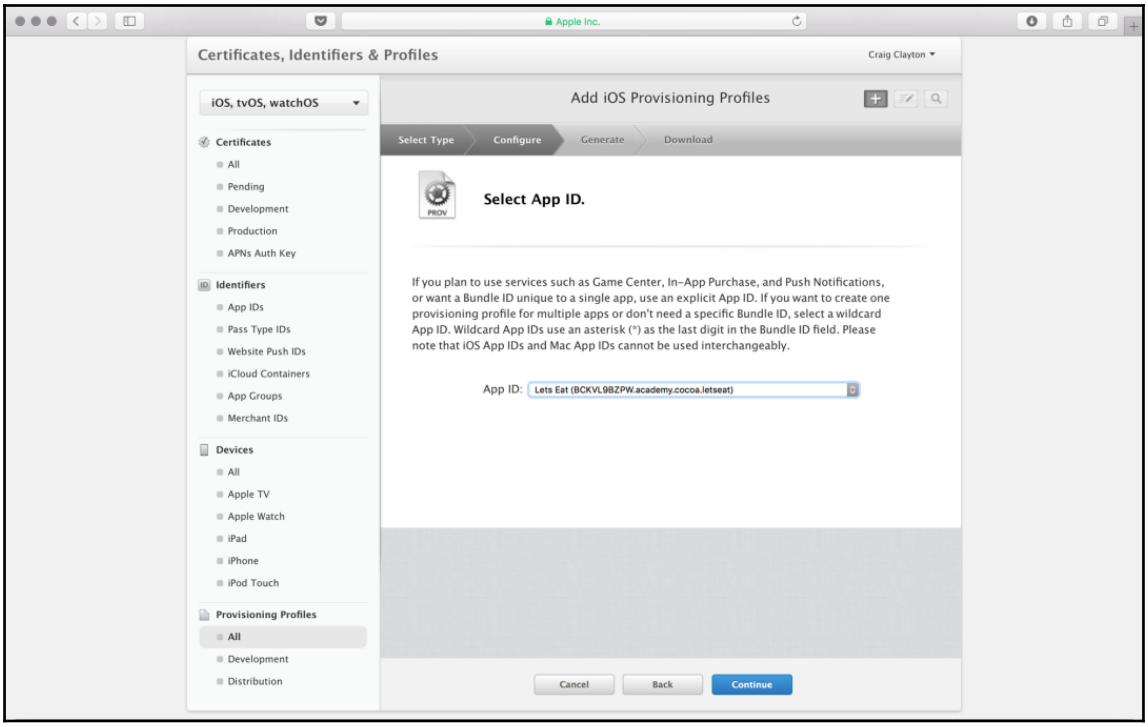

Apple Inc.
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS "Certificate"
Select Type
Request
Generate
Download
Generate your certificate.
When your CSR file is created, a public and private key pair is automatically generated. Your private key is stored on your computer. On a Mac, it is stored in the login Keychain by default and can be viewed in the Keychain Access app under the "Keys" category. Your requested certificate is the public half of your key pair.
Upload CSR file.
Select .certSigningRequest file saved on your Mac.
Choose File...
Cancel
Back
Continue
Copyright © 2016 Apple Inc. All rights reserved.
Terms of Use
Privacy Policy

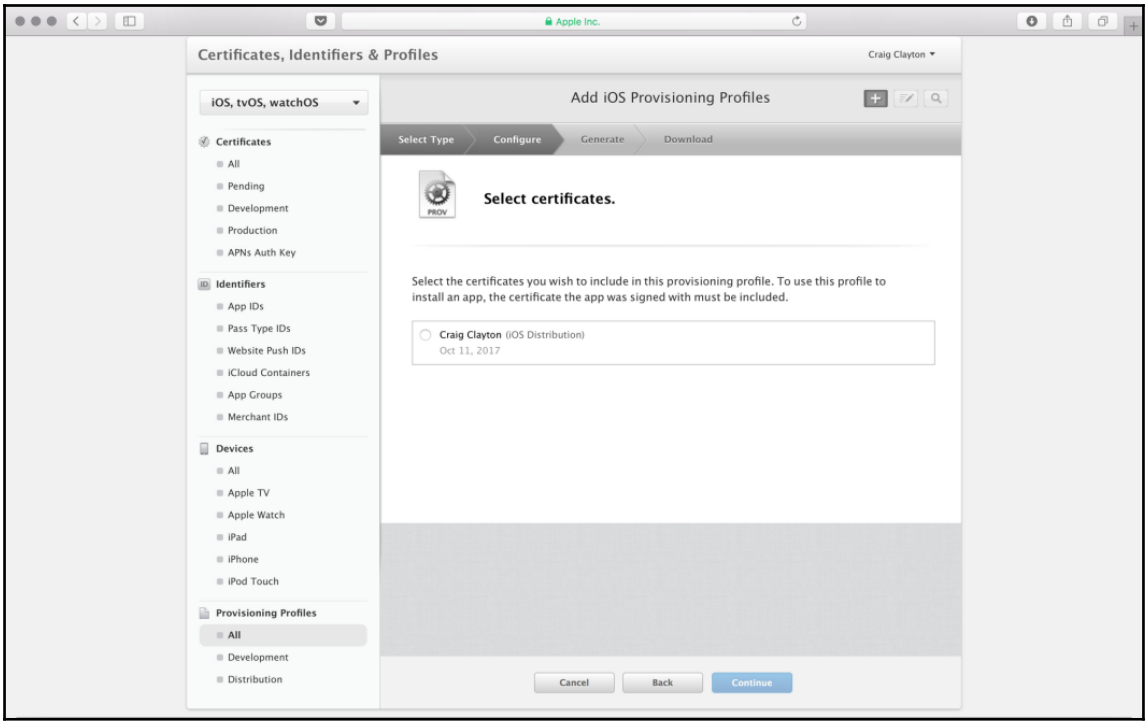

Apple Inc.
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS "Certificate"
Select Type
Request
Generate
Download
Your certificate is ready.
Download, Install and Backup
Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.
Name: iOS Distribution: Craig Clayton
Type: iOS Distribution
Expires: Oct 11, 2017
Download
Documentation
For more information on using and managing your certificates read:
App Distribution Guide
Add Another
Done
Copyright © 2016 Apple Inc. All rights reserved.
Terms of Use
Privacy Policy

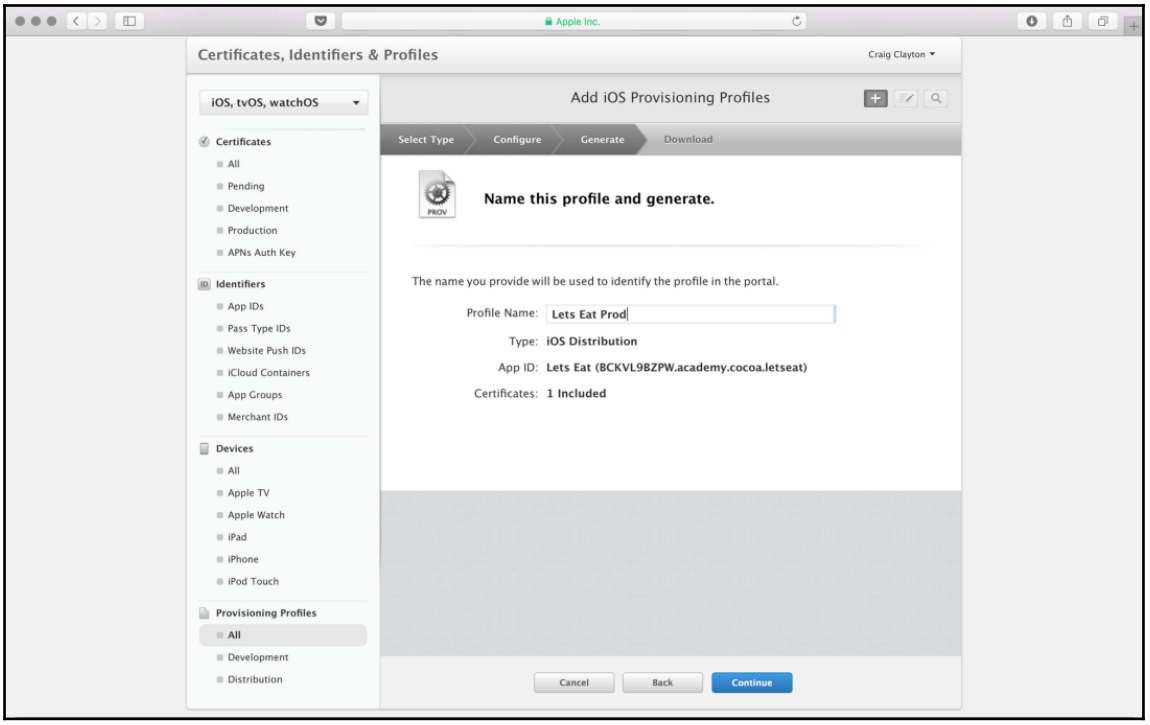

Apple Inc.
Developer
Account
Craig Clayton
Craig Clayton
Program Resources
Overview
Membership
Certificates, IDs & Profiles
iTunes Connect
CloudKit Dashboard
Servers
Code-Level Support
Additional Resources
Events
Documentation
Downloads
Forums
Bug Reporter
News & Updates
Craig Clayton
Apple Developer Program
Certificates, Identifiers & Profiles
Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.
iTunes Connect
Publish and manage your apps on the App Store with iTunes Connect.

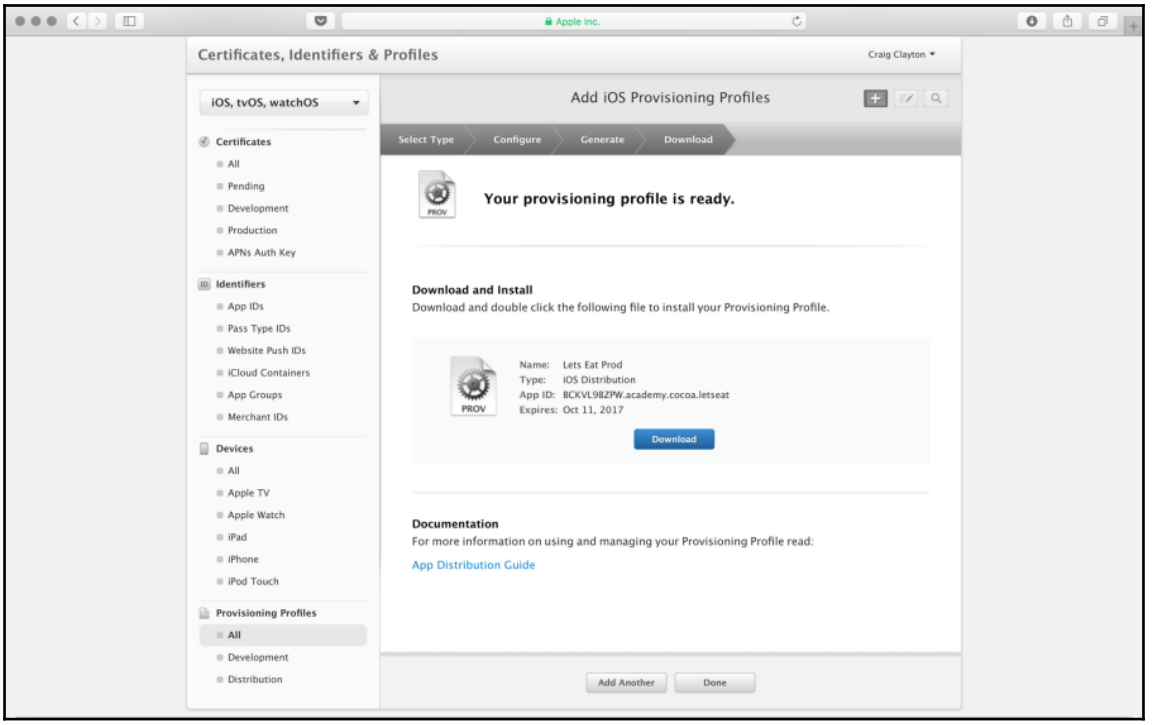

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
iOS Provisioning Profiles
1 profiles total.
Name
Type
Status
Cocoa Academy Dev
iOS Development
Active

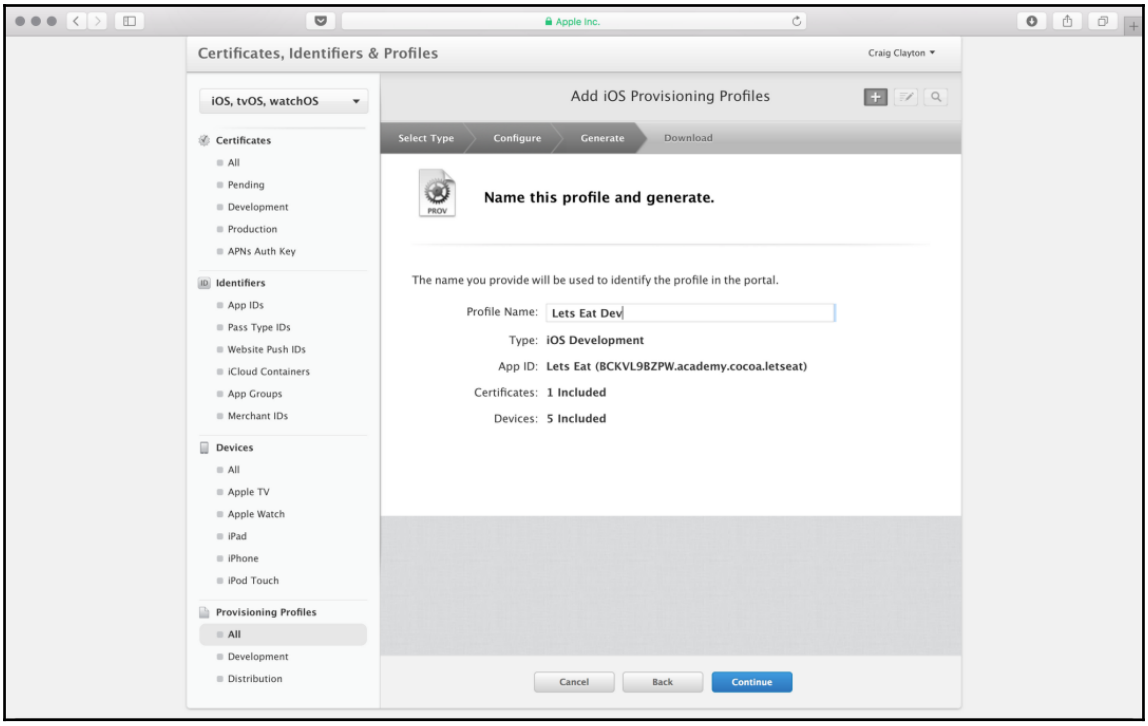

Apple Inc.
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
What type of provisioning profile do you need?
Development
iOS App Development
Create a provisioning profile to install development apps on test devices.
tvOS App Development
Create a provisioning profile to install development apps on tvOS test devices.
Distribution
App Store
Create a distribution provisioning profile to submit your app to the App Store.
tvOS App Store
Create a distribution provisioning profile to submit your tvOS app to the App Store.
Ad Hoc
Create a distribution provisioning profile to install your app on a limited number of registered devices.
tvOS Ad Hoc
Create a distribution provisioning profile to install your app on a limited number of registered tvOS devices.

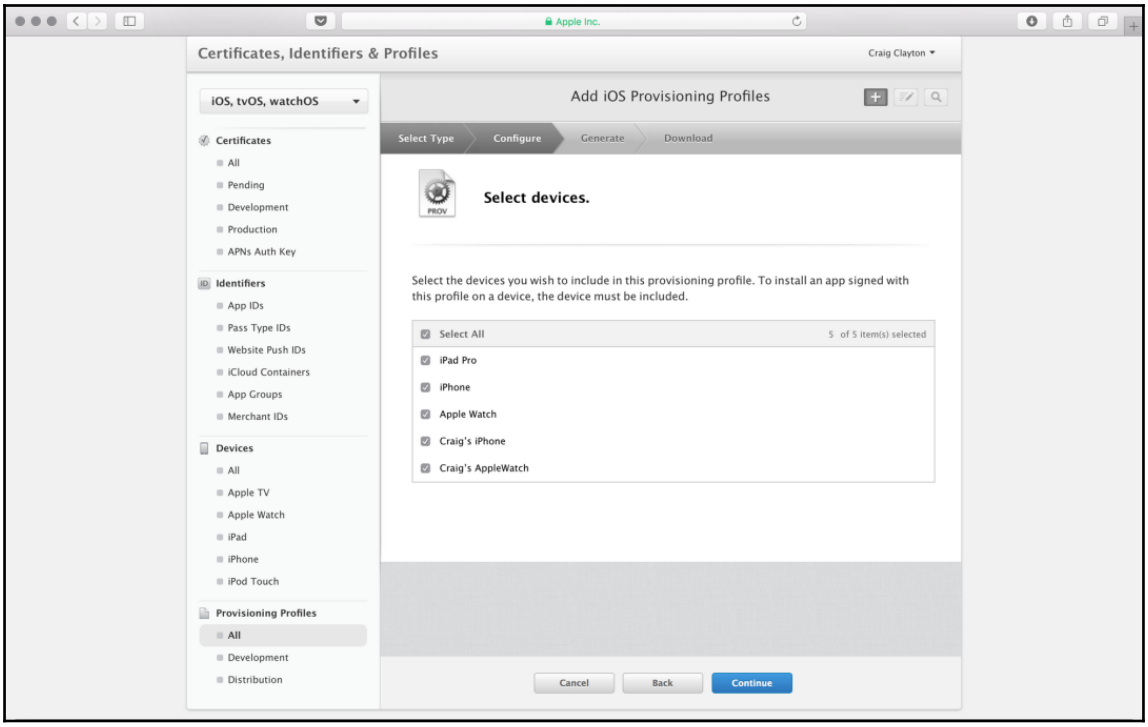

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
Select App ID.
If you plan to use services such as Game Center, In-App Purchase, and Push Notifications, or want a Bundle ID unique to a single app, use an explicit App ID. If you want to create one provisioning profile for multiple apps or don't need a specific Bundle ID, select a wildcard App ID. Wildcard App IDs use an asterisk (*) as the last digit in the Bundle ID field. Please note that iOS App IDs and Mac App IDs cannot be used interchangeably.
App ID:
Lets Eat (BCKVL9BZPW.academy.cocoa.letseat)
Cancel
Back
Continue

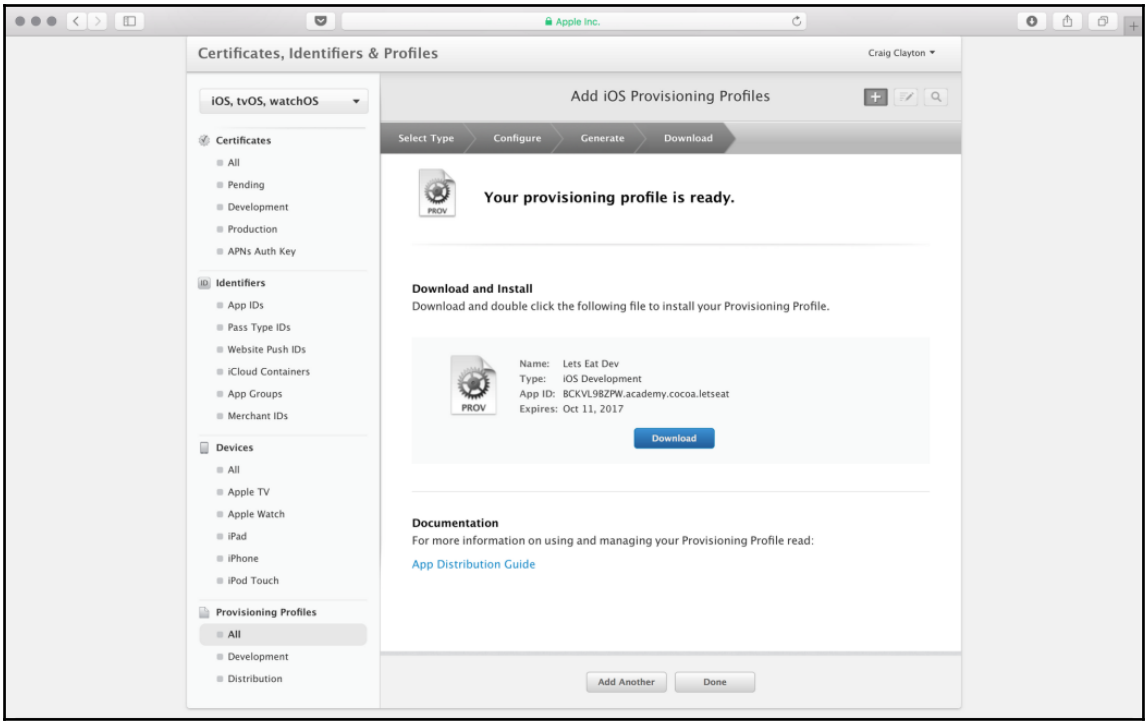

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
Select certificates.
Select the certificates you wish to include in this provisioning profile. To use this profile to install an app, the certificate the app was signed with must be included.
Craig Clayton (iOS Distribution)
Oct 11, 2017
Cancel
Back
Continue

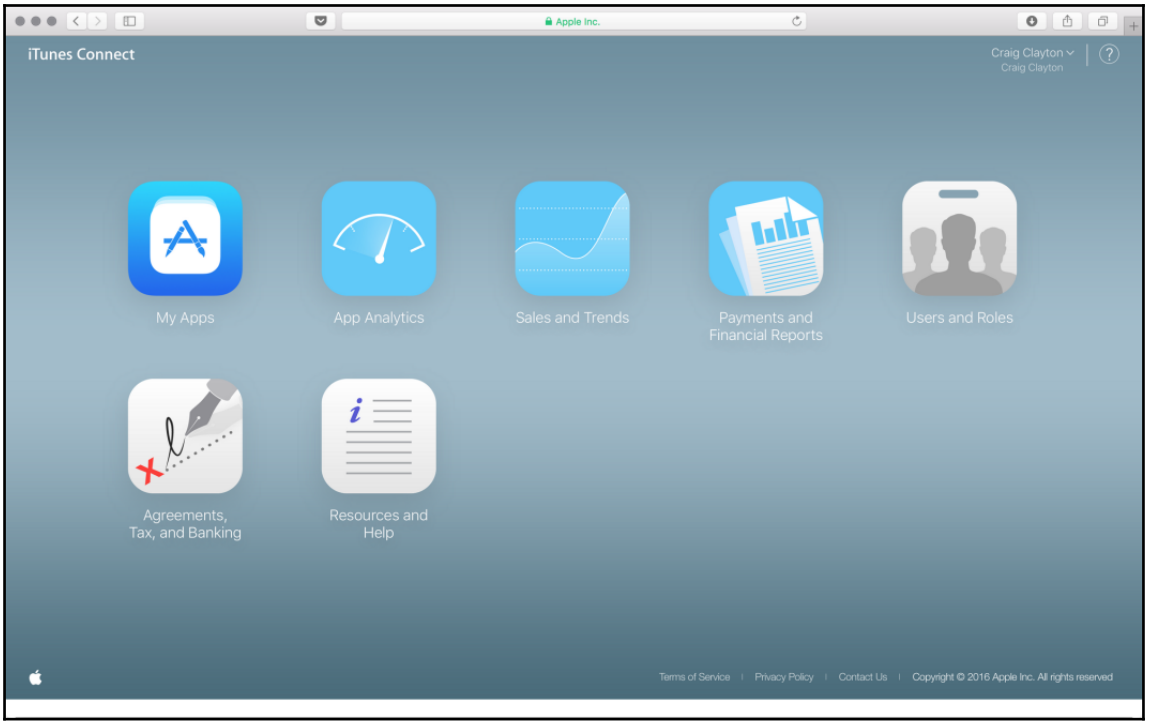

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
Name this profile and generate.
The name you provide will be used to identify the profile in the portal.
Profile Name: Lets Eat Prod
Type: iOS Distribution
App ID: Lets Eat (BCKVL9BZPW.academy.cocoa.letseat)
Certificates: 1 Included
Cancel
Back
Continue

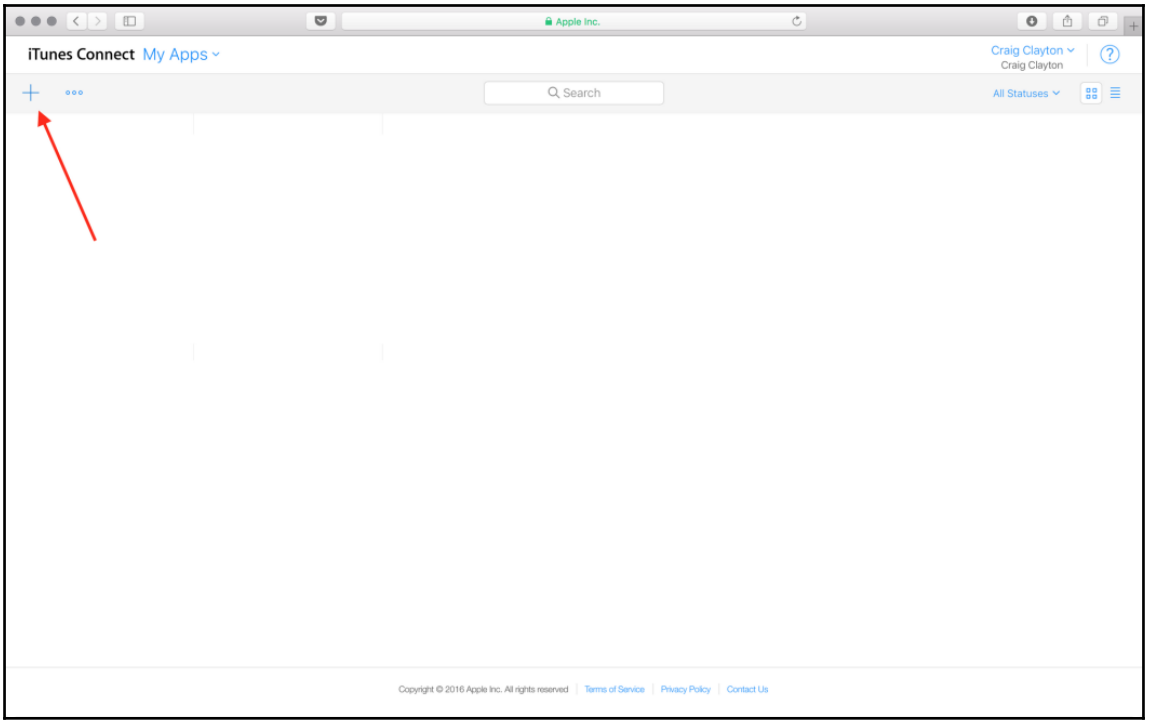

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
Your provisioning profile is ready.
Download and Install
Download and double click the following file to install your Provisioning Profile.
PROV
Name: Lets Eat Prod
Type: iOS Distribution
App ID: BCKVL9BZPW.academy.cocoa.letseat
Expires: Oct 11, 2017
Download
Documentation
For more information on using and managing your Provisioning Profile read:
App Distribution Guide
Add Another
Done

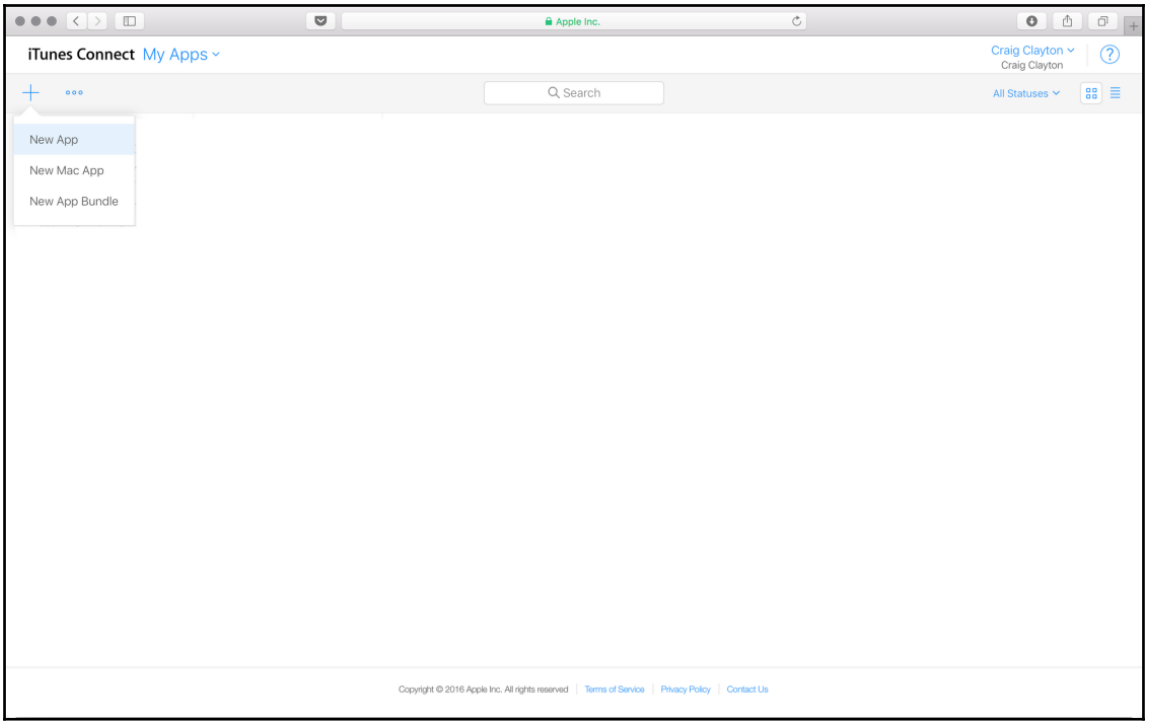

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
Name this profile and generate.
The name you provide will be used to identify the profile in the portal.
Profile Name: Lets Eat Dev
Type: iOS Development
App ID: Lets Eat (BCKVL9BZPW.academy.cocoa.letseat)
Certificates: 1 Included
Devices: 5 Included
Cancel
Back
Continue

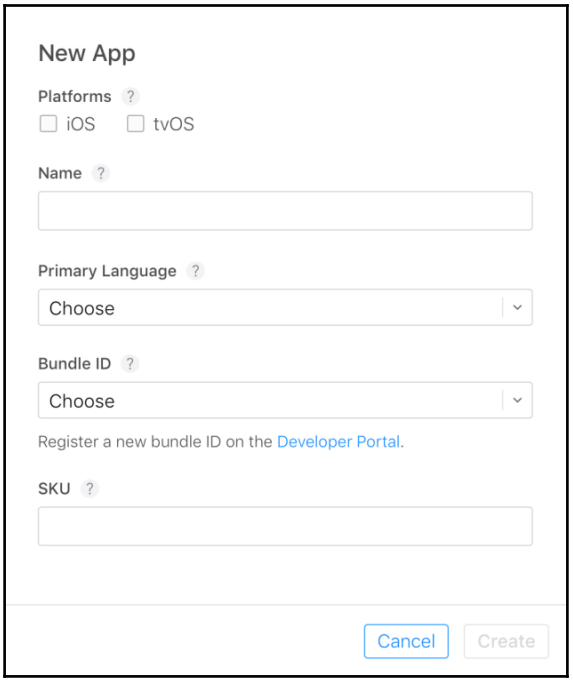

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
PROV
Select devices.
Select the devices you wish to include in this provisioning profile. To install an app signed with this profile on a device, the device must be included.
Select All
5 of 5 item(s) selected
iPad Pro
iPhone
Apple Watch
Craig's iPhone
Craig's AppleWatch
Cancel
Back
Continue

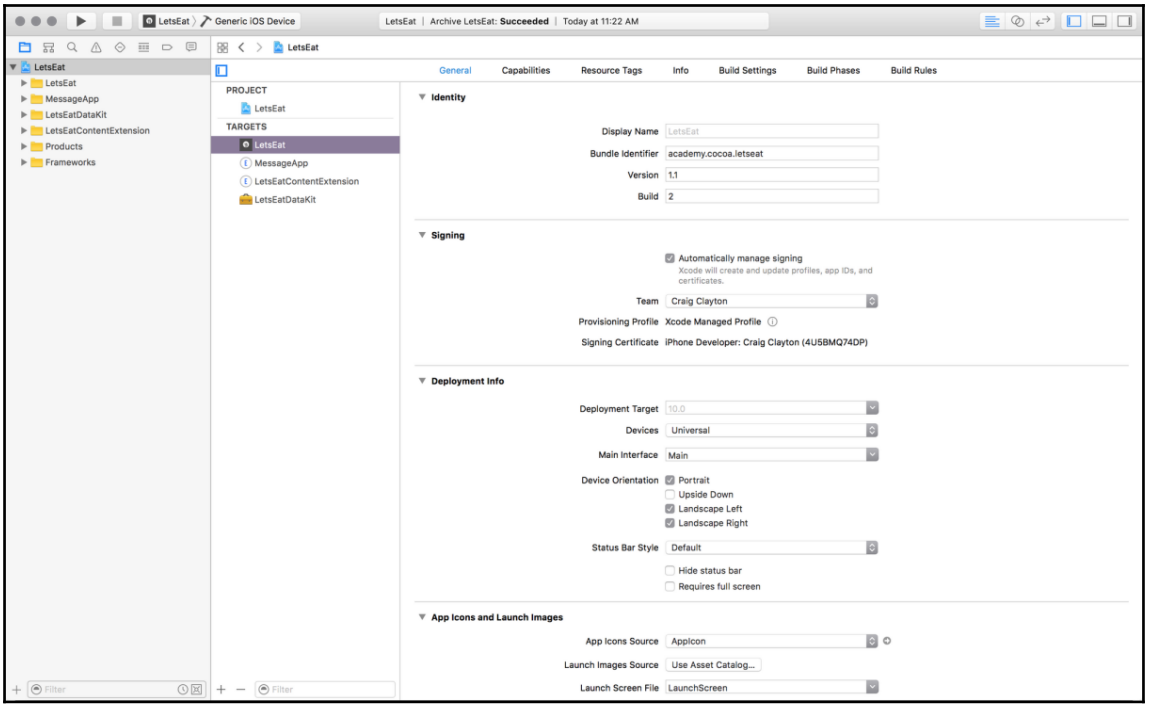

Apple Inc.
Certificates, Identifiers & Profiles
Craig Clayton
iOS, tvOS, watchOS
Certificates
All
Pending
Development
Production
APNs Auth Key
Identifiers
App IDs
Pass Type IDs
Website Push IDs
iCloud Containers
App Groups
Merchant IDs
Devices
All
Apple TV
Apple Watch
iPad
iPhone
iPod Touch
Provisioning Profiles
All
Development
Distribution
Add iOS Provisioning Profiles
Select Type
Configure
Generate
Download
Your provisioning profile is ready.
Download and Install
Download and double click the following file to install your Provisioning Profile.
Name: Lets Eat Dev
Type: iOS Development
App ID: BCKVL98ZPW.academy.cocoa.letseat
Expires: Oct 11, 2017
Download
Documentation
For more information on using and managing your Provisioning Profile read:
App Distribution Guide
Add Another
Done

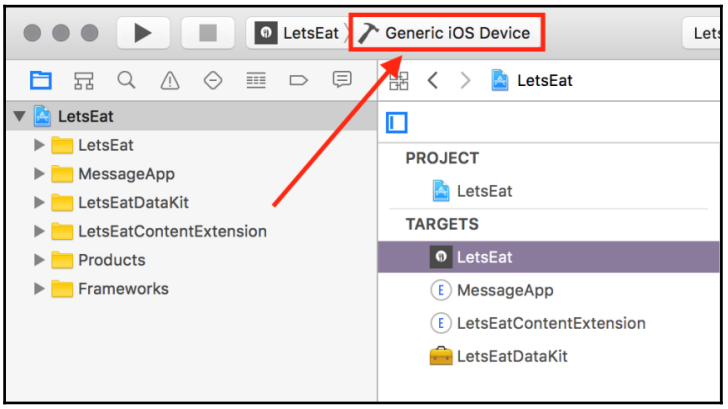

Apple Inc.
iTunes Connect
Craig Clayton
Craig Clayton
My Apps
App Analytics
Sales and Trends
Payments and Financial Reports
Users and Roles
Agreements, Tax, and Banking
Resources and Help
Terms of Service
Privacy Policy
Contact Us
Copyright © 2016 Apple Inc. All rights reserved

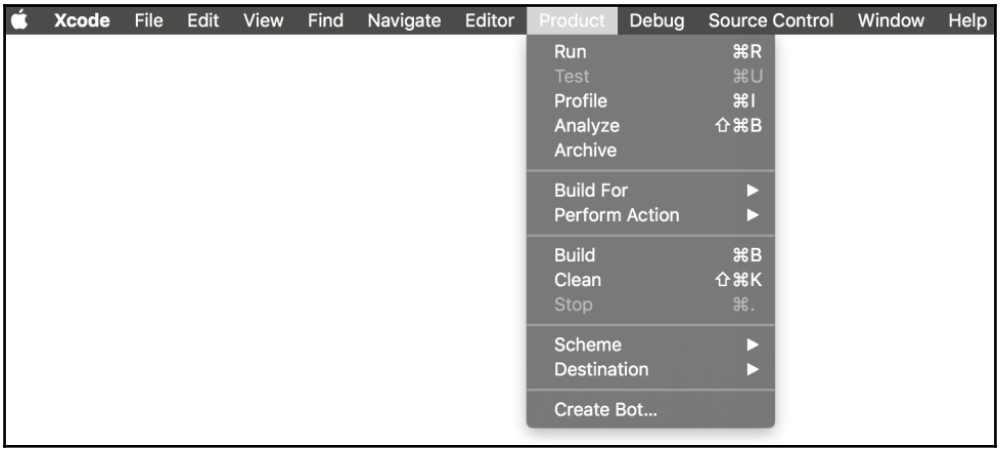

Apple Inc.
iTunes Connect My Apps
Craig Clayton
Craig Clayton
Search
All Statuses
Copyright © 2016 Apple Inc. All rights reserved
Terms of Service
Privacy Policy
Contact Us

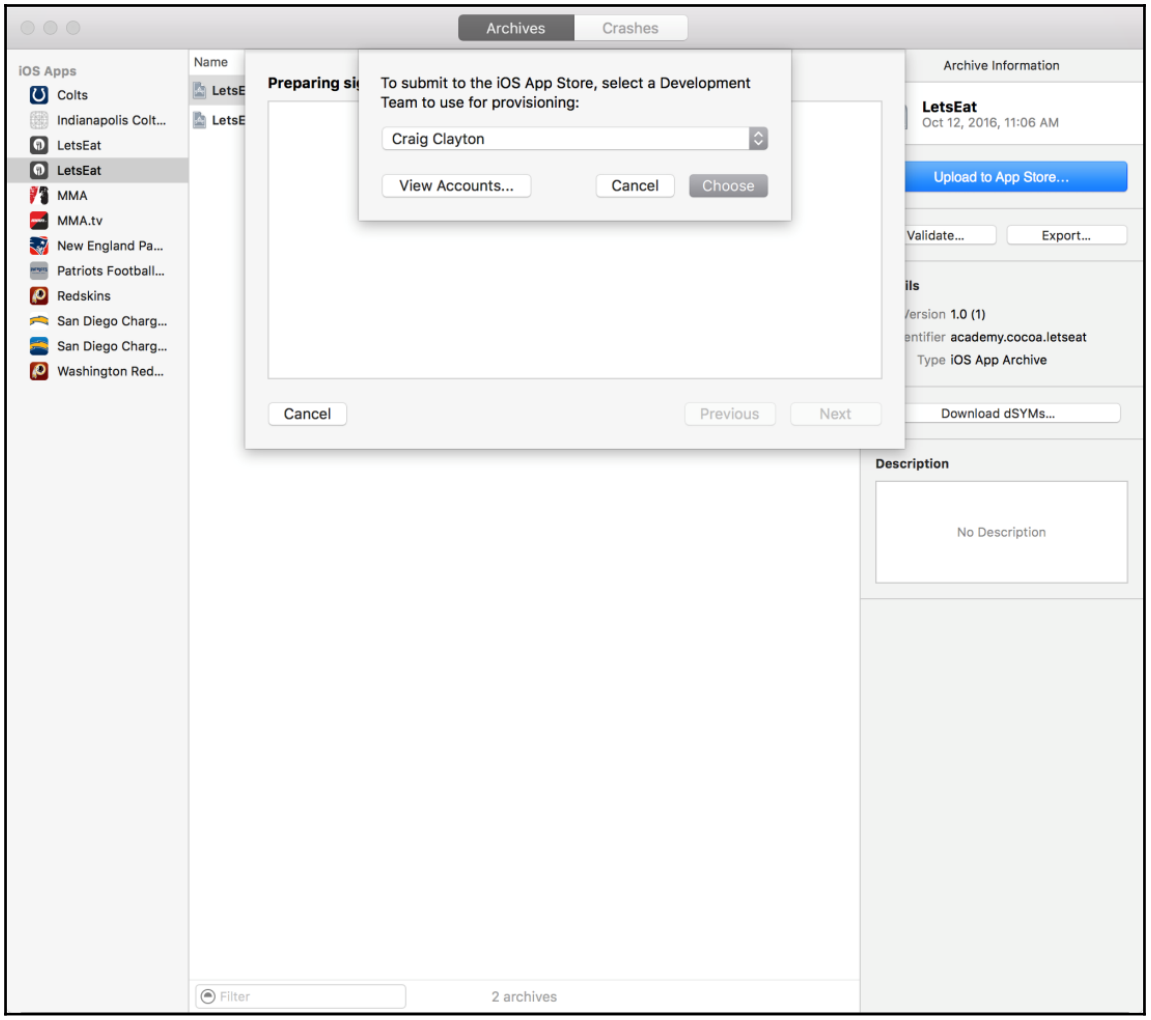

Apple Inc.
iTunes Connect My Apps
Craig Clayton
Craig Clayton
Search
All Statuses
New App
New Mac App
New App Bundle
Copyright © 2016 Apple Inc. All rights reserved
Terms of Service
Privacy Policy
Contact Us

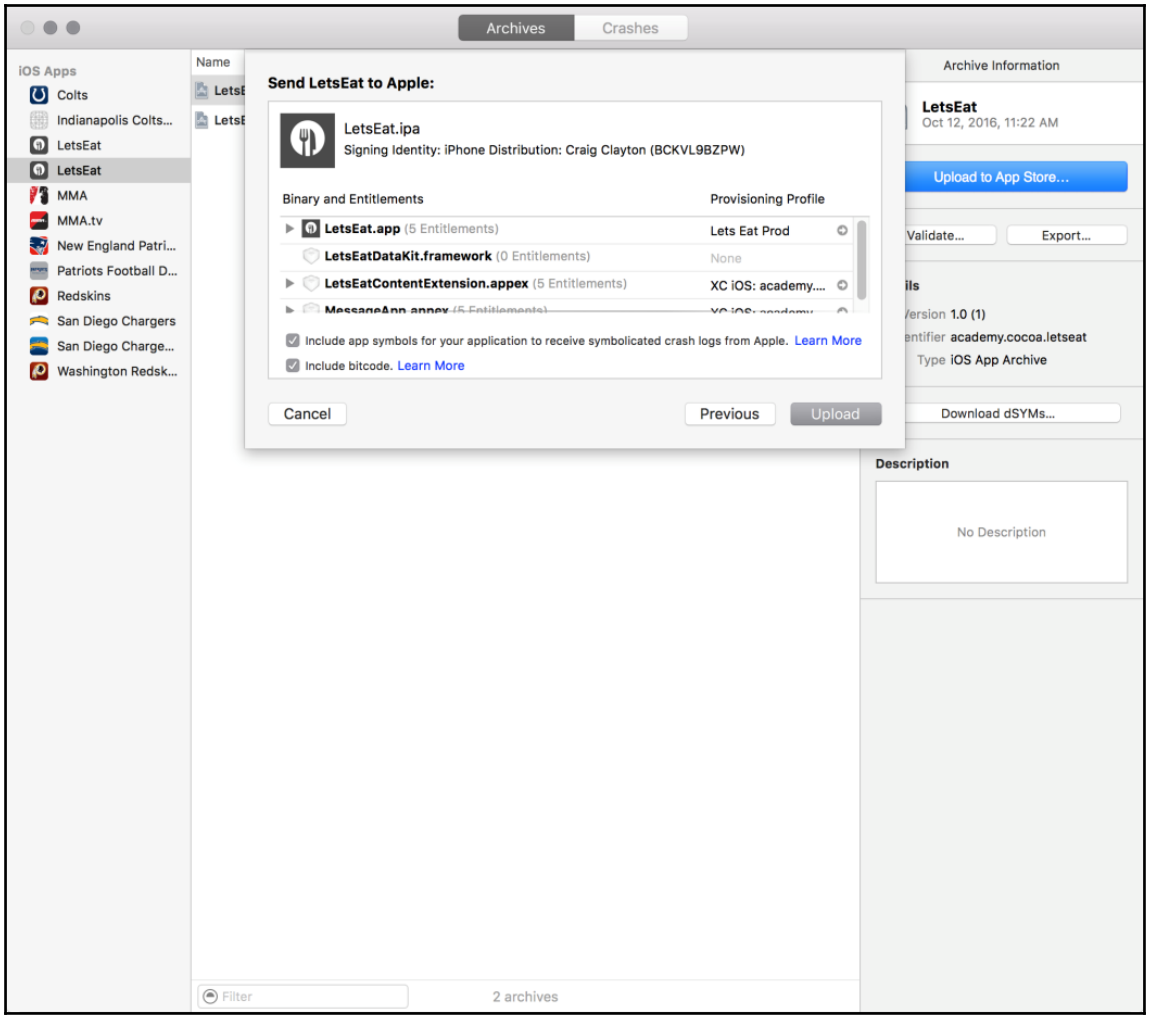

New App
Platforms
iOS
tvOS
Name
Primary Language
Choose
Bundle ID
Choose
Register a new bundle ID on the Developer Portal.
SKU
Cancel
Create

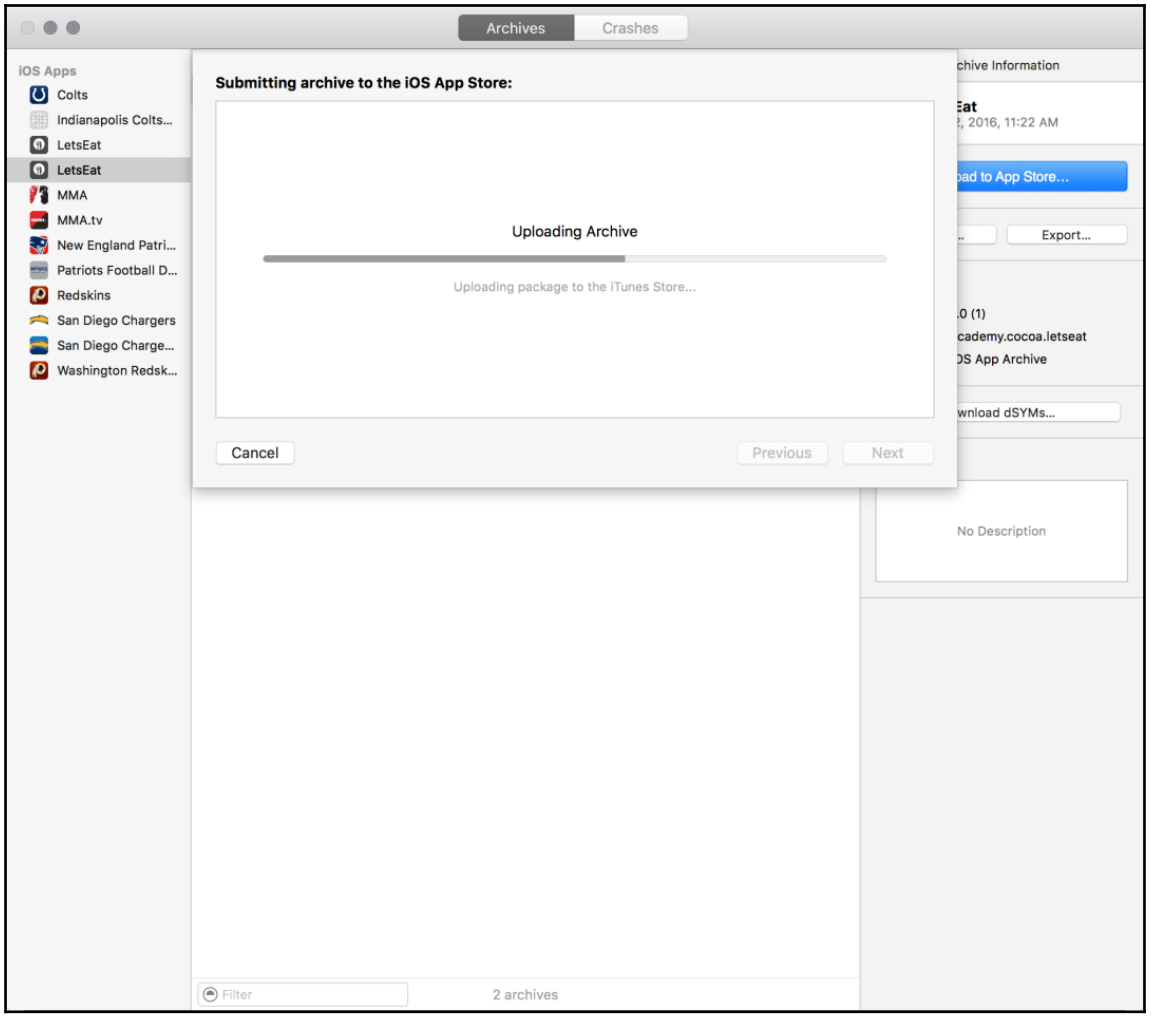

LetsEat
Generic iOS Device
LetsEat | Archive LetsEat: Succeeded | Today at 11:22 AM
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
TARGETS
Identity
Display Name
Bundle Identifier
academy.cocoa.letseat
Version
Build
Signing
Automatically manage signing
Team
Craig Clayton
Provisioning Profile
Xcode Managed Profile
Signing Certificate
iPhone Developer: Craig Clayton (4U5BMQ74DP)
Deployment Info
Deployment Target
Devices
Universal
Main Interface
Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
App Icons and Launch Images
App Icons Source
AppIcon
Launch Images Source
Use Asset Catalog...
Launch Screen File
LaunchScreen

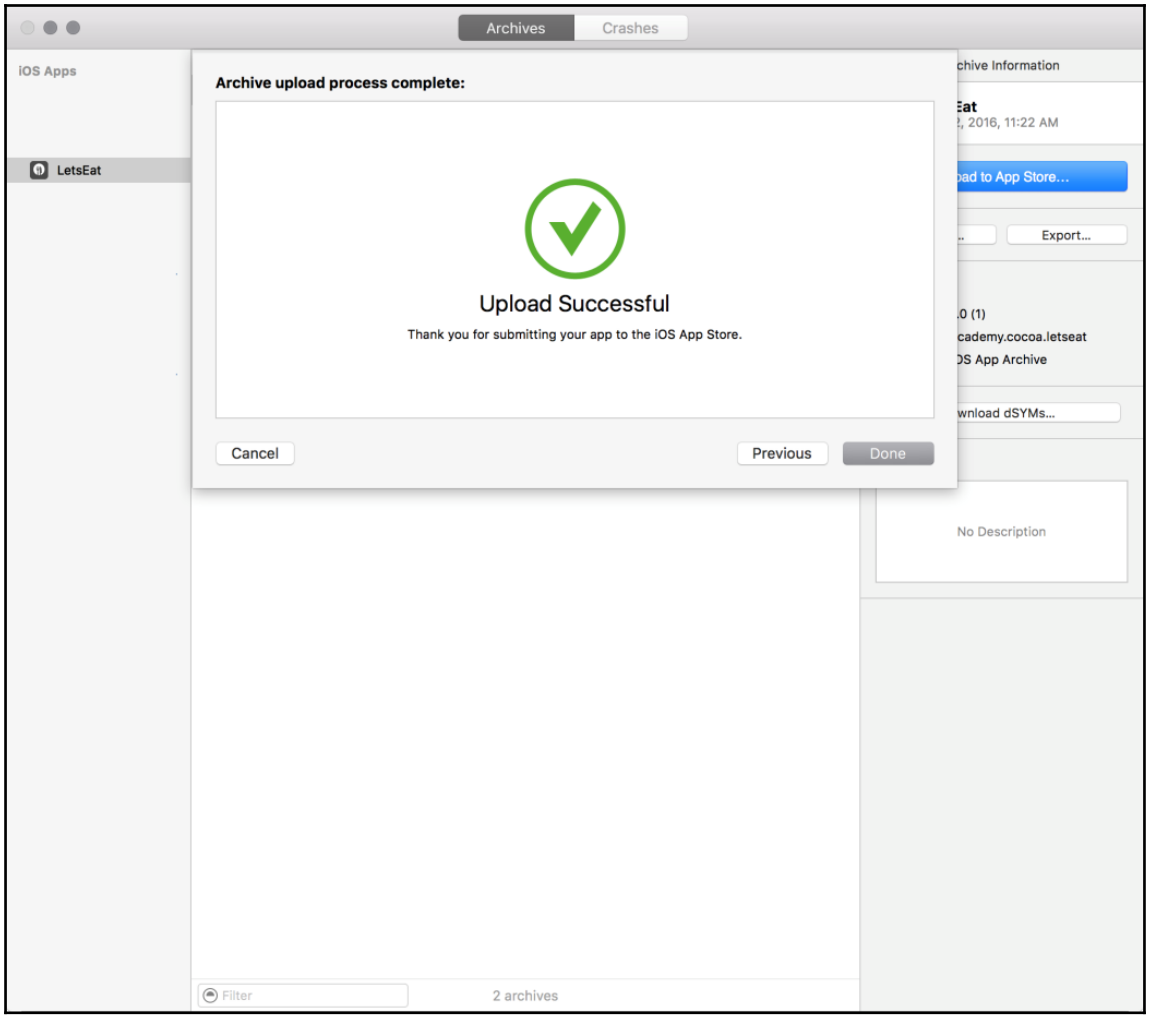

LetsEat
Generic iOS Device
LetsEat
MessageApp
LetsEatDataKit
LetsEatContentExtension
Products
Frameworks
PROJECT
TARGETS
LetsEat
MessageApp
LetsEatContentExtension
LetsEatDataKit

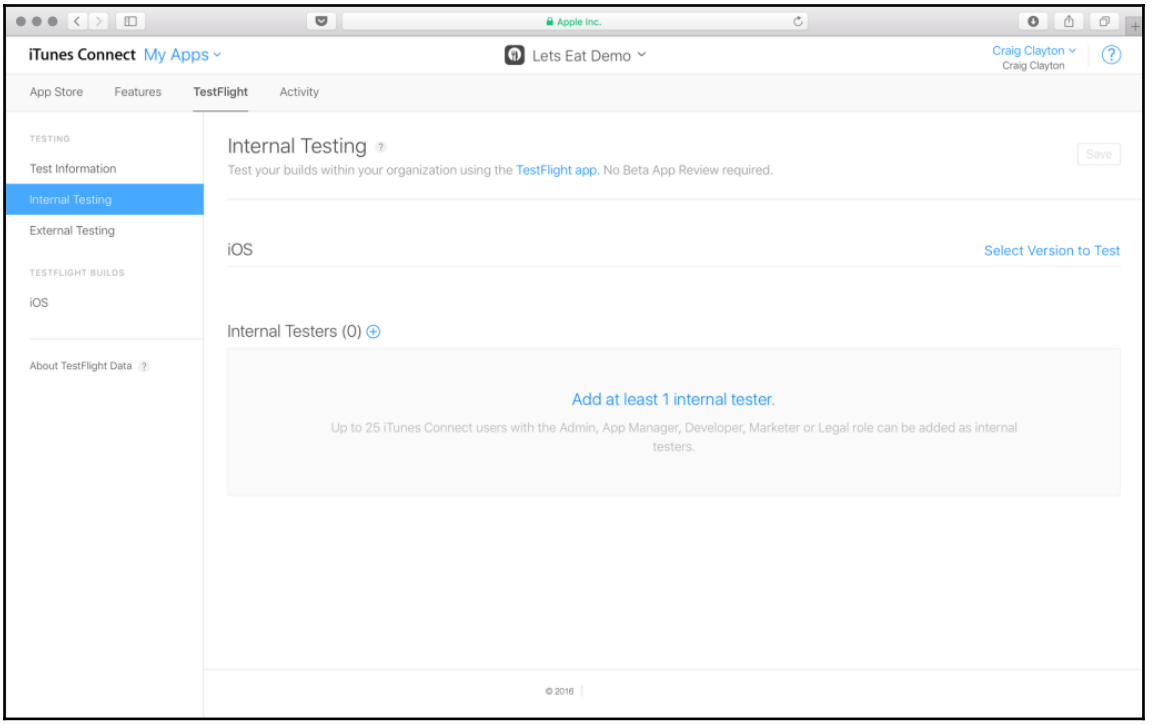

Xcode
File
Edit
View
Find
Navigate
Editor
Product
Debug
Source Control
Window
Help
Run ⌘R
Test ⌘U
Profile ⌘I
Analyze ⇧⌘B
Archive
Build For ▶
Perform Action ▶
Build ⌘B
Clean ⇧⌘K
Stop ⌘.
Scheme ▶
Destination ▶
Create Bot...

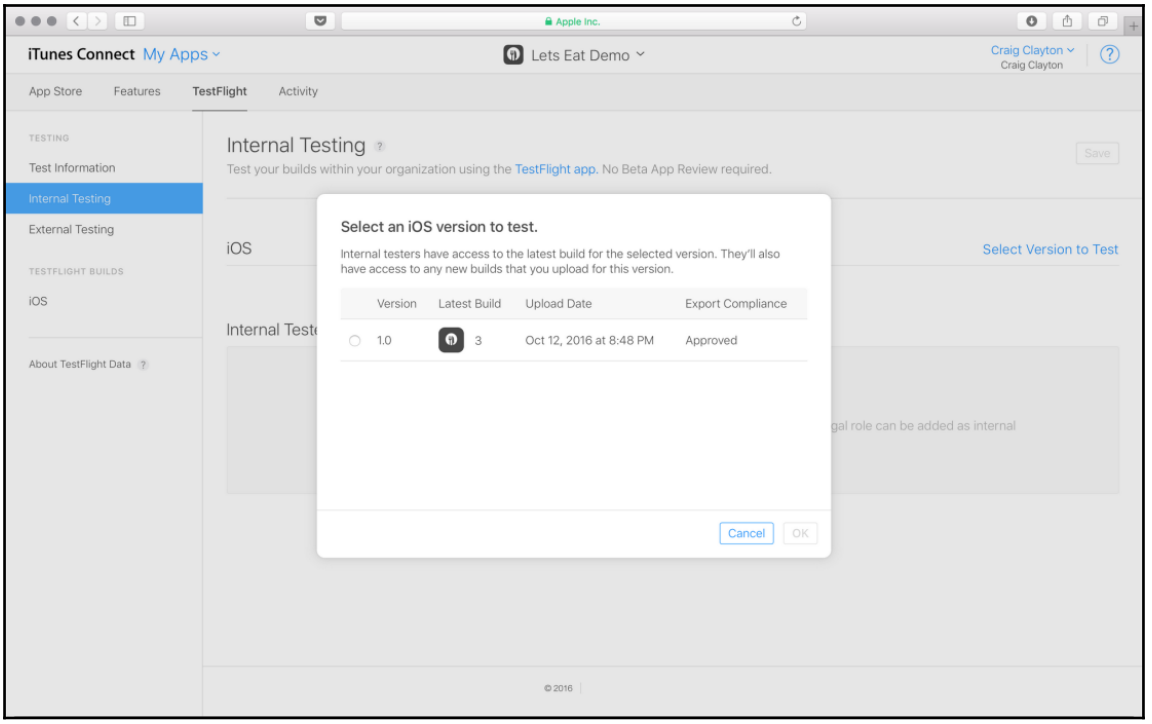

Archives
Crashes
iOS Apps
Colts
Indianapolis Colt...
LetsEat
LetsEat
MMA
MMA.tv
New England Pa...
Patriots Football...
Redskins
San Diego Charg...
San Diego Charg...
Washington Red...
Name
LetsE
LetsE
Preparing si
To submit to the iOS App Store, select a Development Team to use for provisioning:
Craig Clayton
View Accounts...
Cancel
Choose
Cancel
Previous
Next
Filter
2 archives
Archive Information
LetsEat
Oct 12, 2016, 11:06 AM
Upload to App Store...
Validate...
Export...
ils
Version 1.0 (1)
entifier academy.cocoa.letseat
Type iOS App Archive
Download dSYMs...
Description
No Description

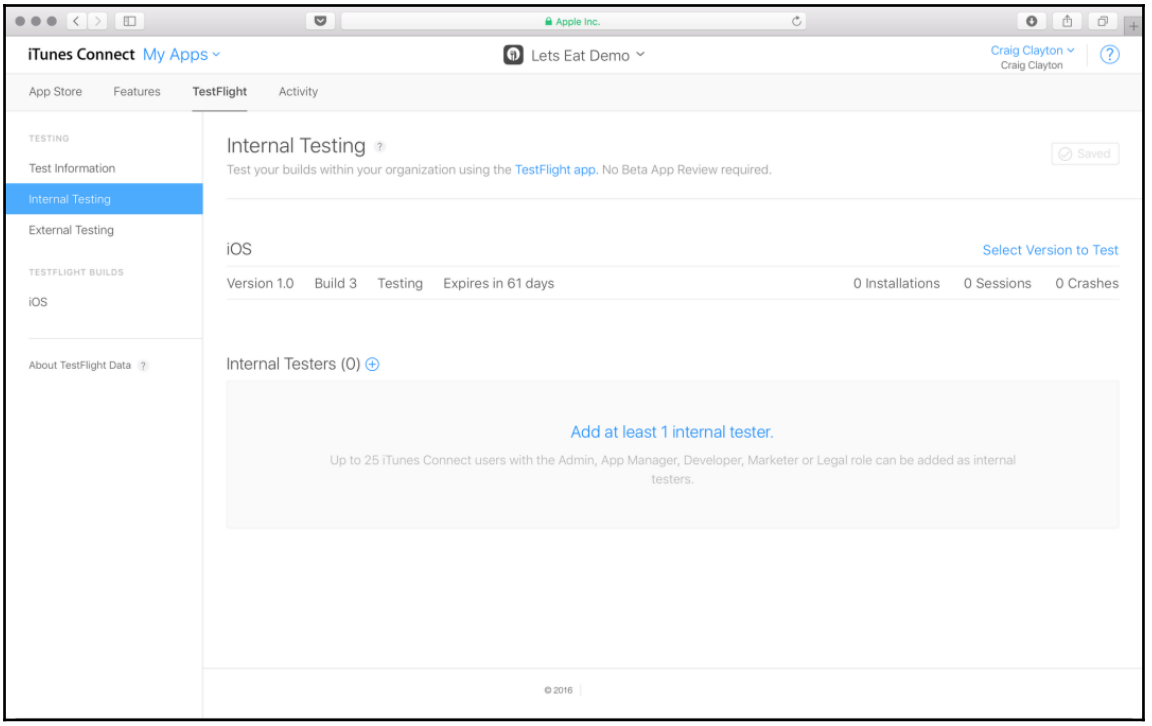

Archives
Crashes
iOS Apps
Colts
Indianapolis Colts...
LetsEat
LetsEat
MMA
MMA.tv
New England Patri...
Patriots Football D...
Redskins
San Diego Chargers
San Diego Charge...
Washington Redsk...
Name
Send LetsEat to Apple:
LetsEat.ipa
Signing Identity: iPhone Distribution: Craig Clayton (BCKVL9BZPW)
Binary and Entitlements
Provisioning Profile
LetsEat.app (5 Entitlements)
Lets Eat Prod
LetsEatDataKit.framework (0 Entitlements)
None
LetsEatContentExtension.appex (5 Entitlements)
XC iOS: academy....
Include app symbols for your application to receive symbolicated crash logs from Apple. Learn More
Include bitcode. Learn More
Cancel
Previous
Upload
Archive Information
LetsEat
Oct 12, 2016, 11:22 AM
Upload to App Store...
Validate...
Export...
Version 1.0 (1)
Type iOS App Archive
Download dSYMs...
Description
No Description
Filter
2 archives

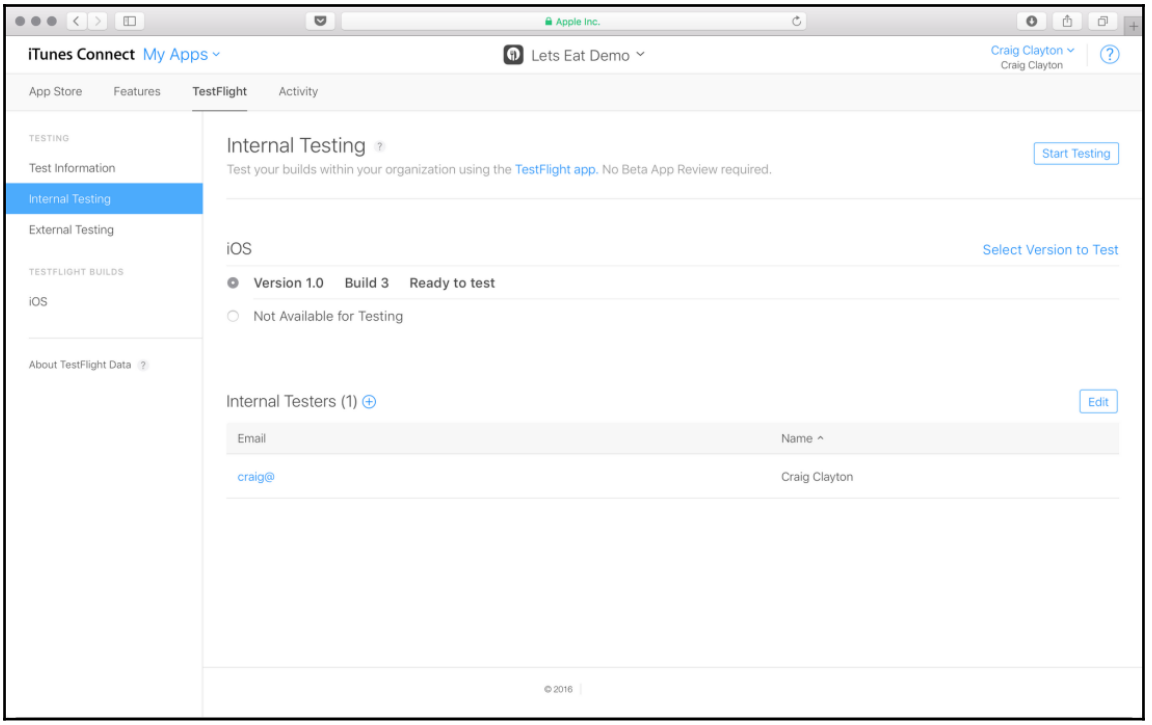

Archives
Crashes
iOS Apps
Colts
Indianapolis Colts...
LetsEat
LetsEat
MMA
MMA.tv
New England Patri...
Patriots Football D...
Redskins
San Diego Chargers
San Diego Charge...
Washington Redsk...
Submitting archive to the iOS App Store:
Uploading Archive
Uploading package to the iTunes Store...
Cancel
Previous
Next
chive Information
Eat
2016, 11:22 AM
ad to App Store...
Export...
.0 (1)
cademy.cocoa.letseat
OS App Archive
wnload dSYMs...
No Description
Filter
2 archives

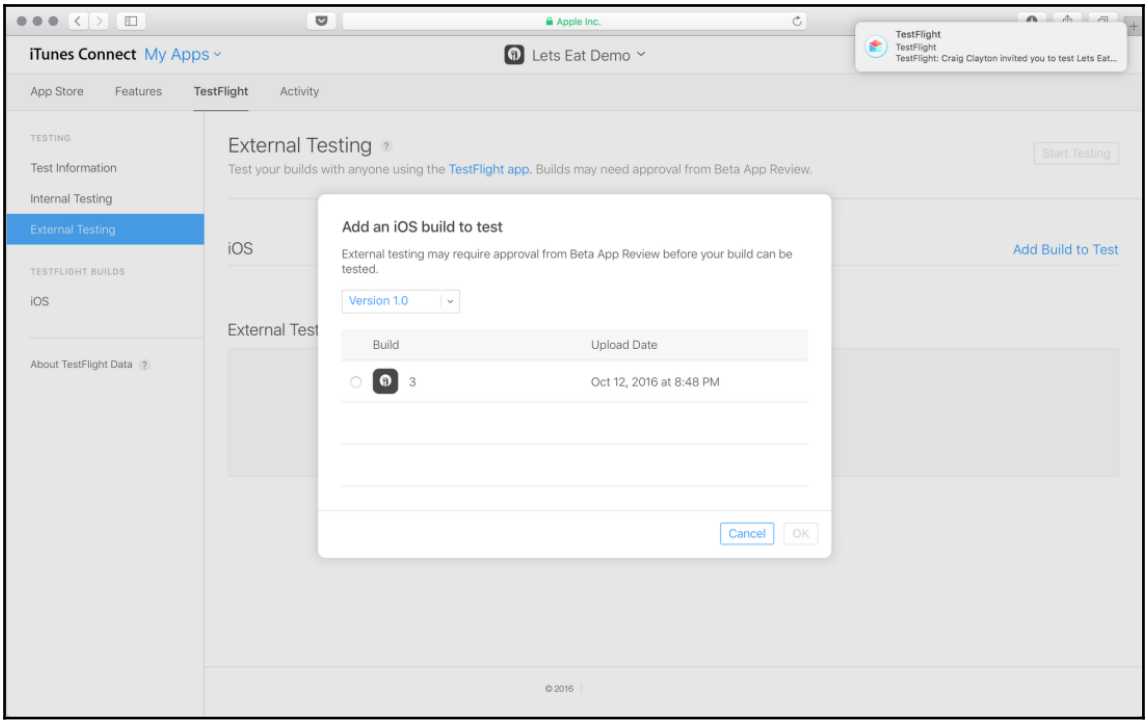

Archives
Crashes
iOS Apps
LetsEat
Archive upload process complete:
Upload Successful
Thank you for submitting your app to the iOS App Store.
Cancel
Previous
Done
chive Information
Export...
No Description
Filter
2 archives

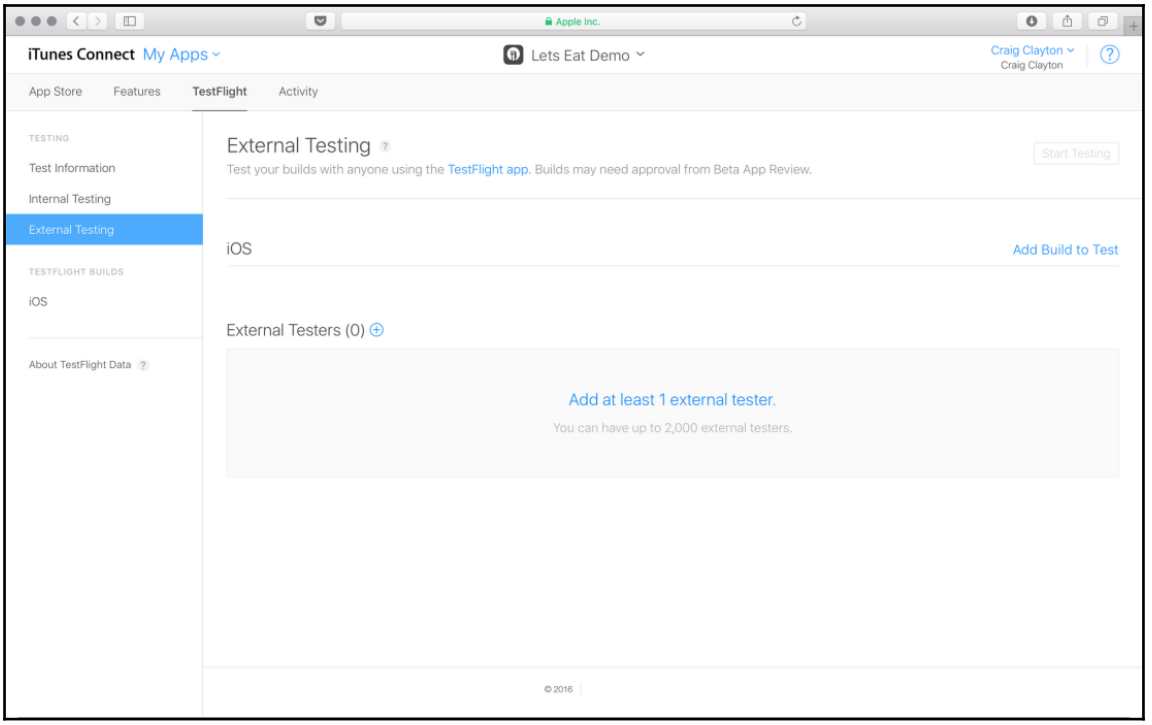

Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
Craig Clayton
Craig Clayton
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
Internal Testing
Test your builds within your organization using the TestFlight app. No Beta App Review required.
Save
iOS
Select Version to Test
Internal Testers (0)
Add at least 1 internal tester.
Up to 25 iTunes Connect users with the Admin, App Manager, Developer, Marketer or Legal role can be added as internal testers.
© 2016

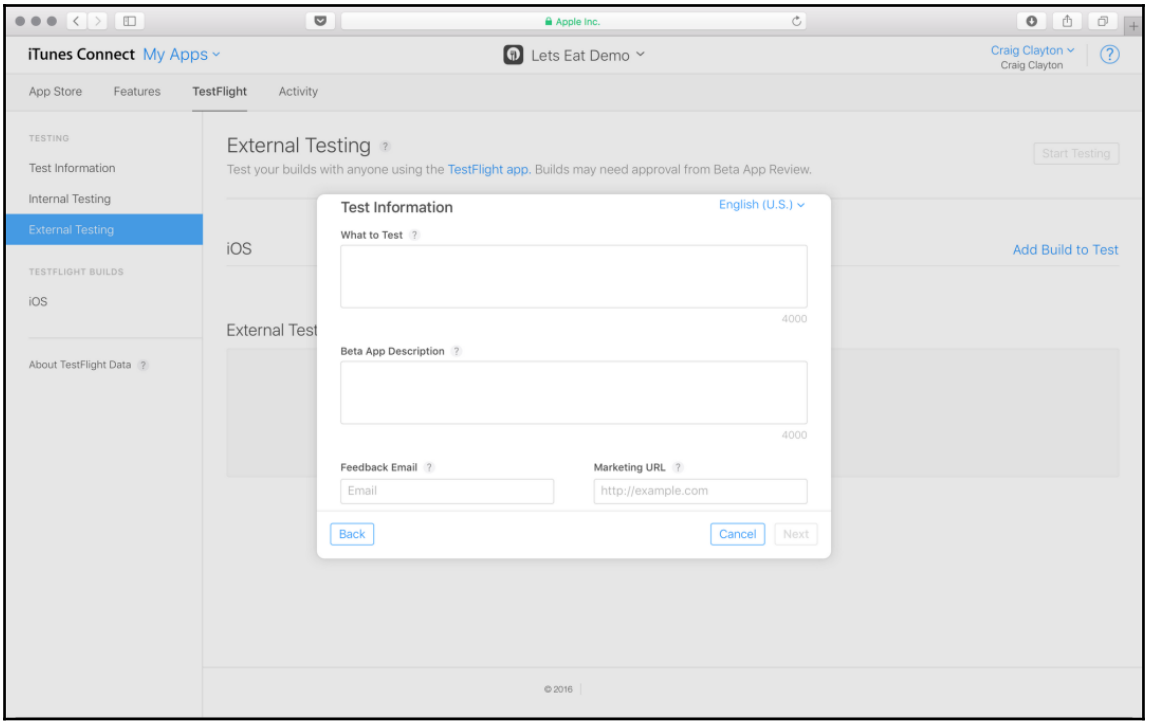

Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
Craig Clayton
Craig Clayton
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
Internal Testing
Test your builds within your organization using the TestFlight app. No Beta App Review required.
Save
iOS
Select Version to Test
Internal Test
gal role can be added as internal
Select an iOS version to test.
Internal testers have access to the latest build for the selected version. They'll also have access to any new builds that you upload for this version.
Version
Latest Build
Upload Date
Export Compliance
1.0
3
Oct 12, 2016 at 8:48 PM
Approved
Cancel
OK
© 2016


Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
Craig Clayton
Craig Clayton
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
Internal Testing
Test your builds within your organization using the TestFlight app. No Beta App Review required.
Saved
iOS
Select Version to Test
Version 1.0 Build 3 Testing Expires in 61 days
0 Installations 0 Sessions 0 Crashes
Internal Testers (0)
Add at least 1 internal tester.
Up to 25 iTunes Connect users with the Admin, App Manager, Developer, Marketer or Legal role can be added as internal testers.
© 2016


Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
Craig Clayton
Craig Clayton
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
Internal Testing
Test your builds within your organization using the TestFlight app. No Beta App Review required.
Start Testing
iOS
Select Version to Test
Version 1.0 Build 3 Ready to test
Not Available for Testing
Internal Testers (1)
Edit
Email
Name
craig@
Craig Clayton
© 2016


Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
TestFlight
TestFlight
TestFlight: Craig Clayton invited you to test Lets Eat...
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
External Testing
Test your builds with anyone using the TestFlight app. Builds may need approval from Beta App Review.
Start Testing
iOS
Add Build to Test
External Test
Add an iOS build to test
External testing may require approval from Beta App Review before your build can be tested.
Version 1.0
Build
Upload Date
3
Oct 12, 2016 at 8:48 PM
Cancel
OK
© 2016


Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
Craig Clayton
Craig Clayton
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
External Testing
Test your builds with anyone using the TestFlight app. Builds may need approval from Beta App Review.
Start Testing
iOS
Add Build to Test
External Testers (0)
Add at least 1 external tester.
You can have up to 2,000 external testers.
© 2016


Apple Inc.
iTunes Connect My Apps
Lets Eat Demo
Craig Clayton
Craig Clayton
App Store
Features
TestFlight
Activity
TESTING
Test Information
Internal Testing
External Testing
TESTFLIGHT BUILDS
iOS
About TestFlight Data
External Testing
Test your builds with anyone using the TestFlight app. Builds may need approval from Beta App Review.
Start Testing
iOS
Add Build to Test
External Test
Test Information
English (U.S.)
What to Test
4000
Beta App Description
4000
Feedback Email
Email
Marketing URL
http://example.com
Back
Cancel
Next
© 2016