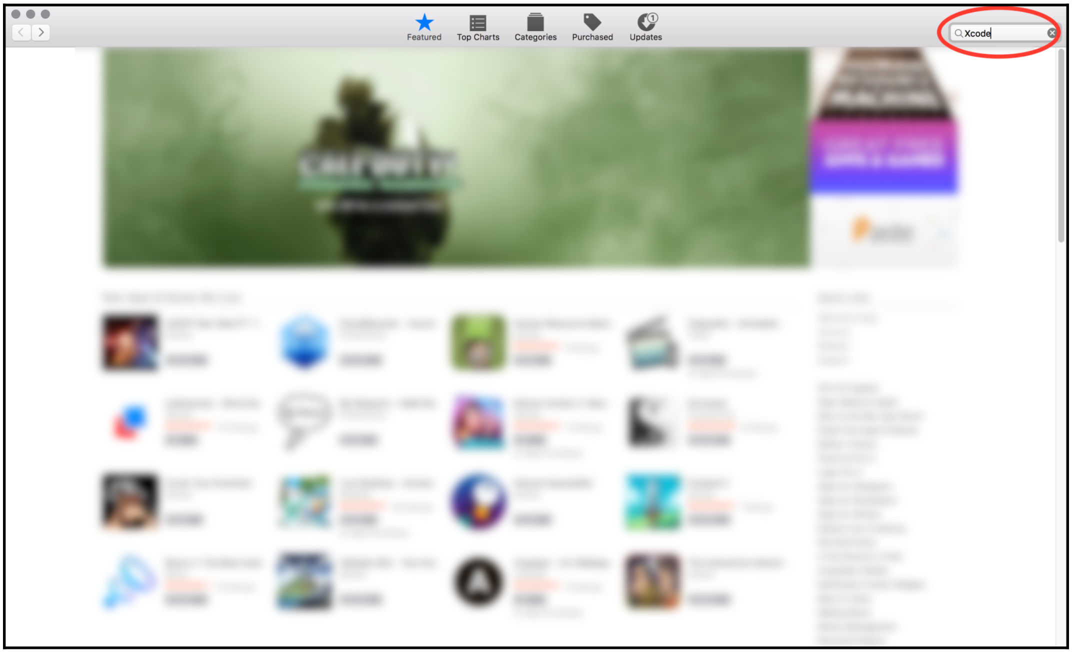
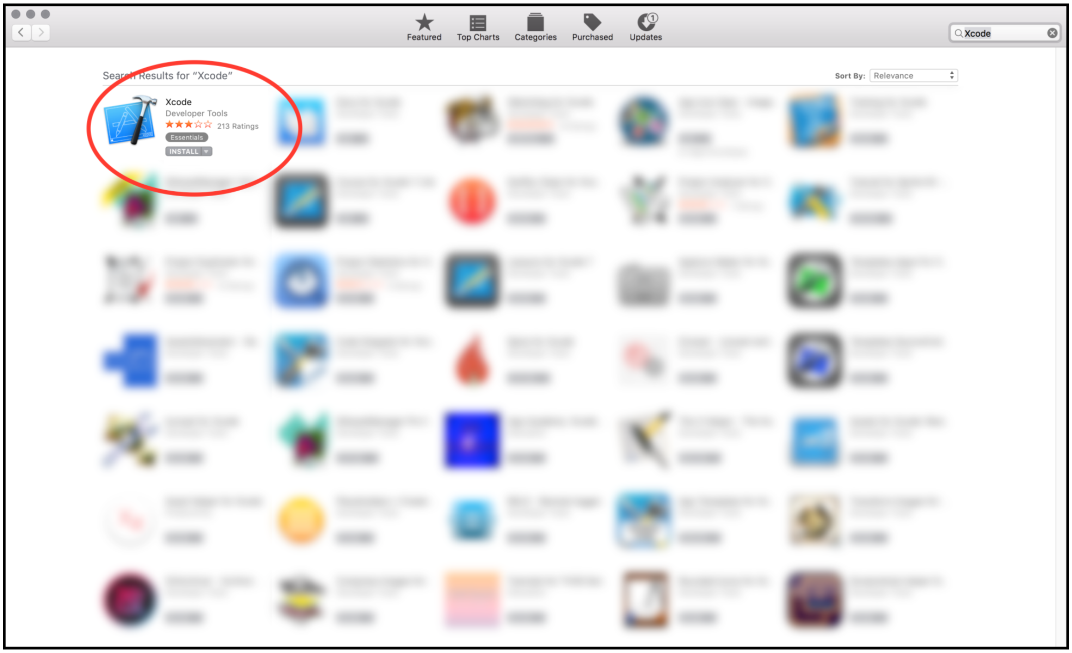
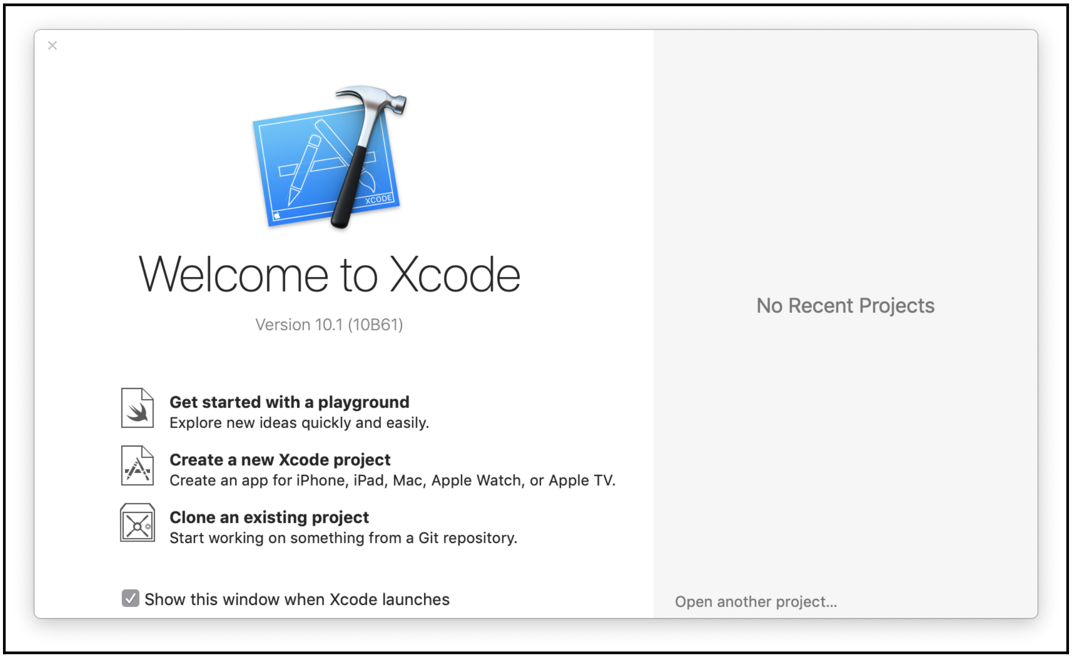
Table of Contents

Chapter 1: Getting Familiar with Xcode







Welcome to Xcode

Version 10.1 (10B61)

Get started with a playground
Explore new ideas quickly and easily.

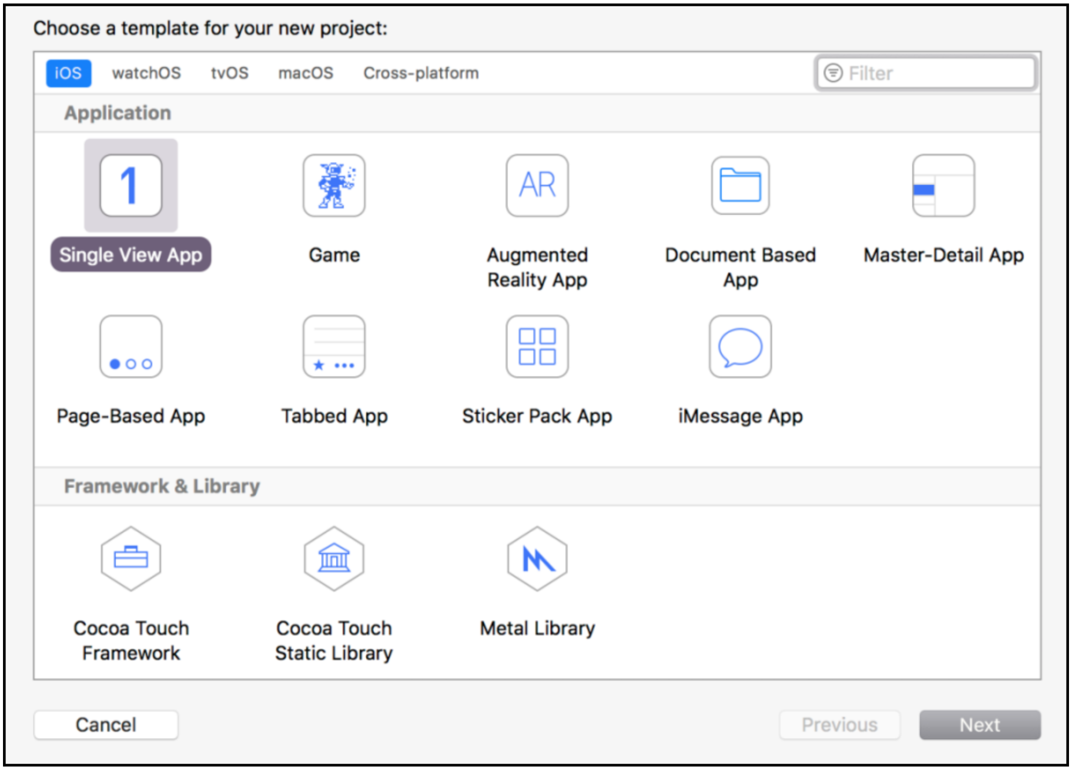
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.

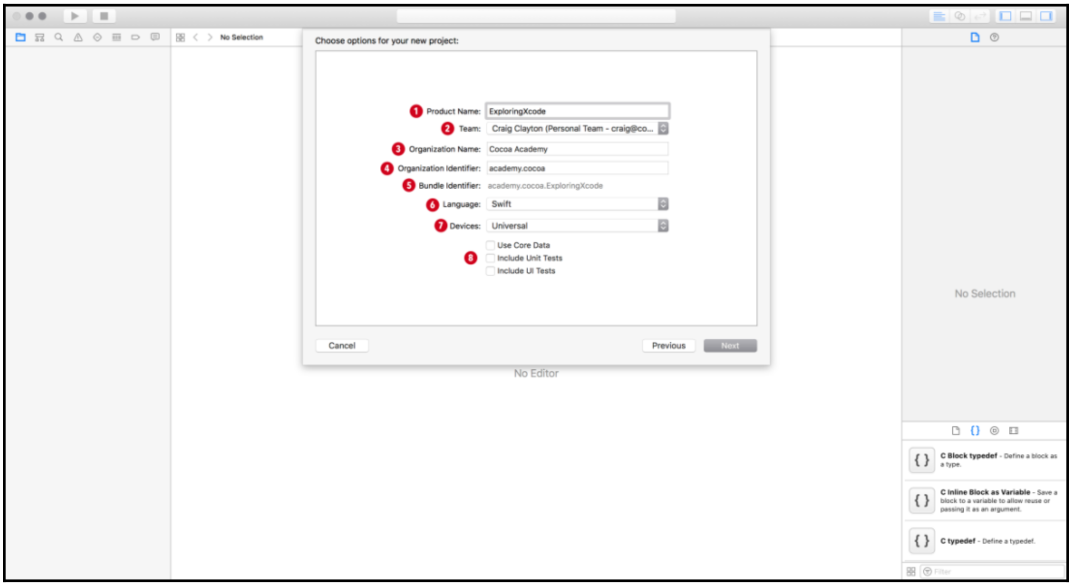
Clone an existing project
Start working on something from a Git repository.

☑ Show this window when Xcode launches

No Recent Projects

Open another project...





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

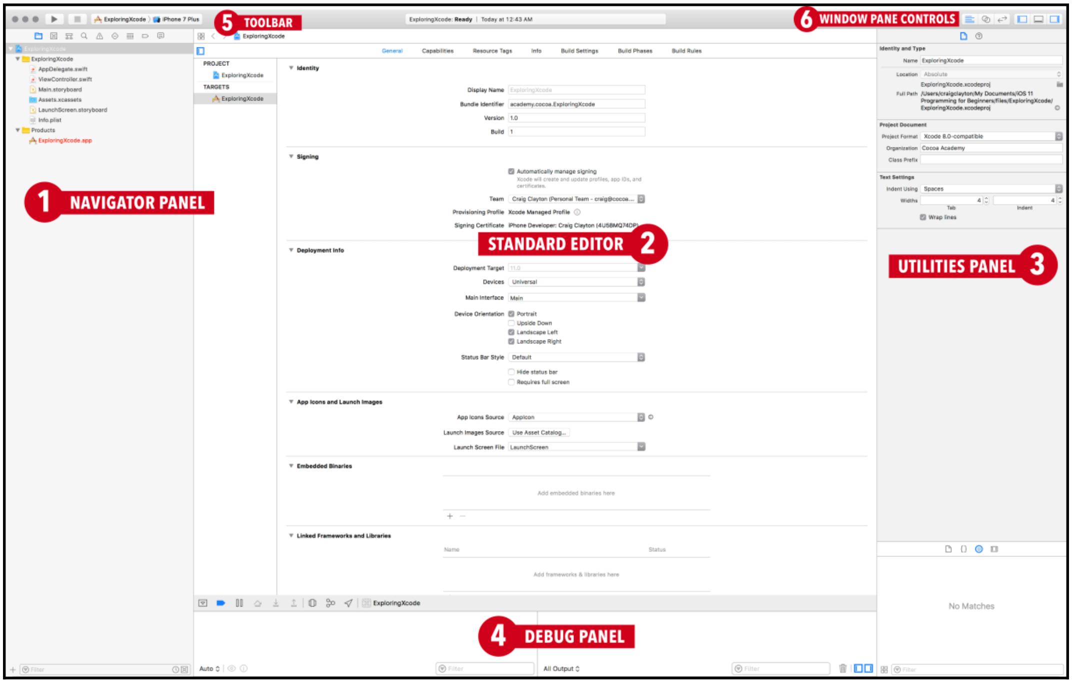
7 Devices: Universal

8
□ Use Core Data
□ Include Unit Tests
□ Include UI Tests

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.



ExploringXcode: **Ready** | Today at 12:43 AM

Identity

Display Name: ExploringXcode
Bundle Identifier: academy.cocoa.ExploringXcode
Version: 1.0
Build: 1

Signing

☑ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team: Craig Clayton (Personal Team - craig@cocoa....
Provisioning Profile: Xcode Managed Profile
Signing Certificate: iPhone Developer: Craig Clayton (4U58MQ74DP)

Deployment Info

Deployment Target: 11.0
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

Add embedded binaries here

Linked Frameworks and Libraries

Name Status
Add frameworks & libraries here

Identity and Type

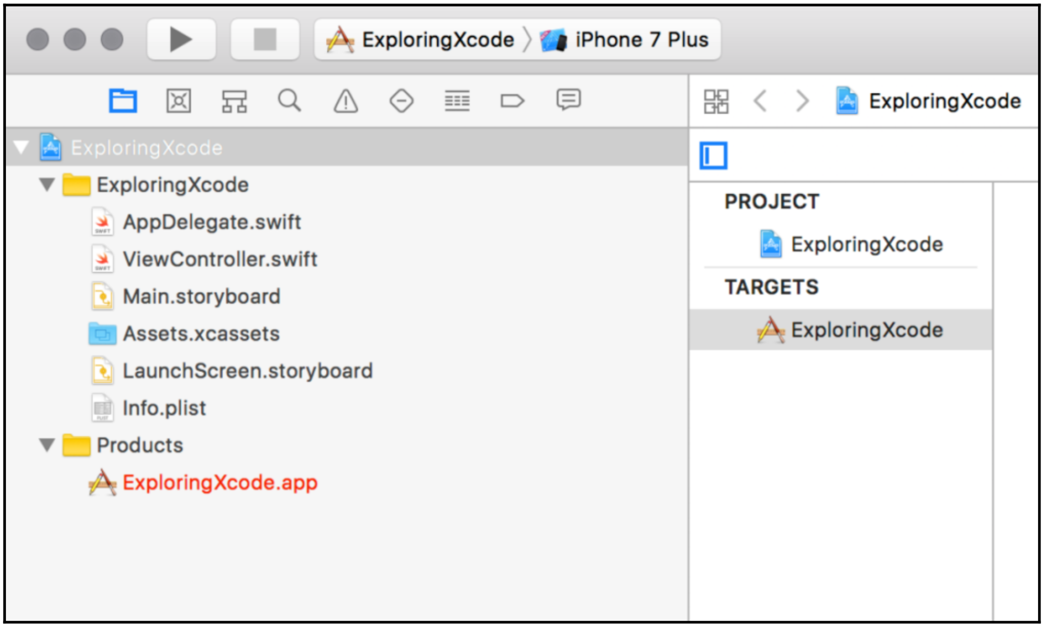
Name: ExploringXcode
Location: Absolute
ExploringXcode.xcodeproj
Full Path: /Users/craigclayton/My Documents/iOS 11 Programming for Beginners/files/ExploringXcode/ExploringXcode.xcodeproj

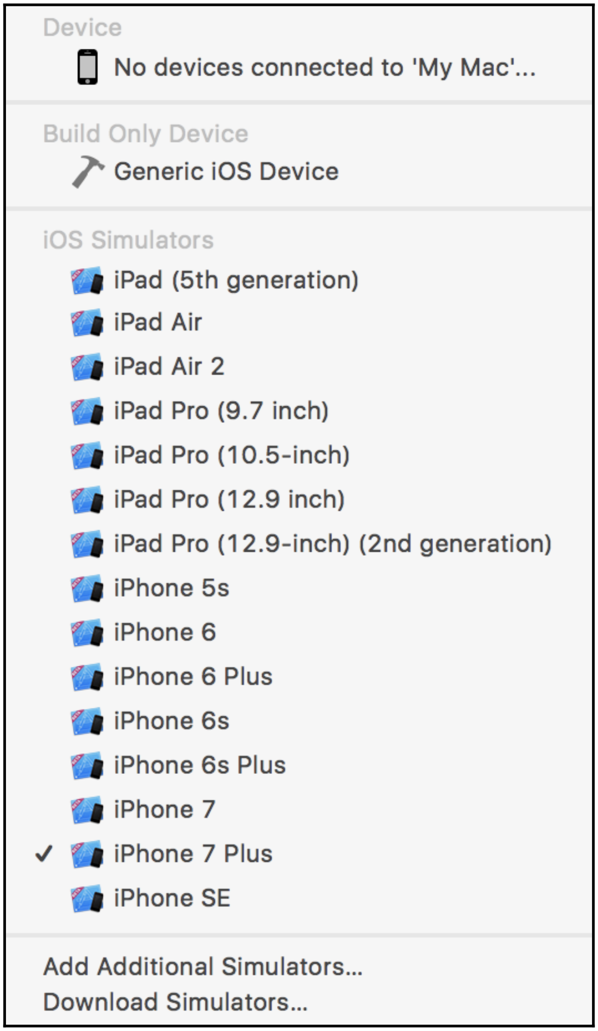
Project Document

Project Format: Xcode 8.0-compatible
Organization: Cocoa Academy
Class Prefix:

Text Settings

Indent Using: Spaces
Widths: 4 (Tab) 4 (Indent)
☑ Wrap lines





Device

No devices connected to 'My Mac'...

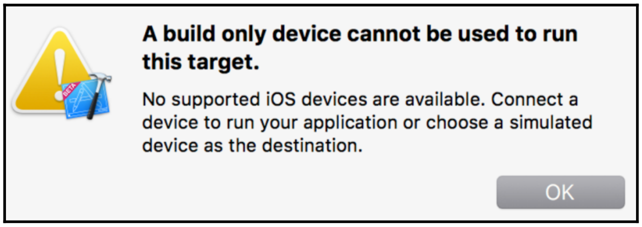
Build Only Device

Generic iOS Device

iOS Simulators

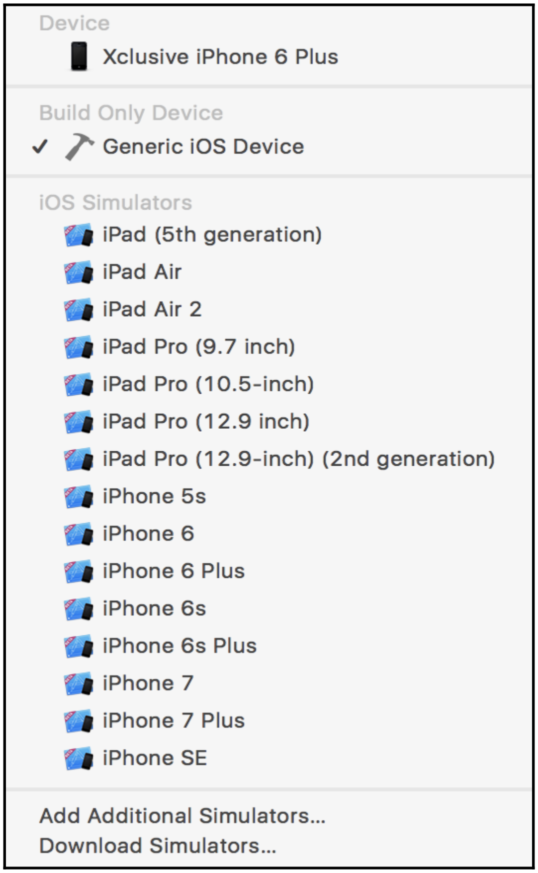
- iPad (5th generation)
- iPad Air
- iPad Air 2
- iPad Pro (9.7 inch)
- iPad Pro (10.5-inch)
- iPad Pro (12.9 inch)
- iPad Pro (12.9-inch) (2nd generation)
- iPhone 5s
- iPhone 6
- iPhone 6 Plus
- iPhone 6s
- iPhone 6s Plus
- iPhone 7
- ✓ iPhone 7 Plus
- iPhone SE

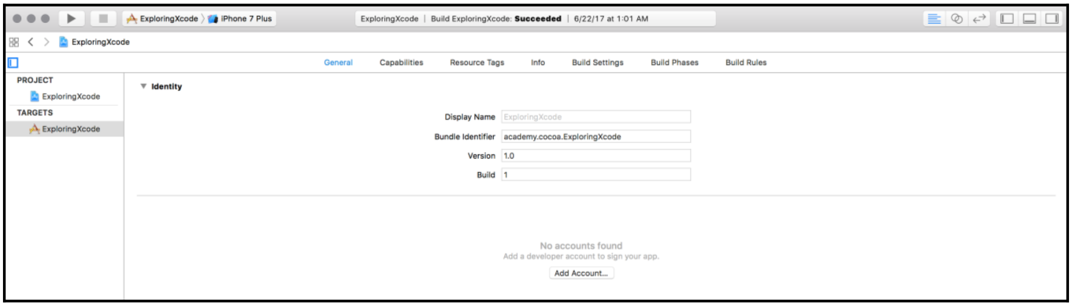
Add Additional Simulators...
Download Simulators...



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.





ExploringXcode | Build ExploringXcode: **Succeeded** | 6/22/17 at 1:01 AM

Display Name ExploringXcode

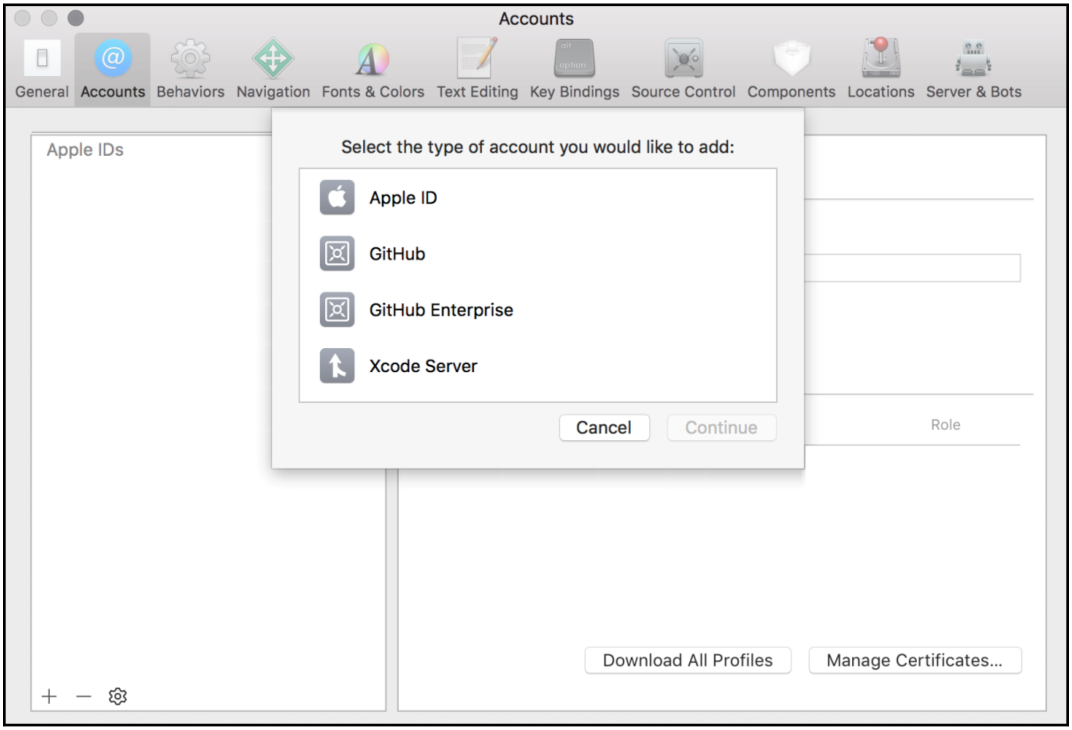
Bundle Identifier academy.cocoa.ExploringXcode

Version 1.0

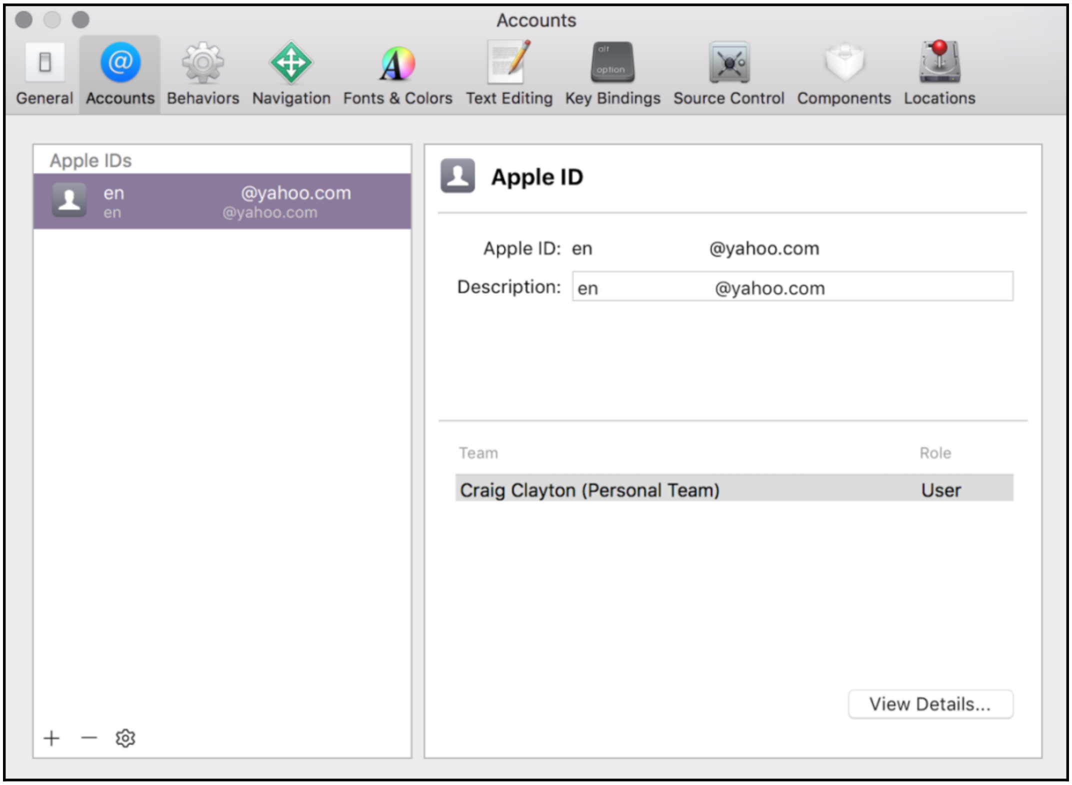
Build 1

No accounts found

Add a developer account to sign your app.



Select the type of account you would like to add:



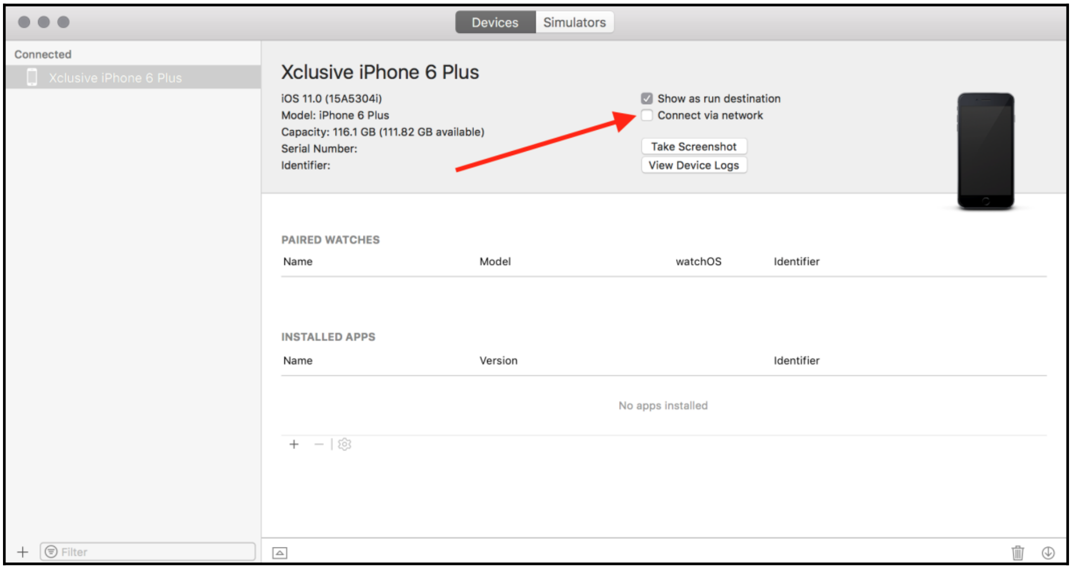
en @yahoo.com
en @yahoo.com

Apple ID

Apple ID: en @yahoo.com

Description: en @yahoo.com

Team	Role
Craig Clayton (Personal Team)	User



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

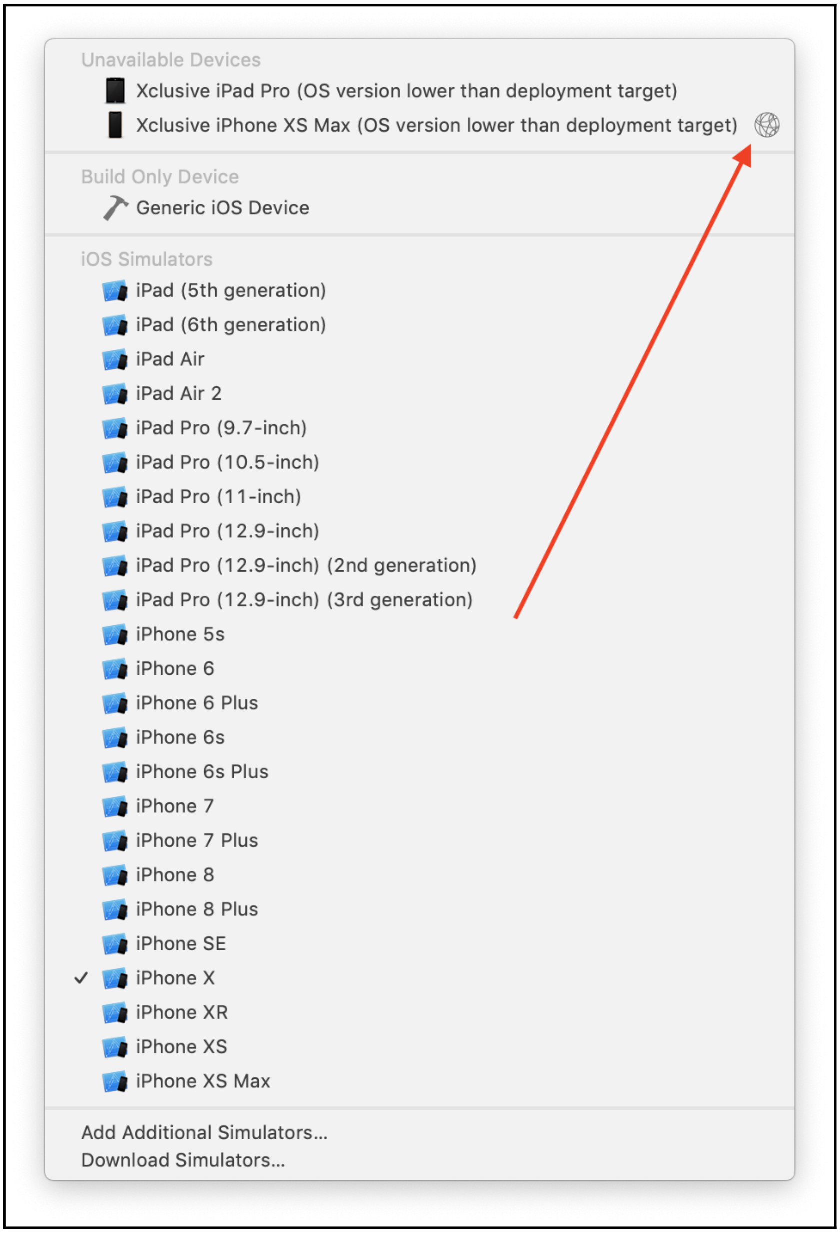
PAIRED WATCHES

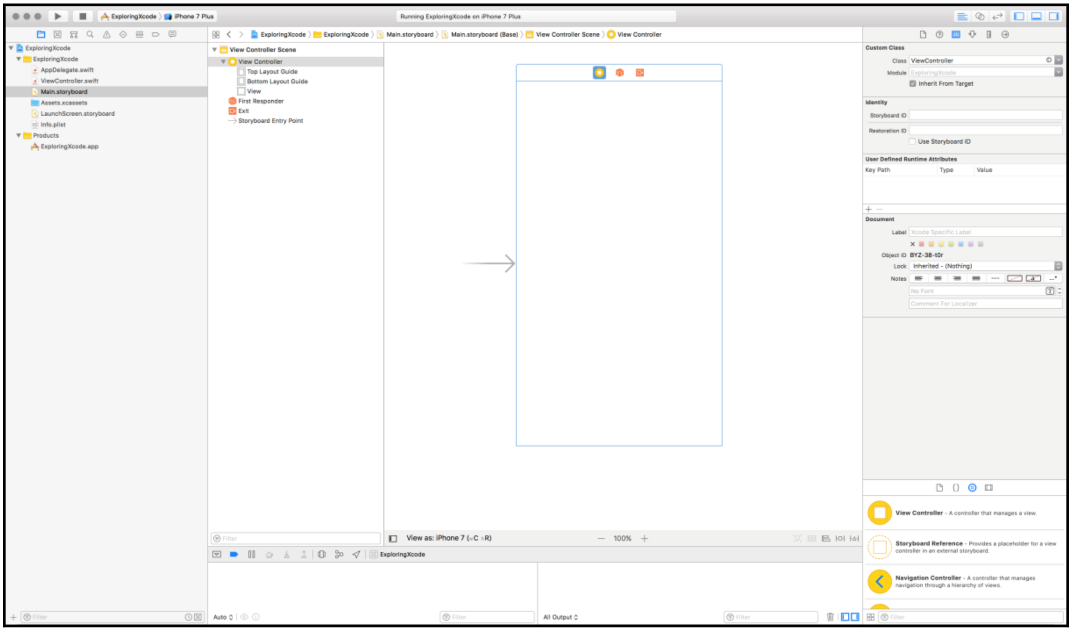
Name	Model	watchOS	Identifier

INSTALLED APPS

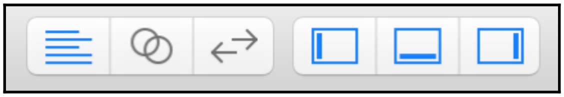
Name	Version	Identifier

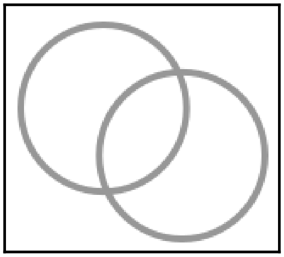
No apps installed

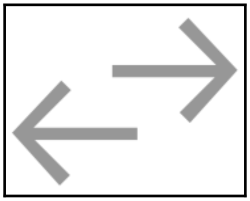

Unavailable Devices
Xclusive iPad Pro (OS version lower than deployment target)
Xclusive iPhone XS Max (OS version lower than deployment target)
Build Only Device
Generic iOS Device
iOS Simulators
iPad (5th generation)
iPad (6th generation)
iPad Air
iPad Air 2
iPad Pro (9.7-inch)
iPad Pro (10.5-inch)
iPad Pro (11-inch)
iPad Pro (12.9-inch)
iPad Pro (12.9-inch) (2nd generation)
iPad Pro (12.9-inch) (3rd generation)
iPhone 5s
iPhone 6
iPhone 6 Plus
iPhone 6s
iPhone 6s Plus
iPhone 7
iPhone 7 Plus
iPhone 8
iPhone 8 Plus
iPhone SE
✓ iPhone X
iPhone XR
iPhone XS
iPhone XS Max
Add Additional Simulators...
Download Simulators...

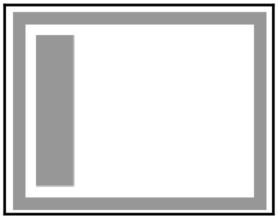

ExploringXcode 〉 iPhone 7 Plus
Running ExploringXcode on iPhone 7 Plus
ExploringXcode 〉 ExploringXcode 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 View Controller Scene 〉 View Controller
ExploringXcode
ExploringXcode
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
ExploringXcode.app
View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
First Responder
Exit
Storyboard Entry Point
View as: iPhone 7 (wC hR)
100%
Auto
All Output
Filter
Custom Class
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label
Xcode Specific Label
Notes
No Font
Comment For Localizer
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.

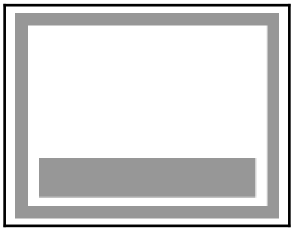
Class ViewController
Module ExploringXcode
☑ Inherit From Target
Object ID BYZ-38-t0r
Lock Inherited - (Nothing)

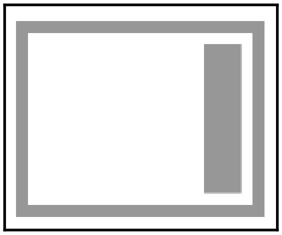




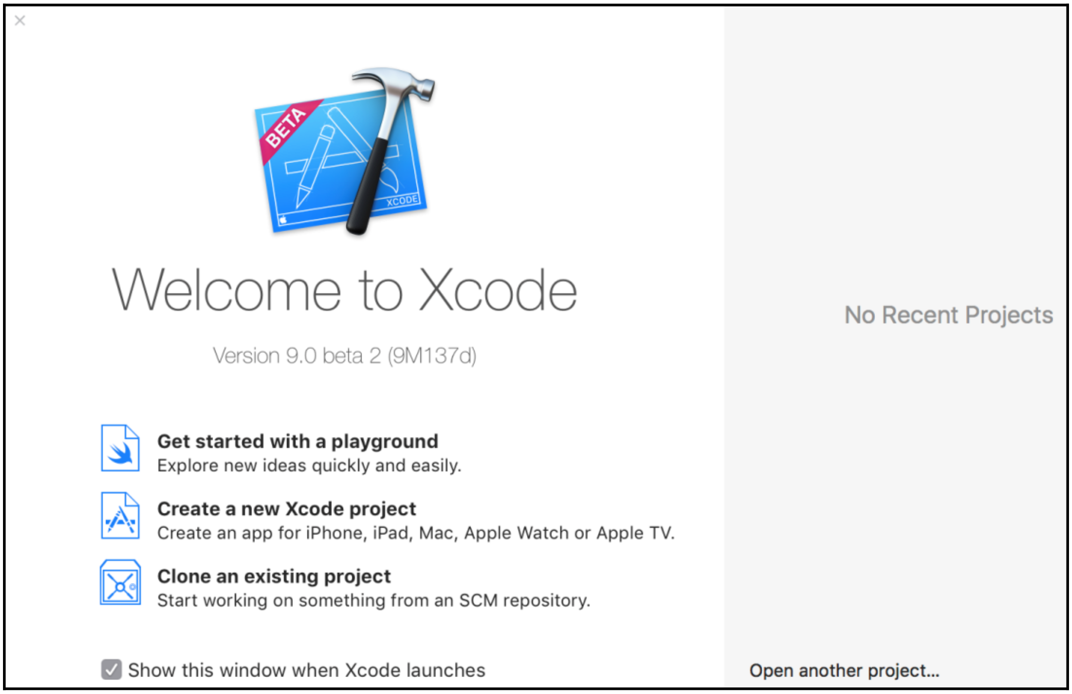








Chapter 2:
Building a Foundation with Swift



Welcome to Xcode

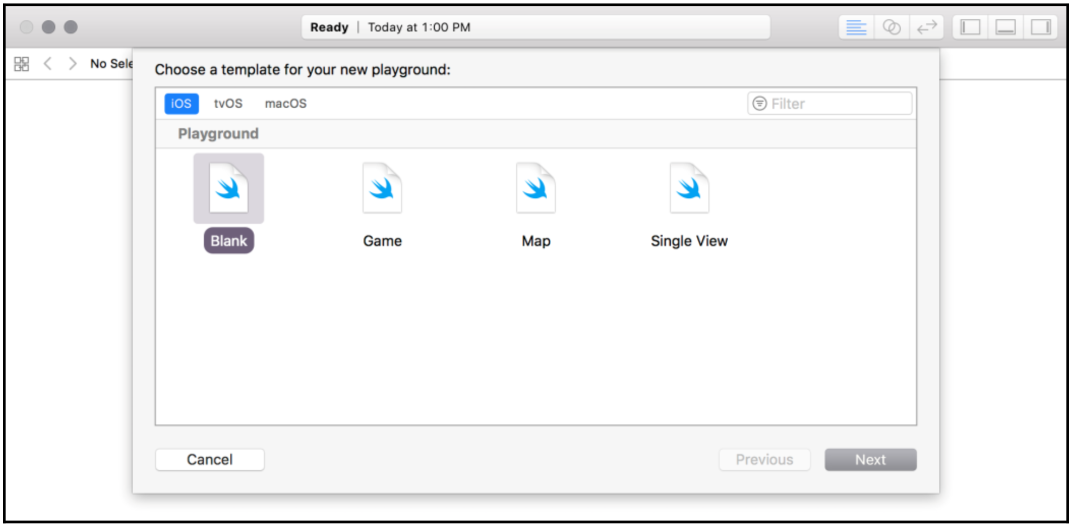
Version 9.0 beta 2 (9M137d)

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.

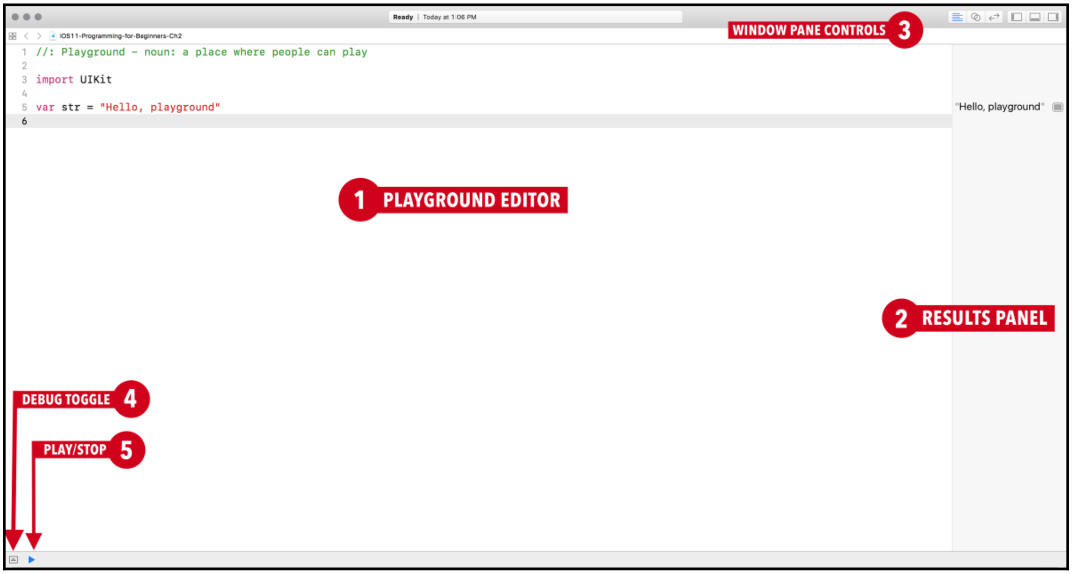
Clone an existing project
Start working on something from an SCM repository.

☑ Show this window when Xcode launches



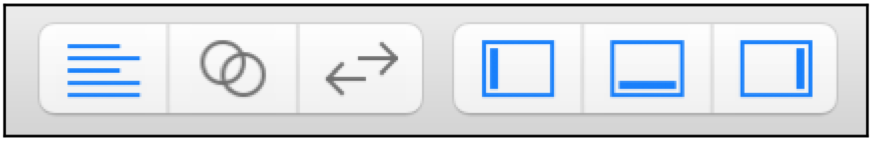
Ready | Today at 1:00 PM

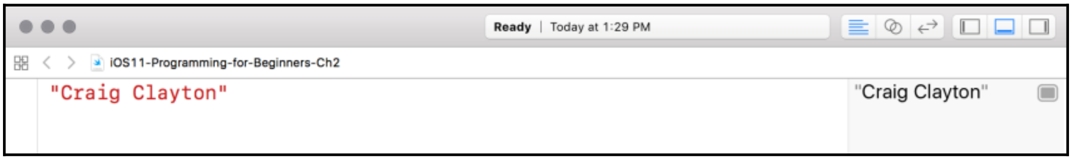
Choose a template for your new playground:



```
//: Playground - noun: a place where people can play

import UIKit

var str = "Hello, playground"
```

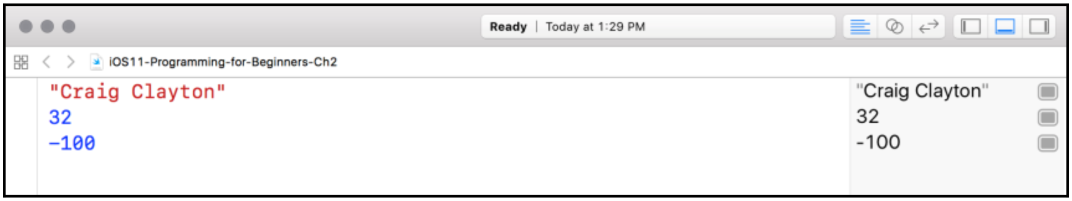
"Hello, playground"





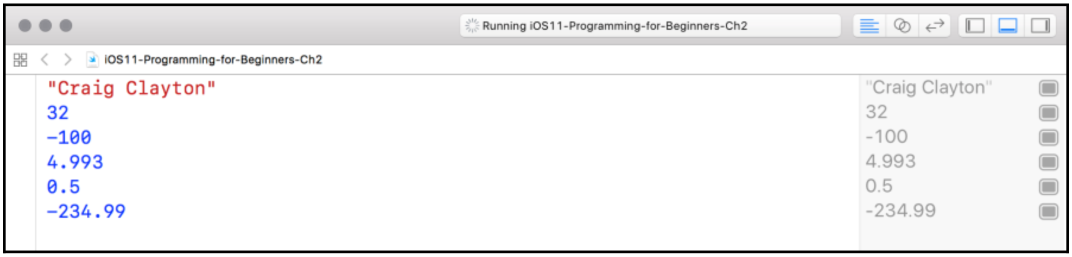
```
"Craig Clayton"
```

"Craig Clayton"



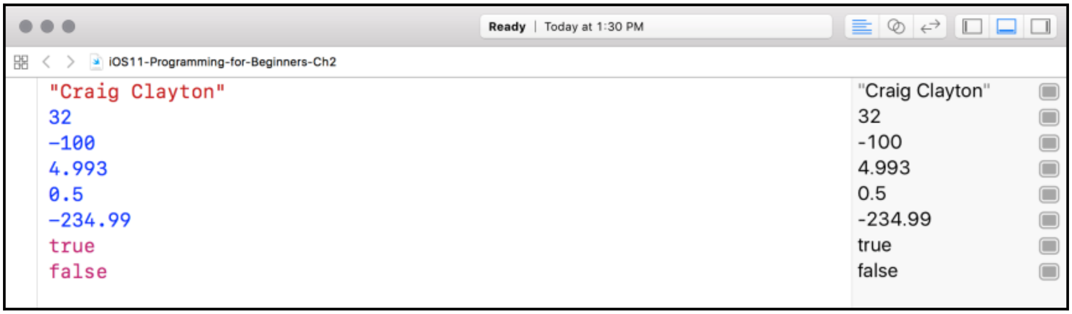
```
"Craig Clayton"
32
-100
```

```
"Craig Clayton"
32
-100
```


Running iOS11-Programming-for-Beginners-Ch2
iOS11-Programming-for-Beginners-Ch2

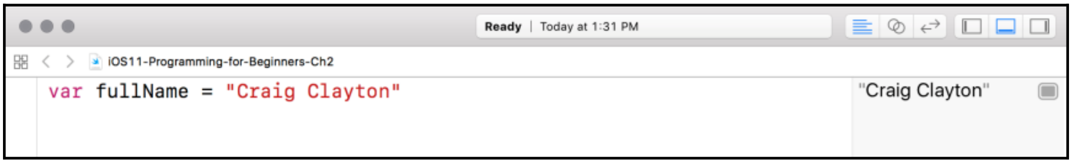
```
"Craig Clayton"
32
-100
4.993
0.5
-234.99
```

```
"Craig Clayton"
32
-100
4.993
0.5
-234.99
```



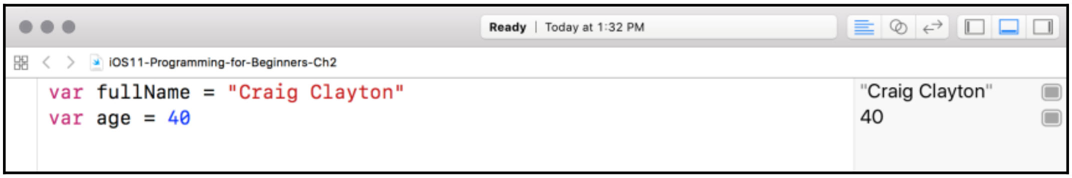
```
"Craig Clayton"
32
-100
4.993
0.5
-234.99
true
false
```

```
"Craig Clayton"
32
-100
4.993
0.5
-234.99
true
false
```



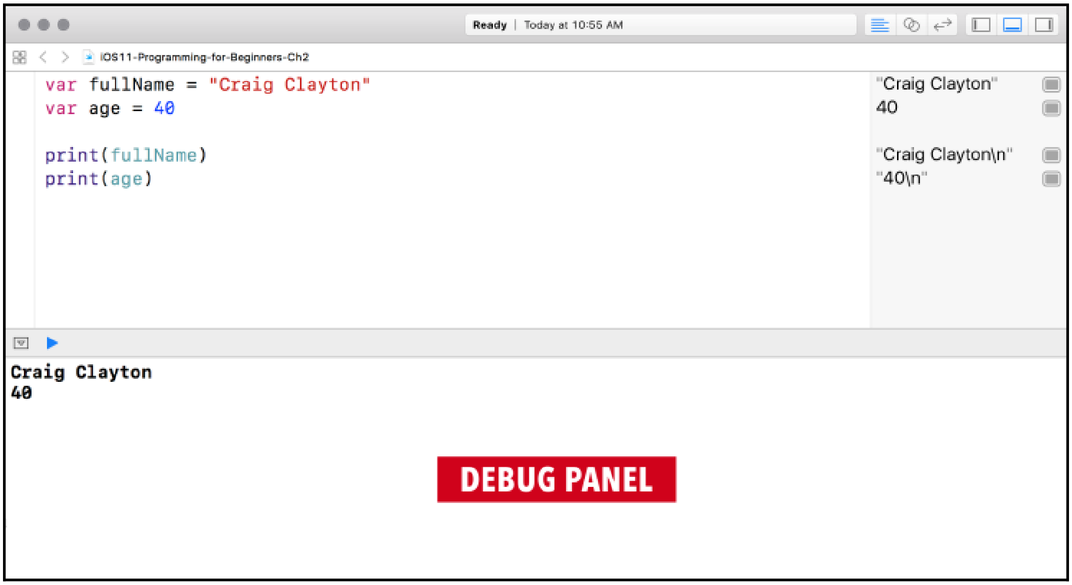
```
var fullName = "Craig Clayton"
```

```
"Craig Clayton"
```



```
var fullName = "Craig Clayton"
var age = 40
```

```
"Craig Clayton"
40
```

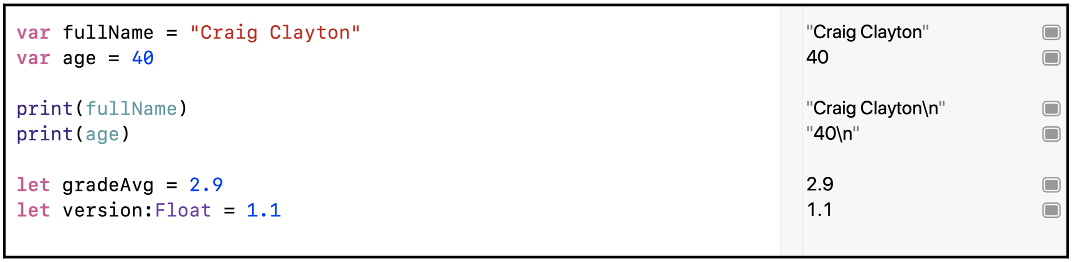


Ready | Today at 10:55 AM
iOS11-Programming-for-Beginners-Ch2

```
var fullName = "Craig Clayton"
var age = 40

print(fullName)
print(age)
```

```
"Craig Clayton"
40

"Craig Clayton\n"
"40\n"
```

```
Craig Clayton
40
```

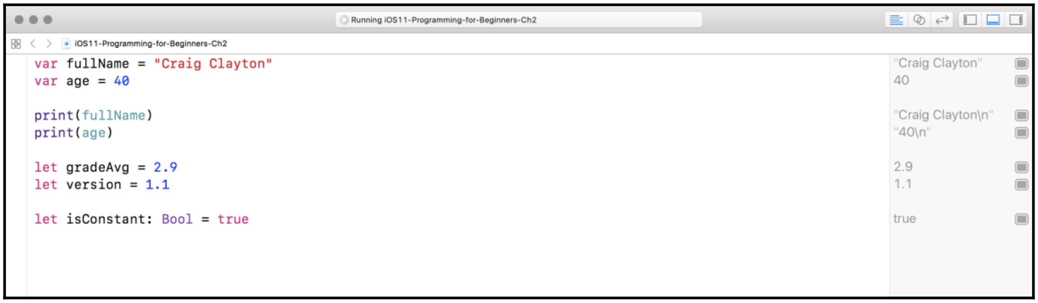


```
var fullName = "Craig Clayton"
var age = 40

print(fullName)
print(age)

let gradeAvg = 2.9
let version:Float = 1.1
```

```
"Craig Clayton"
40

"Craig Clayton\n"
"40\n"

2.9
1.1
```

Double vs Float

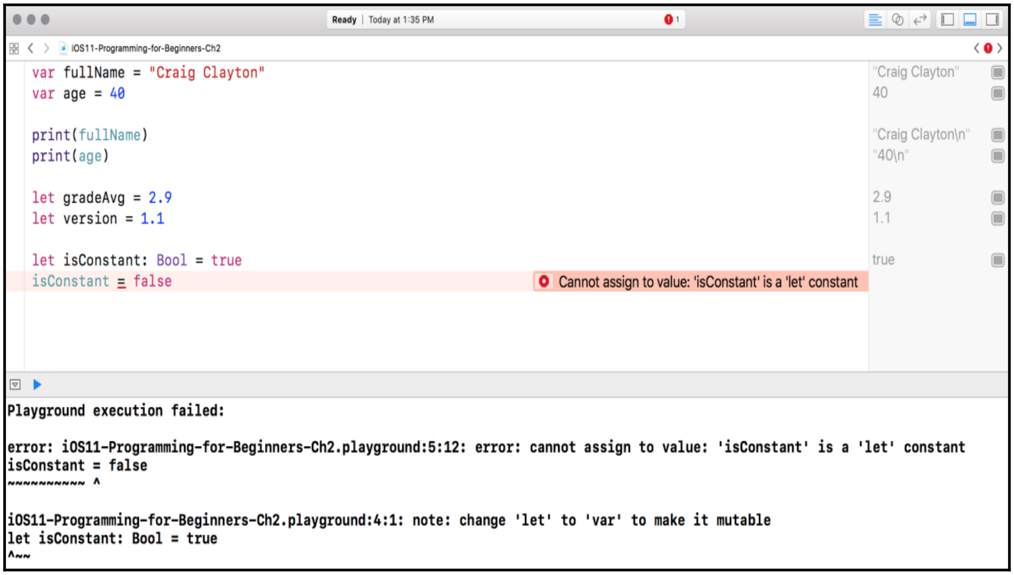
let lessPrecisePI = Float("3.14")
let morePrecisePI = Double("3.1415926536")



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
```

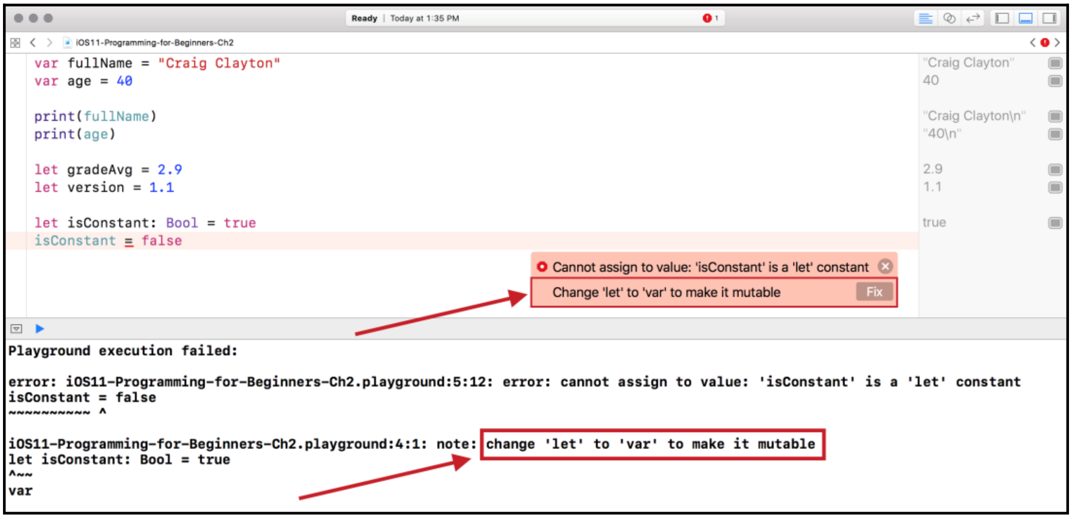
```
"Craig Clayton"
40

"Craig Clayton\n"
"40\n"

2.9
1.1

true
```



```swift
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

```
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
^~~
```

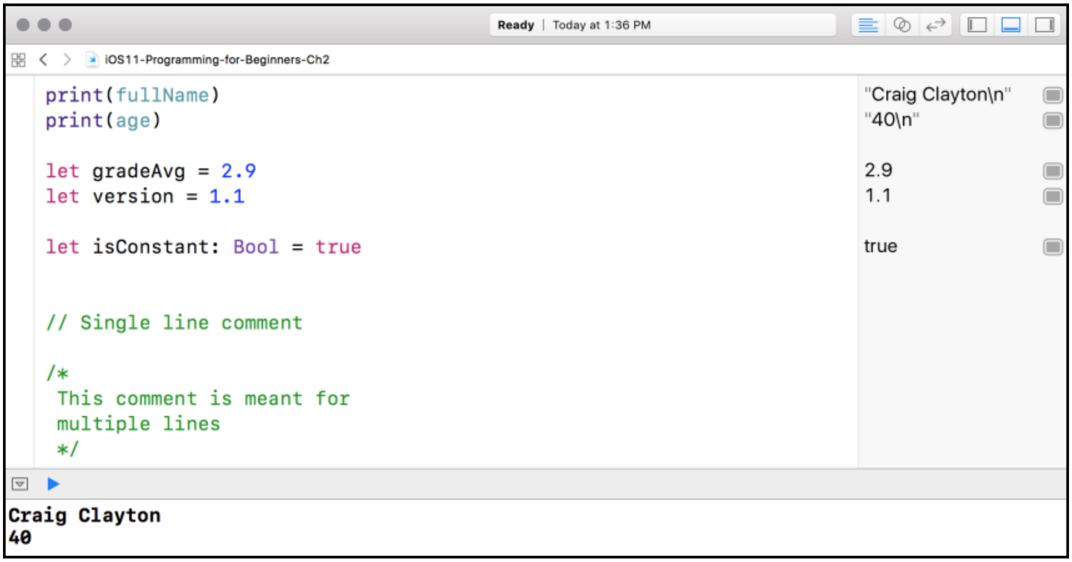


```swift
var fullName = "Craig Clayton"
var age = 40

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

Change 'let' to 'var' to make it mutable

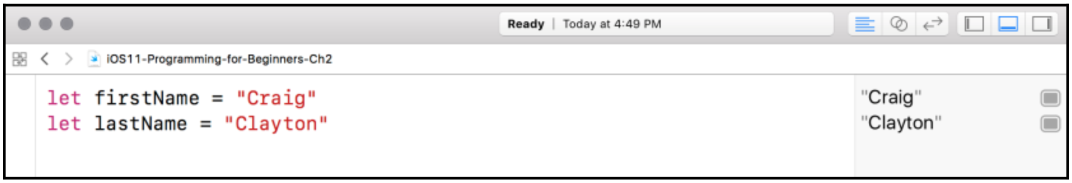
```
Playground execution failed:

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
```



```swift
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
 */
```

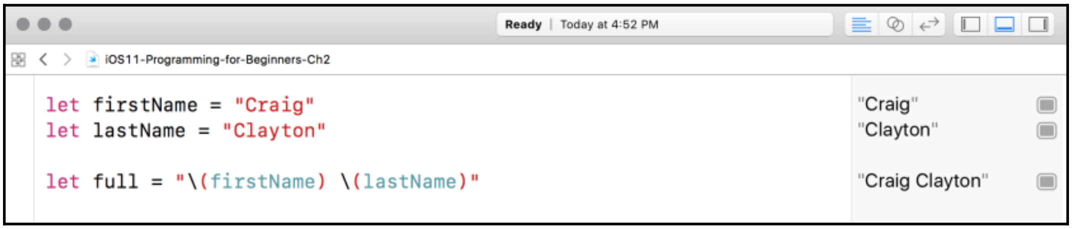
```
"Craig Clayton\n"
"40\n"

2.9
1.1

true
```

```
Craig Clayton
40
```



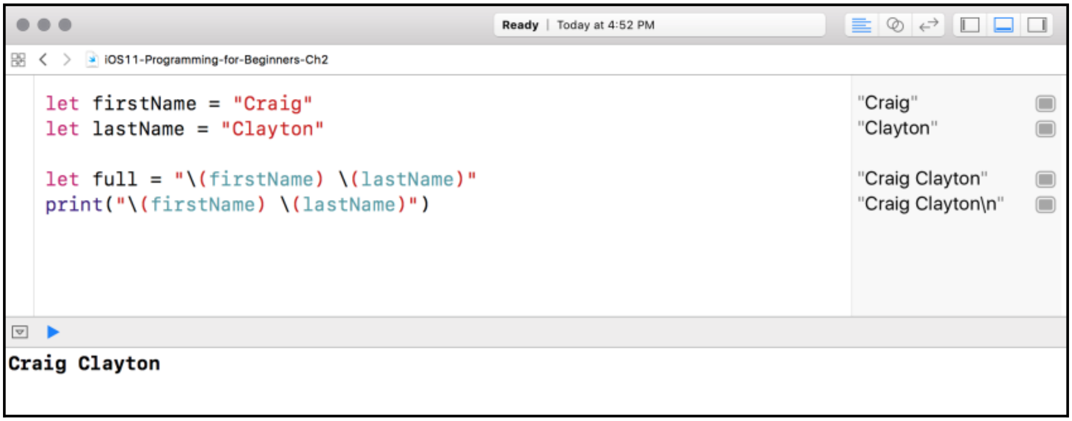
```swift
let firstName = "Craig"
let lastName = "Clayton"
```

```
"Craig"
"Clayton"
```



```swift
let firstName = "Craig"
let lastName = "Clayton"

let full = "\(firstName) \(lastName)"
```

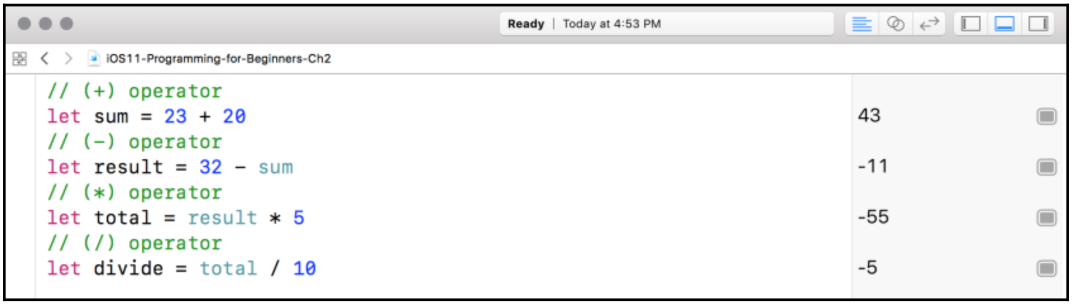
```
"Craig"
"Clayton"

"Craig Clayton"
```



```swift
let firstName = "Craig"
let lastName = "Clayton"

let full = "\(firstName) \(lastName)"
print("\(firstName) \(lastName)")
```

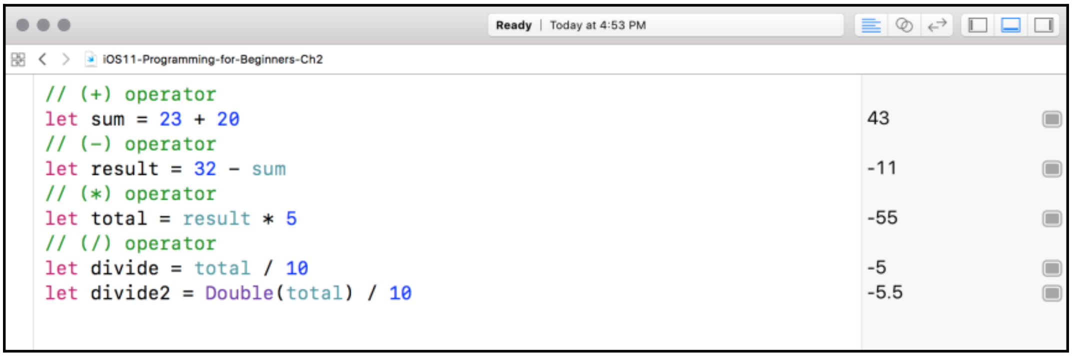
```
"Craig"
"Clayton"

"Craig Clayton"
"Craig Clayton\n"
```

```
Craig Clayton
```



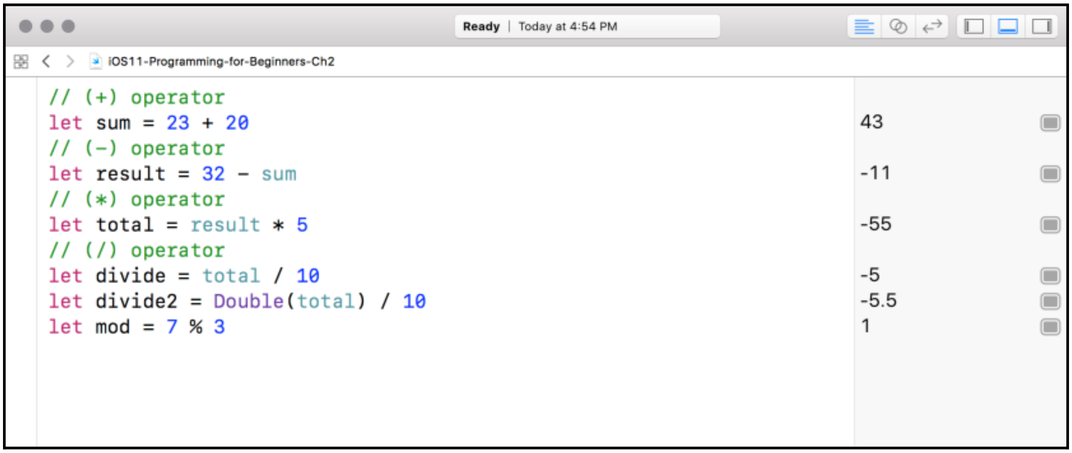
```swift
// (+) operator
let sum = 23 + 20
// (-) operator
let result = 32 - sum
// (*) operator
let total = result * 5
// (/) operator
let divide = total / 10
```

```
43

-11

-55

-5
```

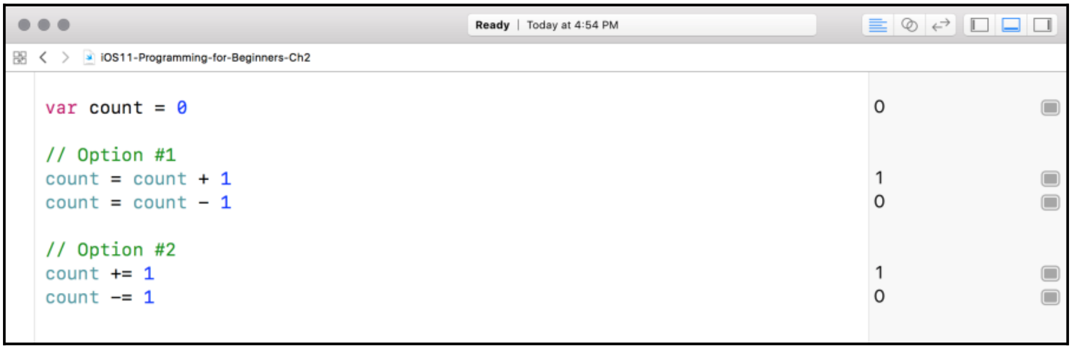


```swift
// (+) operator
let sum = 23 + 20
// (-) operator
let result = 32 - sum
// (*) operator
let total = result * 5
// (/) operator
let divide = total / 10
let divide2 = Double(total) / 10
```

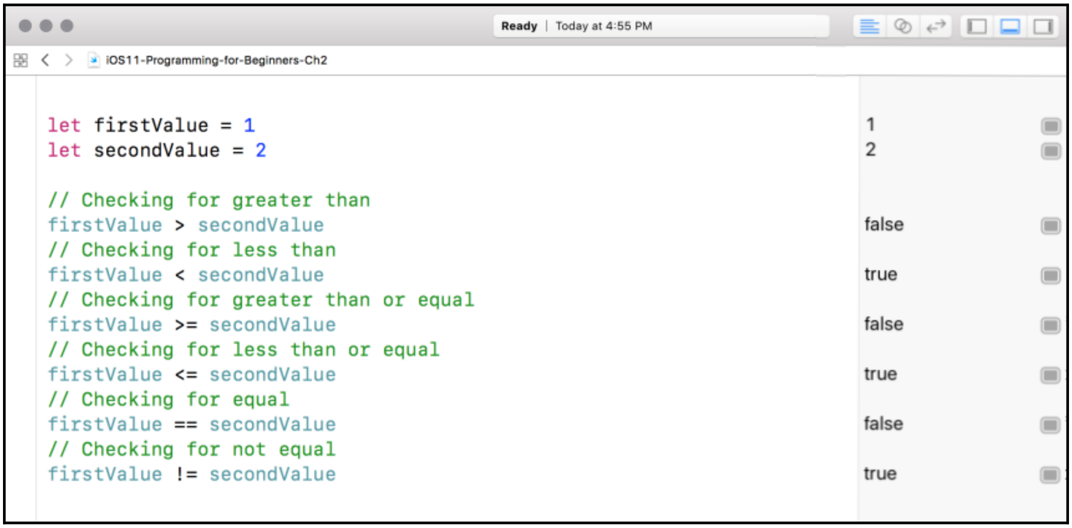
```
43

-11

-55

-5
-5.5
```



```
// (+) operator
let sum = 23 + 20                                   43
// (-) operator
let result = 32 - sum                               -11
// (*) operator
let total = result * 5                              -55
// (/) operator
let divide = total / 10                             -5
let divide2 = Double(total) / 10                    -5.5
let mod = 7 % 3                                     1
```

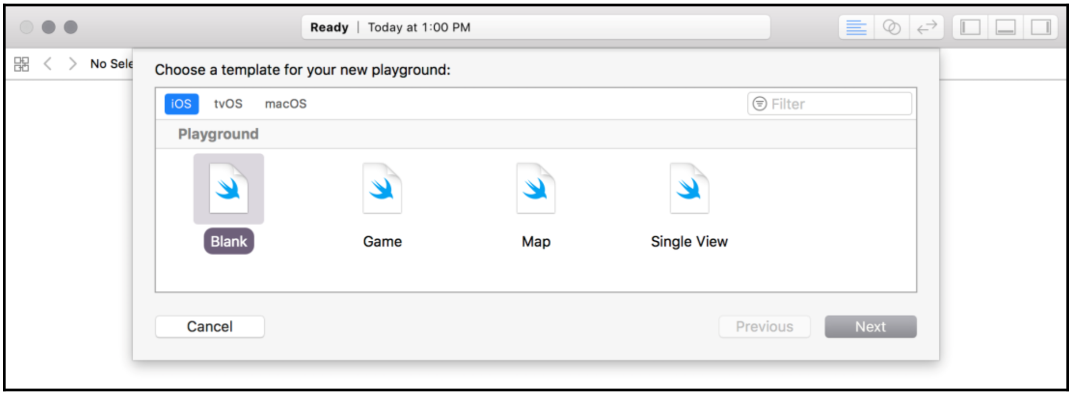


```
var count = 0                                       0

// Option #1
count = count + 1                                   1
count = count - 1                                   0

// Option #2
count += 1                                          1
count -= 1                                          0
```

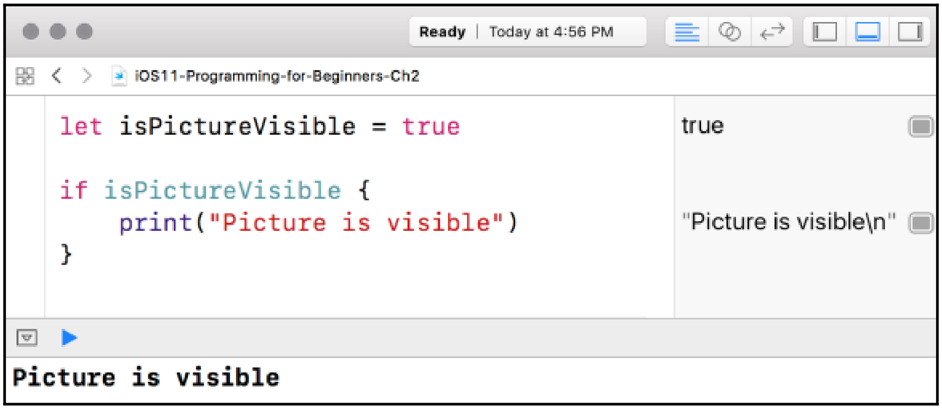


```
let firstValue = 1                                  1
let secondValue = 2                                 2

// Checking for greater than
firstValue > secondValue                            false
// Checking for less than
firstValue < secondValue                            true
// Checking for greater than or equal
firstValue >= secondValue                           false
// Checking for less than or equal
firstValue <= secondValue                           true
// Checking for equal
firstValue == secondValue                           false
// Checking for not equal
firstValue != secondValue                           true
```

Chapter 3:
Building on the Swift Foundation



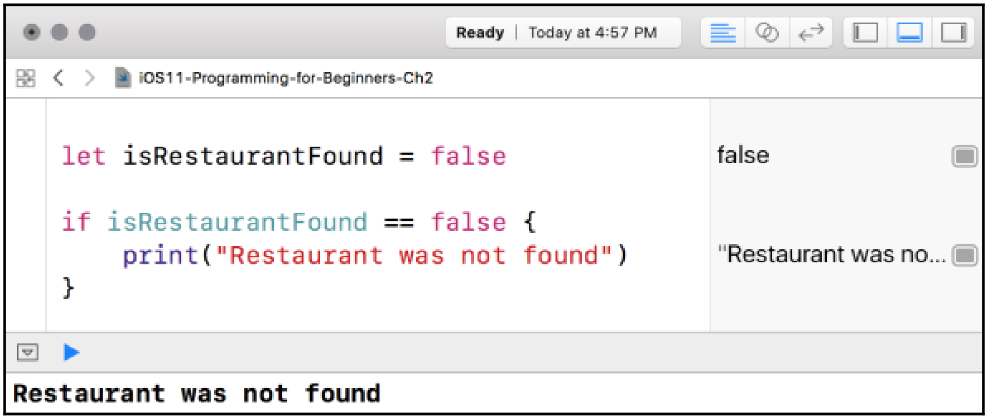
Choose a template for your new playground:



```
let isPictureVisible = true

if isPictureVisible {
    print("Picture is visible")
}
```

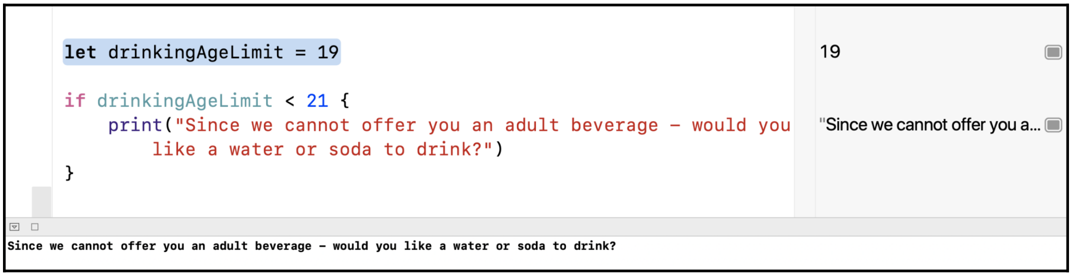
```
true

"Picture is visible\n"
```

```
Picture is visible
```



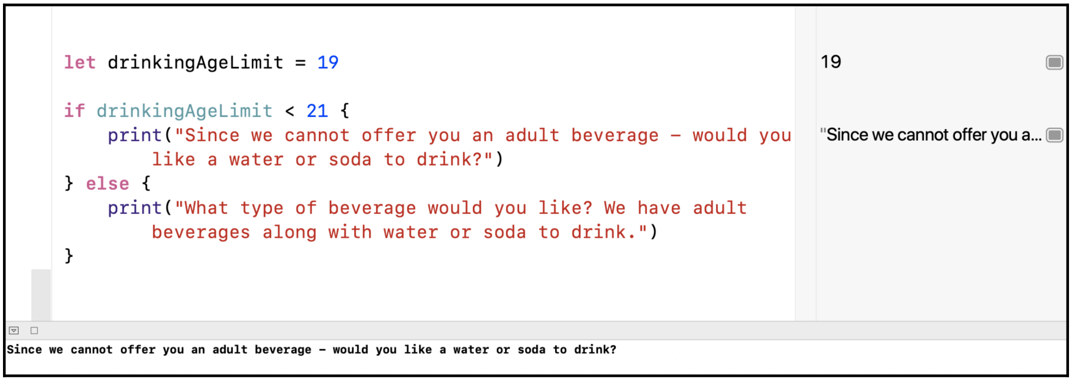
```
let isRestaurantFound = false

if isRestaurantFound == false {
    print("Restaurant was not found")
}
```

```
Restaurant was not found
```



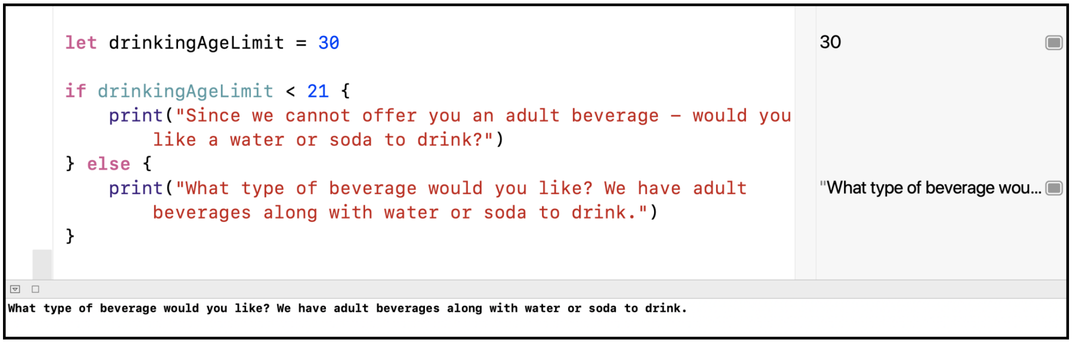
```
let drinkingAgeLimit = 19

if drinkingAgeLimit < 21 {
    print("Since we cannot offer you an adult beverage - would you
        like a water or soda to drink?")
}
```

```
Since we cannot offer you an adult beverage - would you like a water or soda to drink?
```



```
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.")
}
```

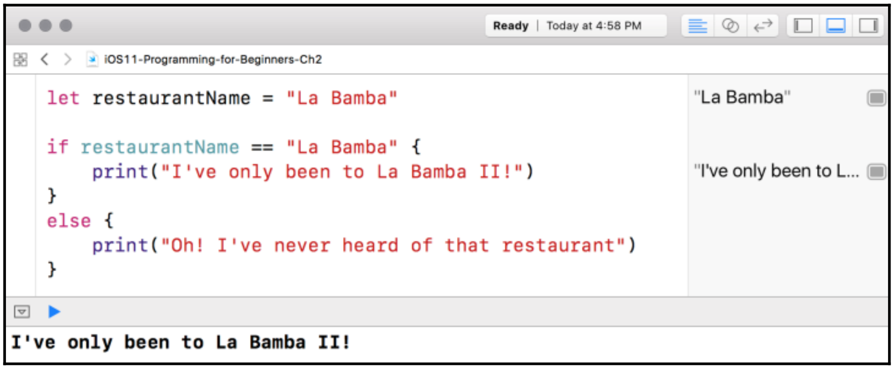
```
Since we cannot offer you an adult beverage - would you like a water or soda to drink?
```



```
let drinkingAgeLimit = 30

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.")
}
```

```
30
"What type of beverage wou...
```

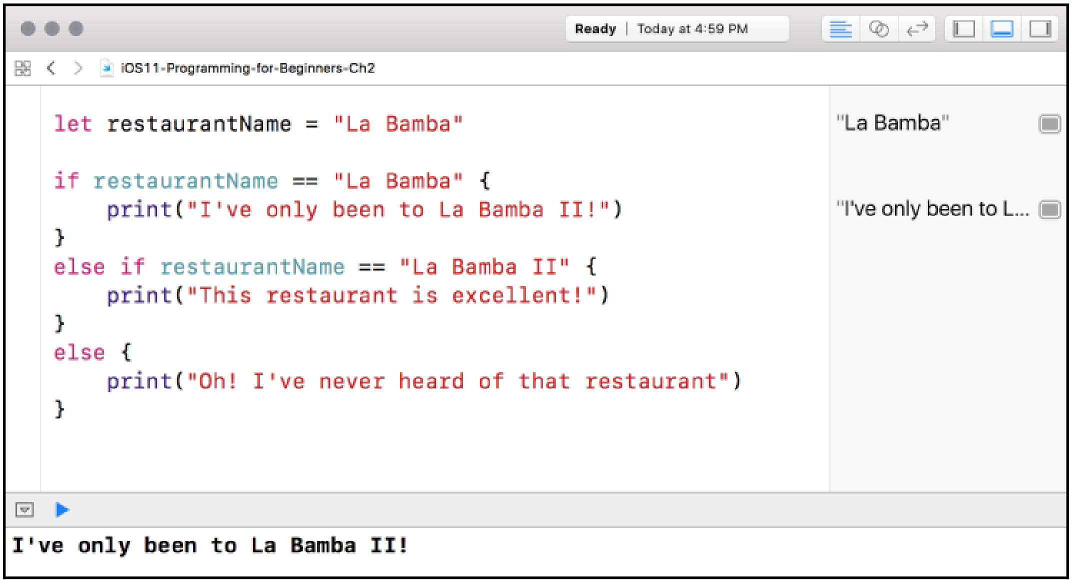
```
What type of beverage would you like? We have adult beverages along with water or soda to drink.
```



```
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...
```

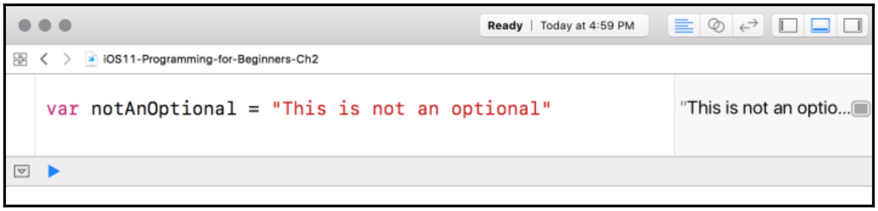
```
I've only been to La Bamba II!
```



```
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")
}
```

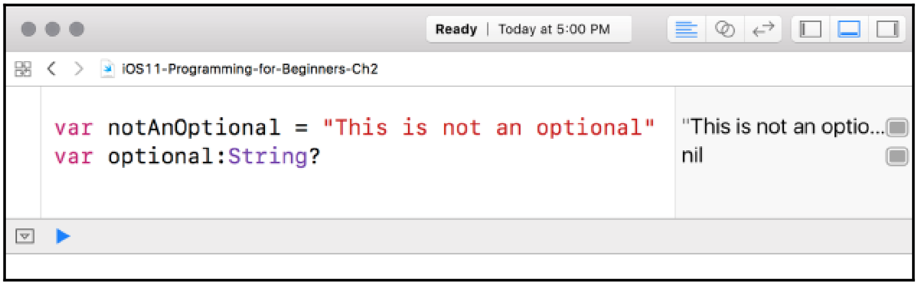
```
"La Bamba"
"I've only been to L...
```

```
I've only been to La Bamba II!
```



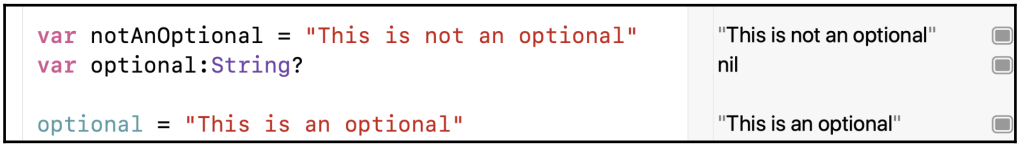
```
var notAnOptional = "This is not an optional"
```

```
"This is not an optio...
```



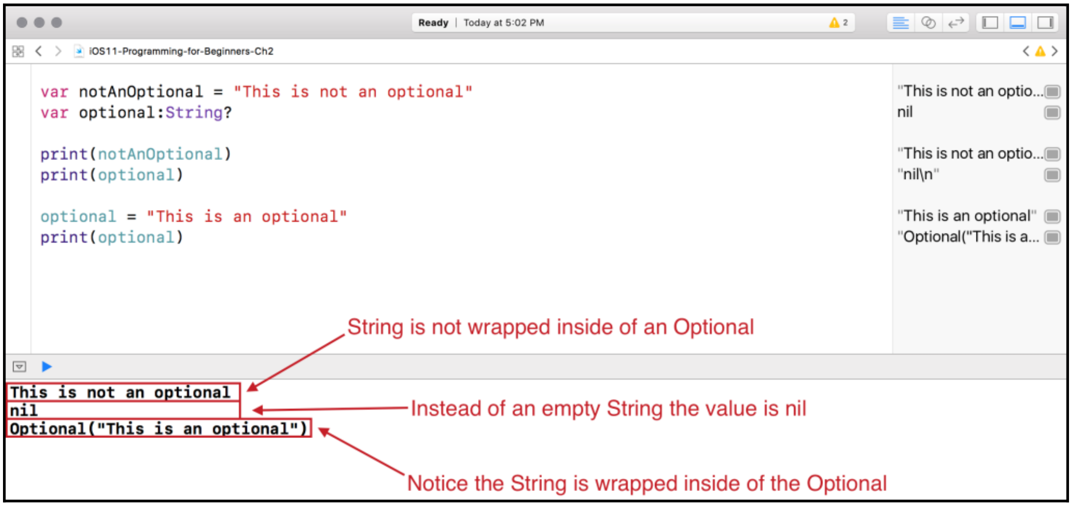
```
var notAnOptional = "This is not an optional"
var optional:String?
```

```
"This is not an optio...
nil
```



```
var notAnOptional = "This is not an optional"
var optional:String?

optional = "This is an optional"
```

```
"This is not an optional"
nil

"This is an optional"
```



```
var notAnOptional = "This is not an optional"
var optional:String?

print(notAnOptional)
print(optional)

optional = "This is an optional"
print(optional)
```

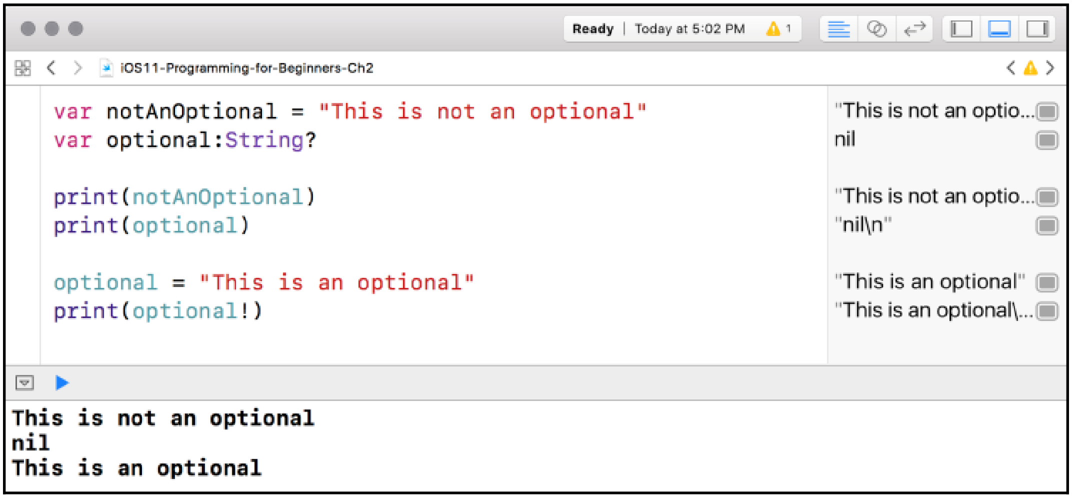
```
"This is not an optio...
nil

"This is not an optio...
"nil\n"

"This is an optional"
"Optional("This is a...
```

```
This is not an optional
nil
Optional("This is an optional")
```

String is not wrapped inside of an Optional

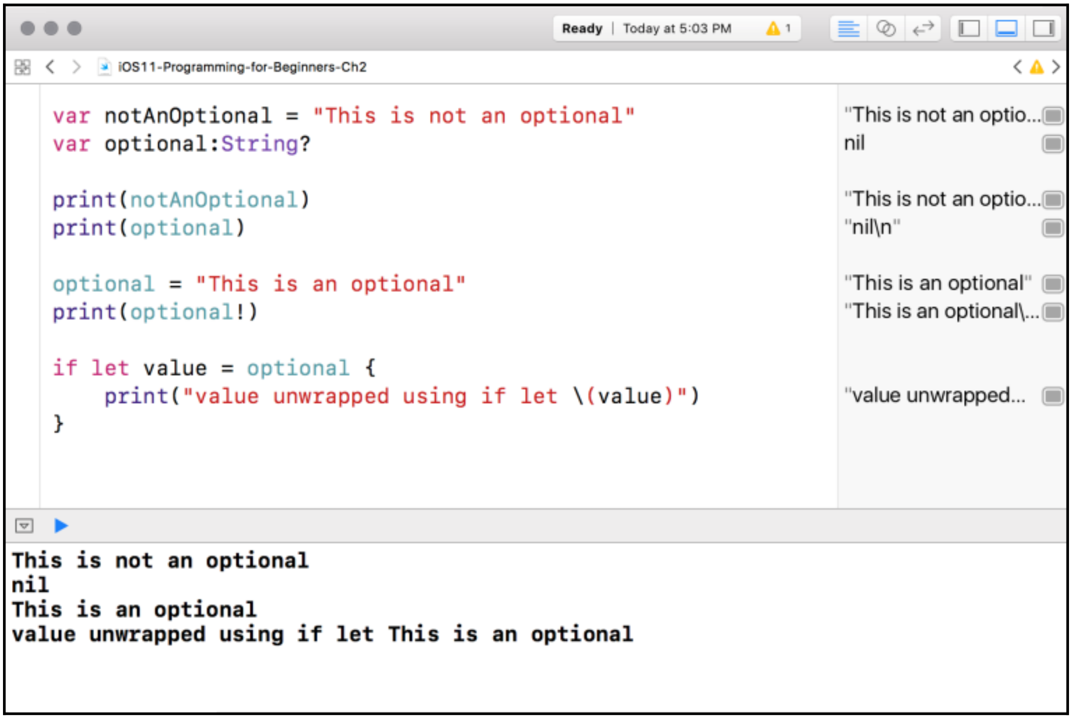
Instead of an empty String the value is nil

Notice the String is wrapped inside of the Optional



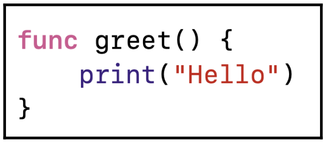
```
var notAnOptional = "This is not an optional"
var optional:String?

print(notAnOptional)
print(optional)

optional = "This is an optional"
print(optional!)
```

```
This is not an optional
nil
This is an optional
```

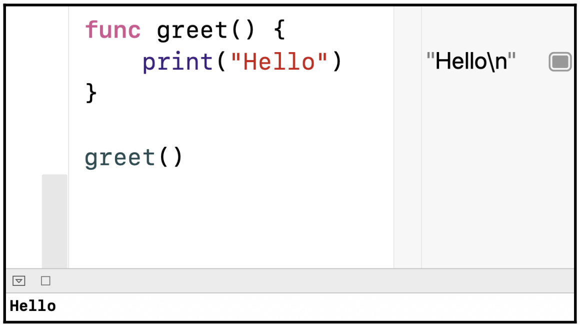


```
var notAnOptional = "This is not an optional"
var optional:String?

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 optional
nil
This is an optional
value unwrapped using if let This is an optional
```

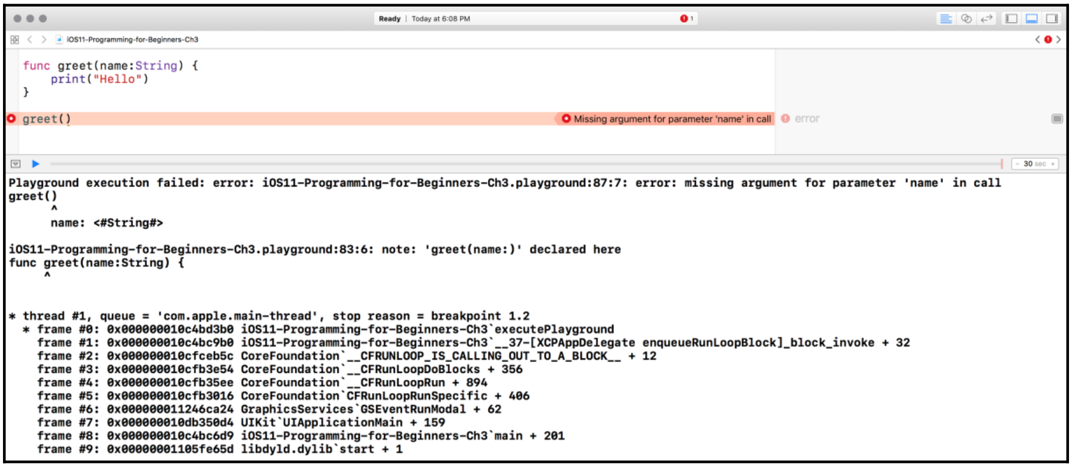


```swift
func greet() {
    print("Hello")
}
```



```swift
func greet() {
    print("Hello")
}

greet()
```

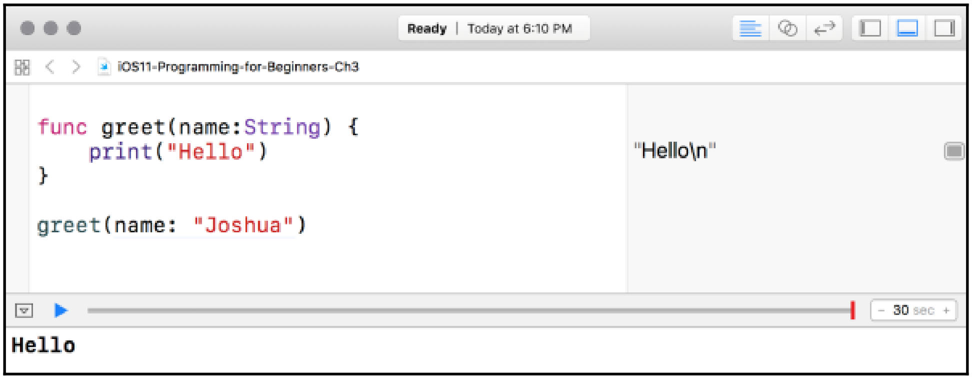
```
Hello
```



```swift
func greet(name:String) {
    print("Hello")
}

greet()
```

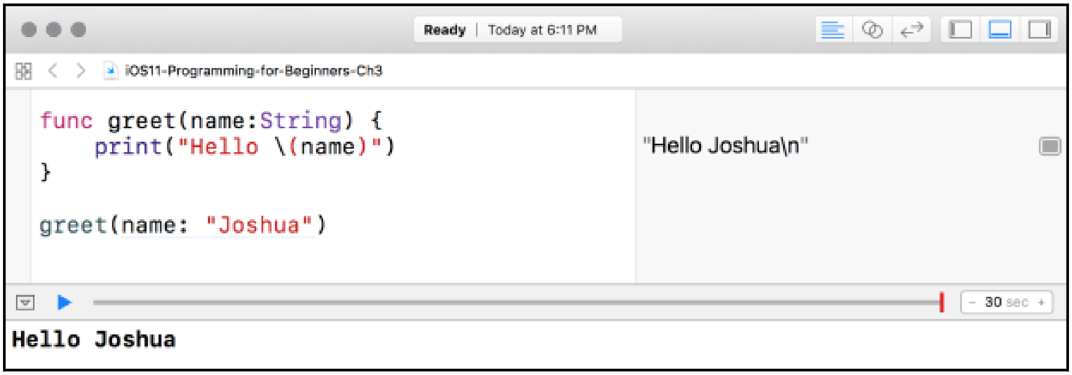
Missing argument for parameter 'name' in call

```
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: 0x000000011246ca24 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
```



```swift
func greet(name:String) {
    print("Hello")
}

greet(name: "Joshua")
```

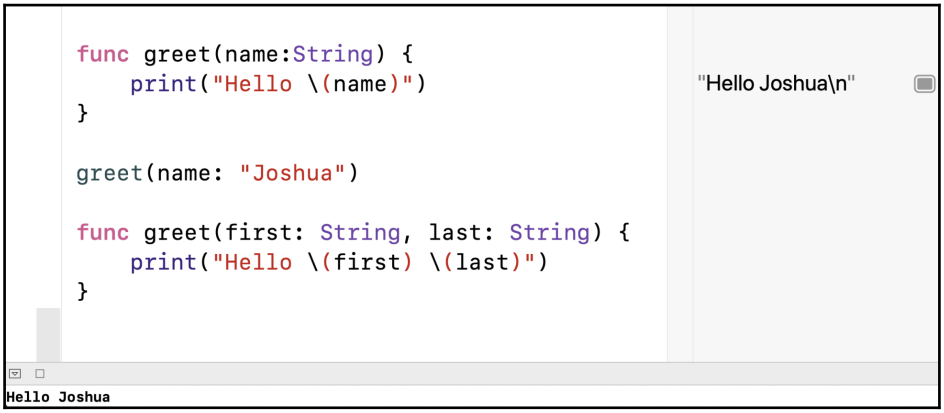
```
Hello
```



```
func greet(name:String) {
    print("Hello \(name)")
}

greet(name: "Joshua")
```

```
"Hello Joshua\n"
```

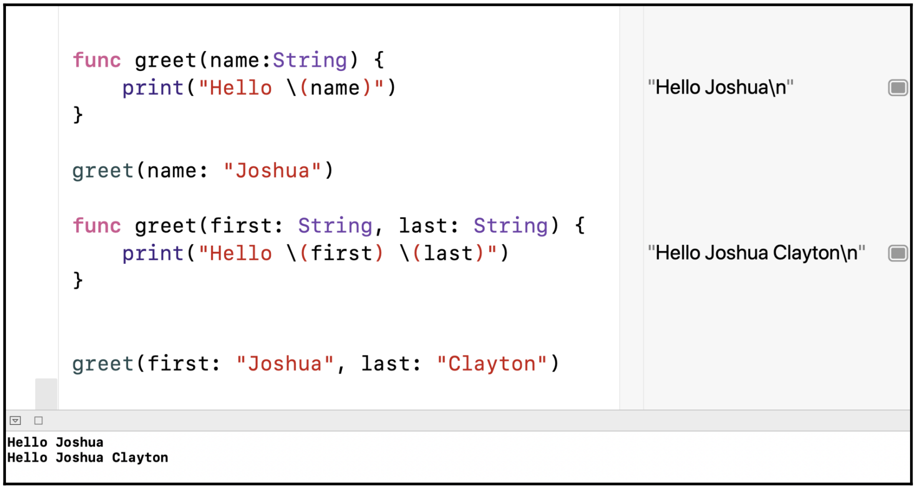
```
Hello Joshua
```



```
func greet(name:String) {
    print("Hello \(name)")
}

greet(name: "Joshua")

func greet(first: String, last: String) {
    print("Hello \(first) \(last)")
}
```

```
"Hello Joshua\n"
```

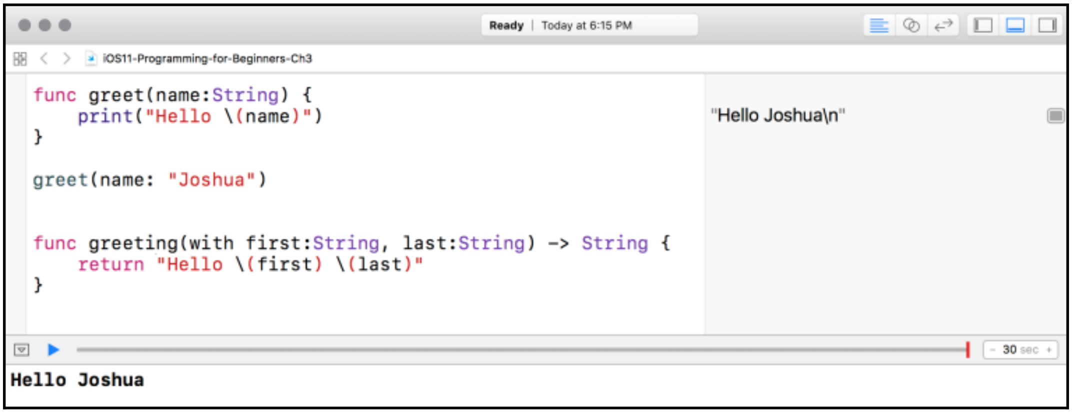
```
Hello Joshua
```



```
func greet(name:String) {
    print("Hello \(name)")
}

greet(name: "Joshua")

func greet(first: String, last: String) {
    print("Hello \(first) \(last)")
}


greet(first: "Joshua", last: "Clayton")
```

```
"Hello Joshua\n"
"Hello Joshua Clayton\n"
```

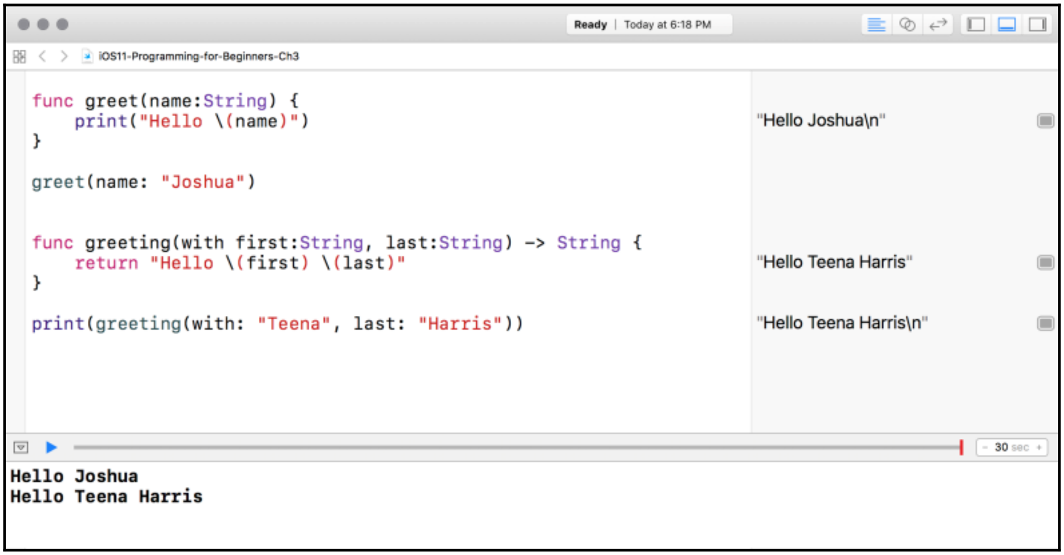
```
Hello Joshua
Hello Joshua Clayton
```



```
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
```

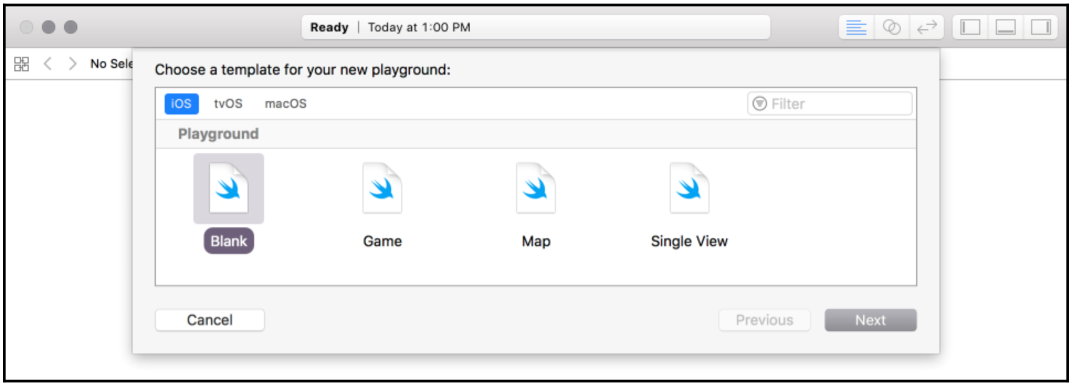


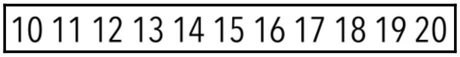
```
func greet(name:String) {
    print("Hello \(name)")
}

greet(name: "Joshua")


func greeting(with first:String, last:String) -> String {
    return "Hello \(first) \(last)"
}

print(greeting(with: "Teena", last: "Harris"))
```

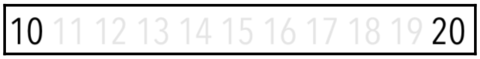
```
"Hello Joshua\n"
"Hello Teena Harris"
"Hello Teena Harris\n"
```

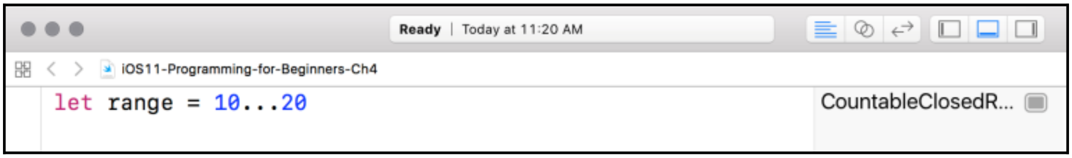
```
Hello Joshua
Hello Teena Harris
```

Chapter 4: Digging Deeper

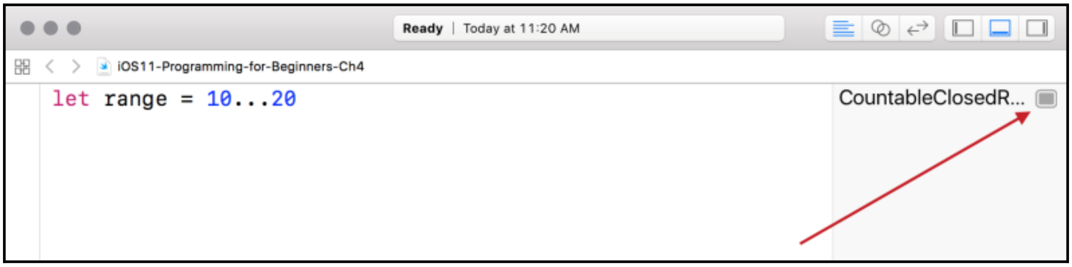




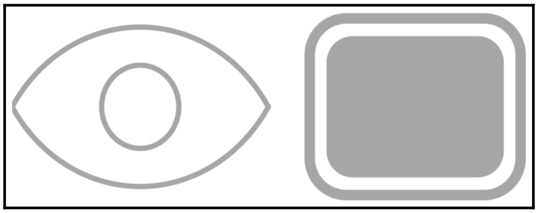


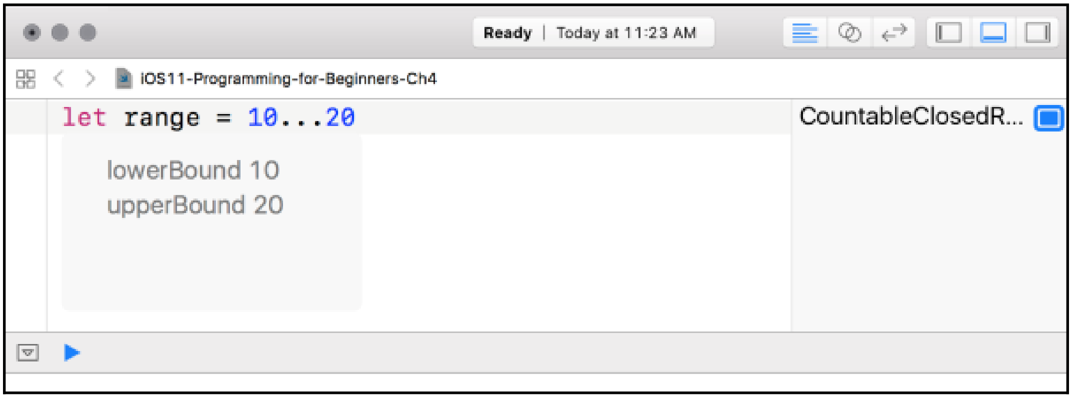

Ready | Today at 11:20 AM
iOS11-Programming-for-Beginners-Ch4
CountableClosedR...

```
let range = 10...20
```


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

```
let range = 10...20
```



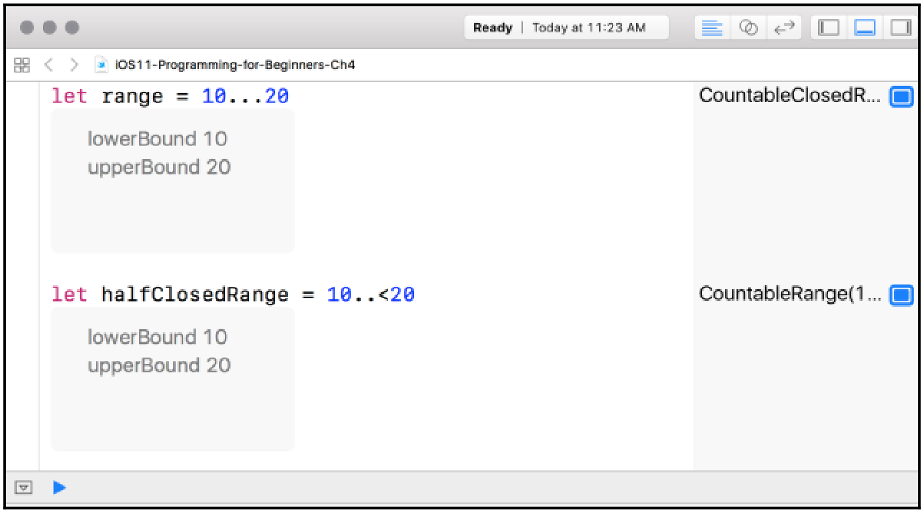


Ready | Today at 11:23 AM
iOS11-Programming-for-Beginners-Ch4

```
let range = 10...20
```

```
lowerBound 10
upperBound 20
```

CountableClosedR...



Ready | Today at 11:23 AM
iOS11-Programming-for-Beginners-Ch4

```
let range = 10...20
```

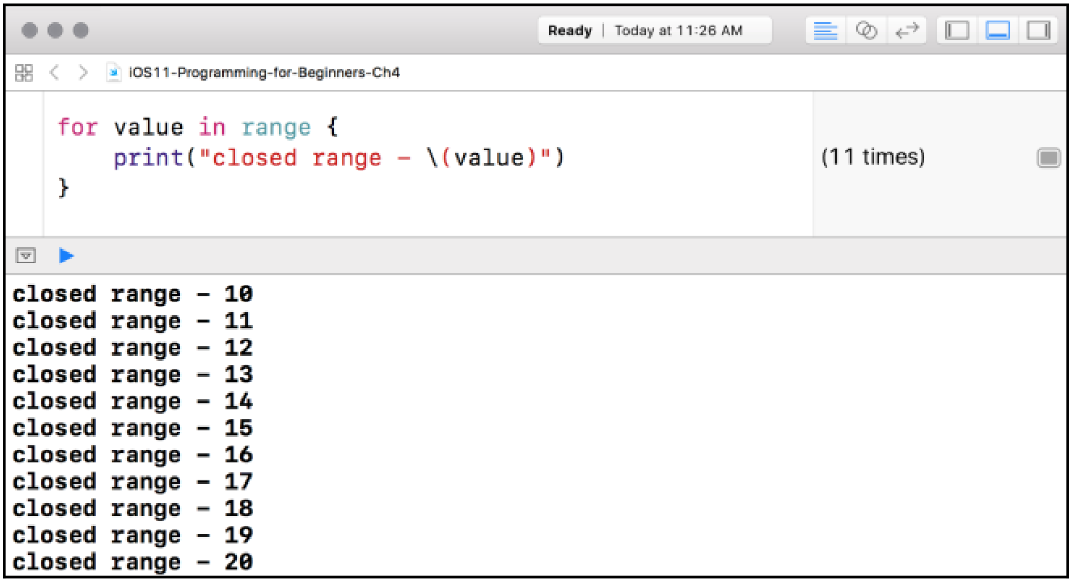
```
lowerBound 10
upperBound 20
```

CountableClosedR...

```
let halfClosedRange = 10..<20
```

```
lowerBound 10
upperBound 20
```

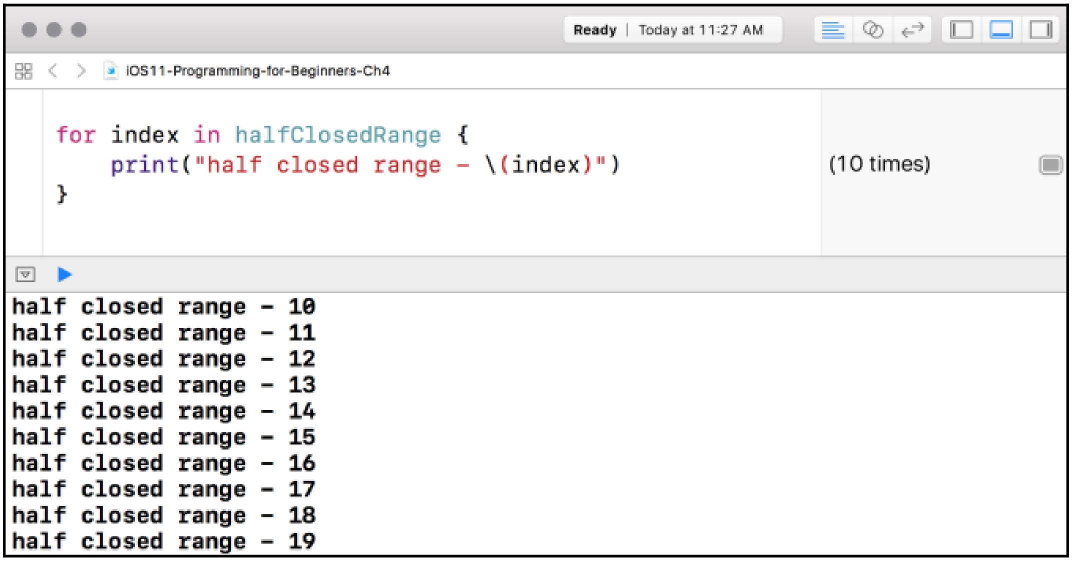
CountableRange(1...



Ready | Today at 11:26 AM
iOS11-Programming-for-Beginners-Ch4
(11 times)

```
for value in range {
    print("closed range - \(value)")
}
```

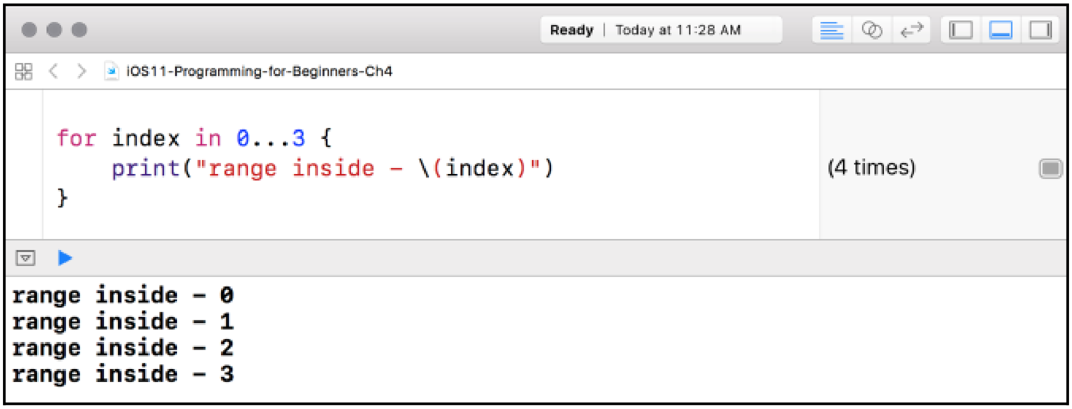
```
closed range - 10
closed range - 11
closed range - 12
closed range - 13
closed range - 14
closed range - 15
closed range - 16
closed range - 17
closed range - 18
closed range - 19
closed range - 20
```



Ready | Today at 11:27 AM
iOS11-Programming-for-Beginners-Ch4
(10 times)

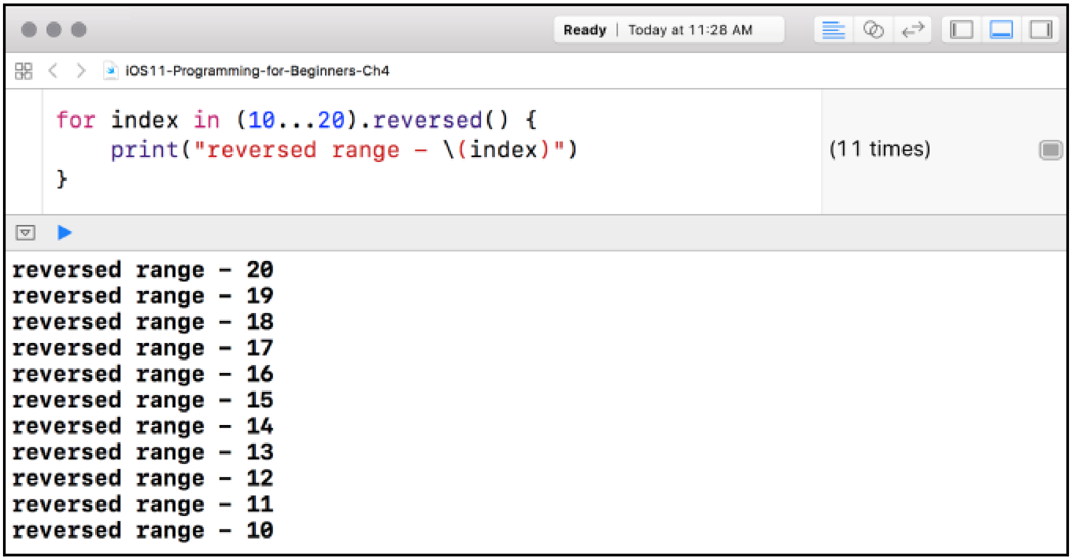
```
for index in halfClosedRange {
    print("half closed range - \(index)")
}
```

```
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
(4 times)

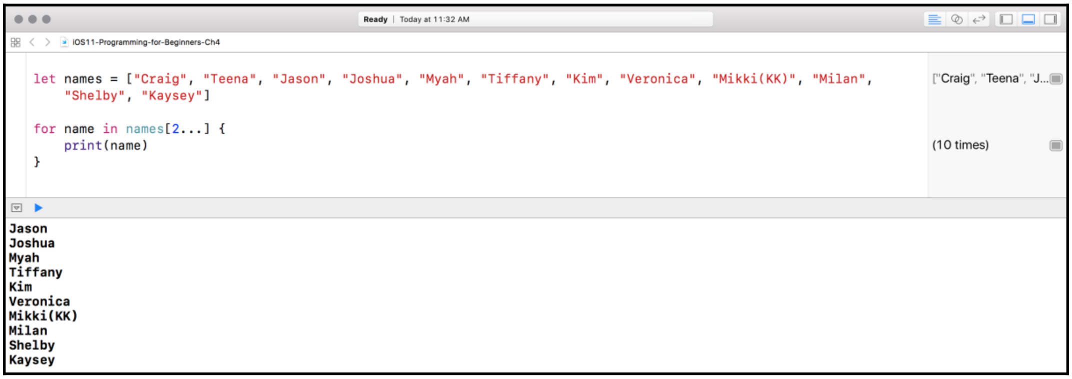
```swift
for index in 0...3 {
    print("range inside - \(index)")
}
```

```
range inside - 0
range inside - 1
range inside - 2
range inside - 3
```


Ready | Today at 11:28 AM
iOS11-Programming-for-Beginners-Ch4
(11 times)

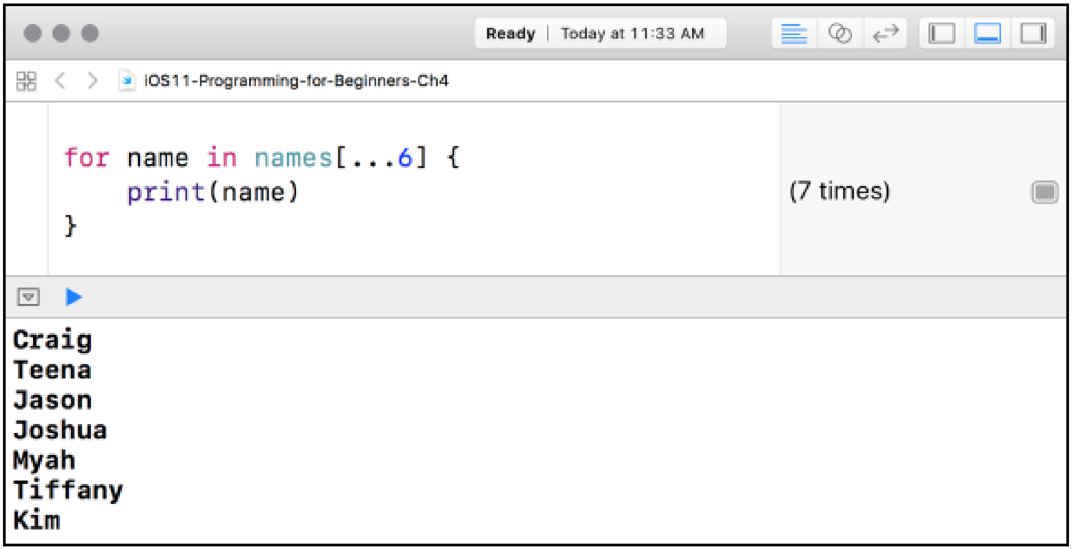
```swift
for index in (10...20).reversed() {
    print("reversed range - \(index)")
}
```

```
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
["Craig", "Teena", "J...
(10 times)

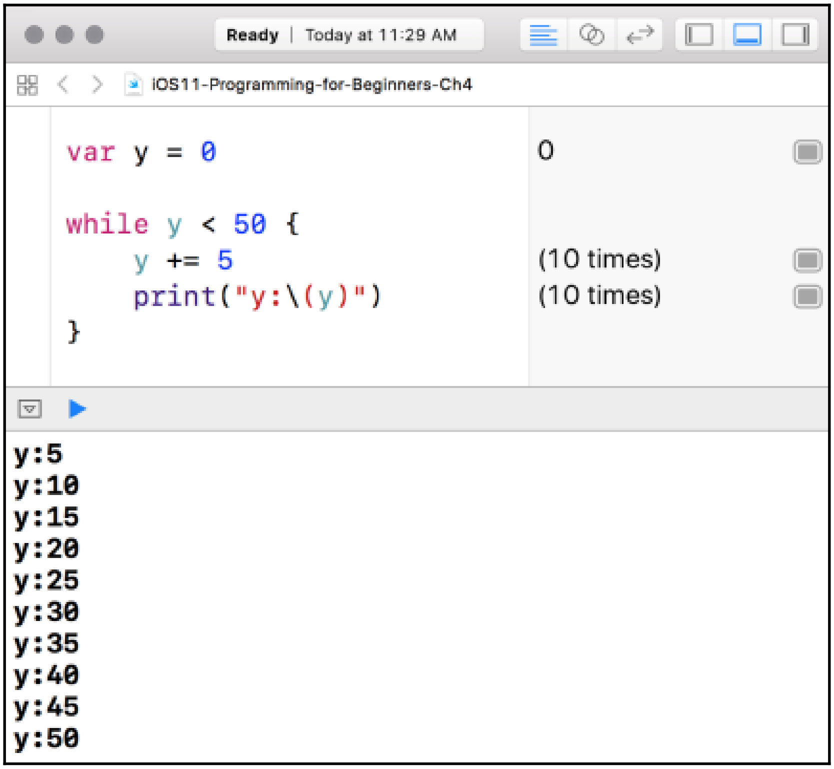
```swift
let names = ["Craig", "Teena", "Jason", "Joshua", "Myah", "Tiffany", "Kim", "Veronica", "Mikki(KK)", "Milan",
    "Shelby", "Kaysey"]

for name in names[2...] {
    print(name)
}
```

```
Jason
Joshua
Myah
Tiffany
Kim
Veronica
Mikki(KK)
Milan
Shelby
Kaysey
```


Ready | Today at 11:33 AM
iOS11-Programming-for-Beginners-Ch4
(7 times)

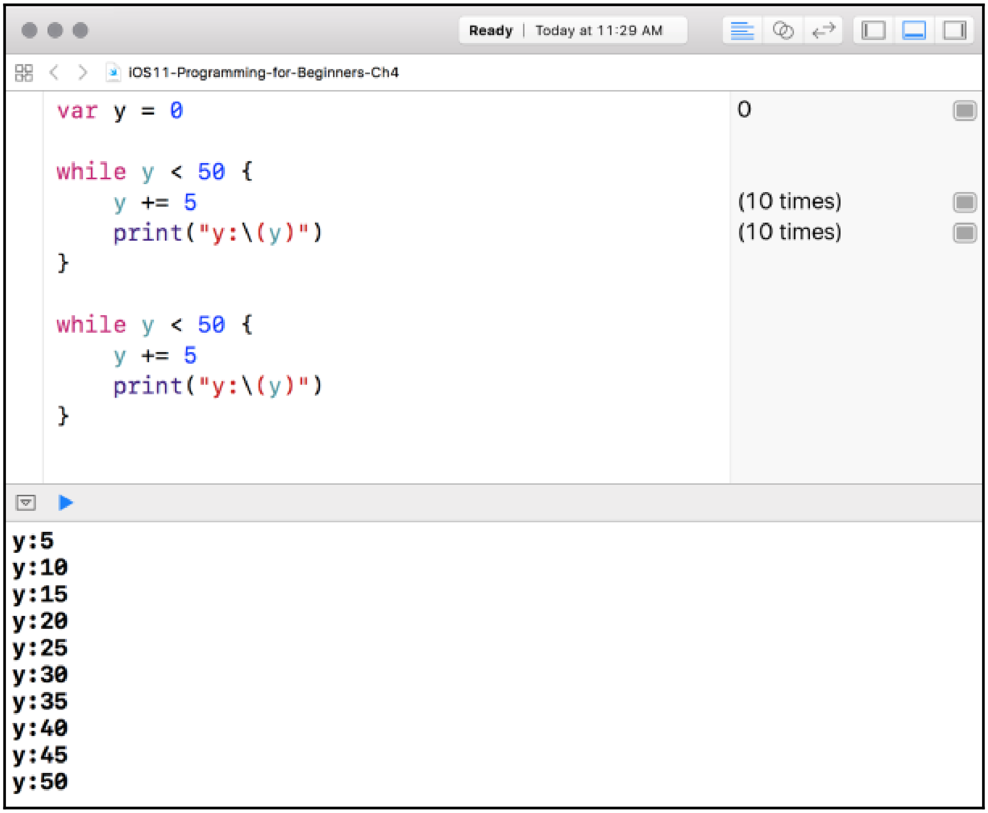
```
for name in names[...6] {
    print(name)
}
```

```
Craig
Teena
Jason
Joshua
Myah
Tiffany
Kim
```


Ready | Today at 11:29 AM
iOS11-Programming-for-Beginners-Ch4
0
(10 times)
(10 times)

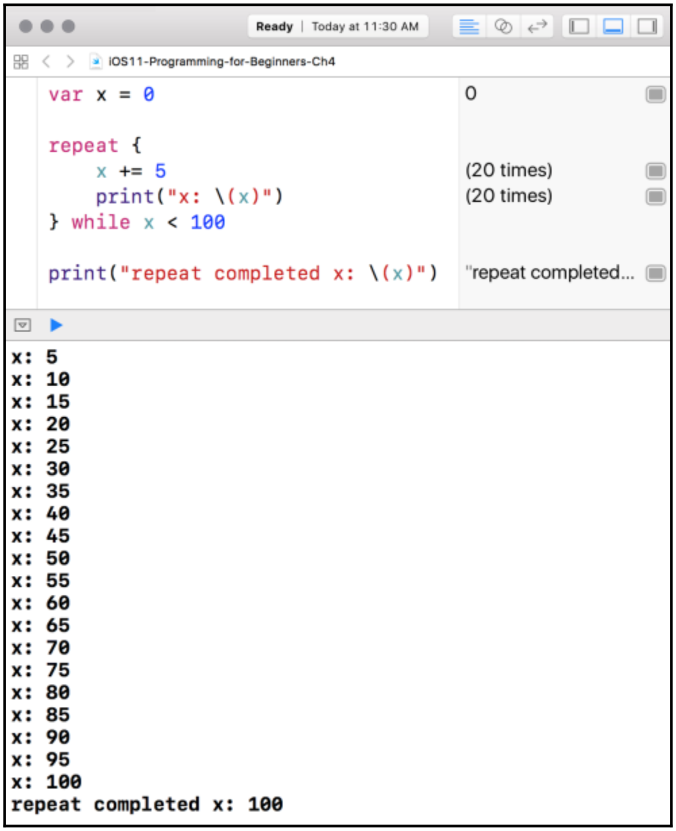
```
var y = 0

while y < 50 {
    y += 5
    print("y:\(y)")
}
```

```
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
0
(10 times)
(10 times)

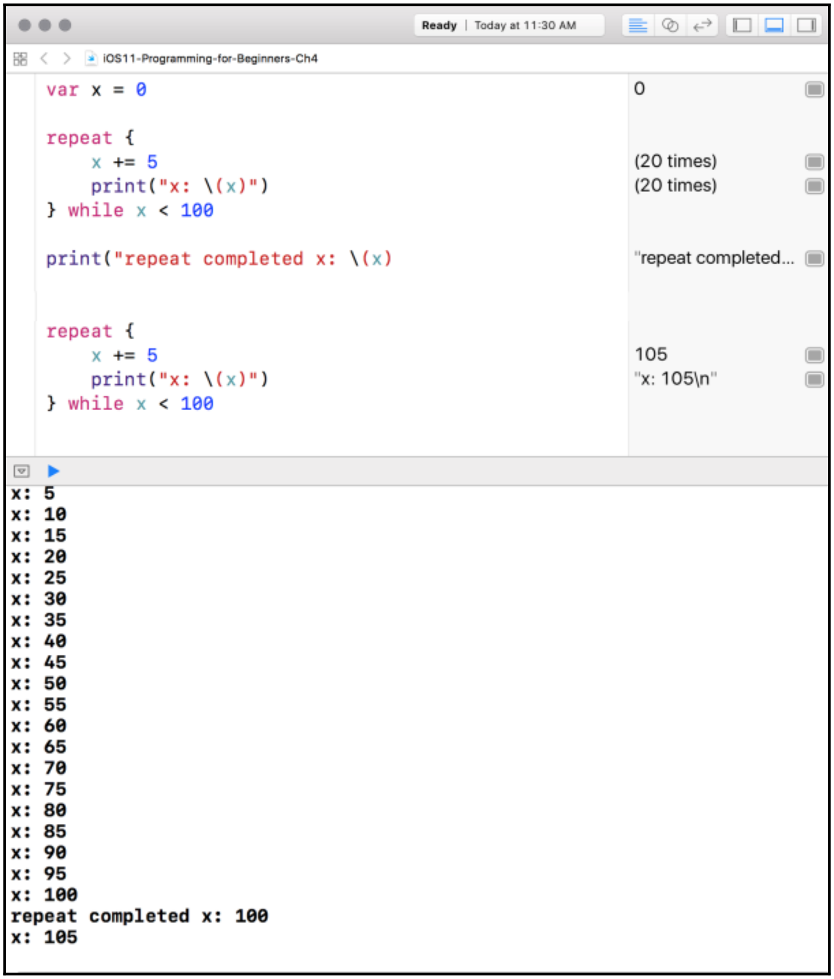
```
var y = 0

while y < 50 {
    y += 5
    print("y:\(y)")
}

while y < 50 {
    y += 5
    print("y:\(y)")
}
```

```
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
0
(20 times)
(20 times)
"repeat completed...

```
var x = 0

repeat {
    x += 5
    print("x: \(x)")
} while x < 100

print("repeat completed x: \(x)")
```

```
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
```

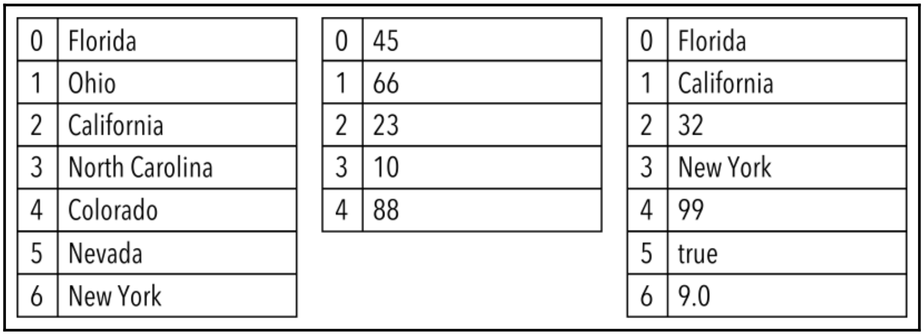

Ready | Today at 11:30 AM
iOS11-Programming-for-Beginners-Ch4

```
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
```

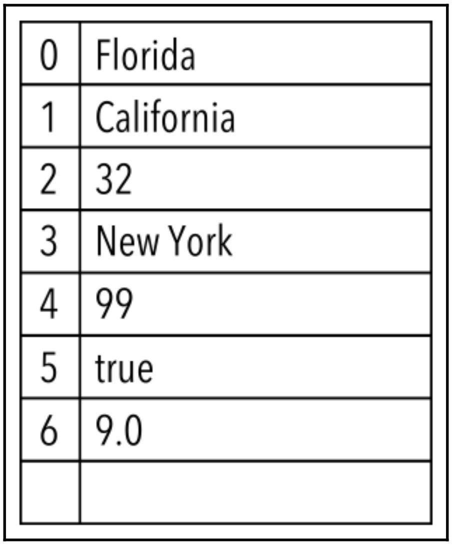
Chapter 5: Digging into Collections



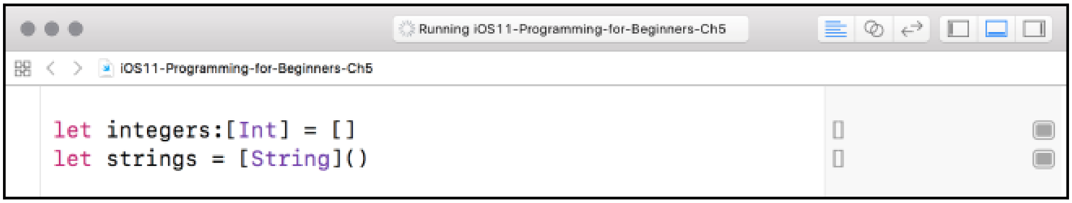
0	Florida
1	Ohio
2	California
3	North Carolina
4	Colorado
5	Nevada
6	New York

0	45
1	66
2	23
3	10
4	88

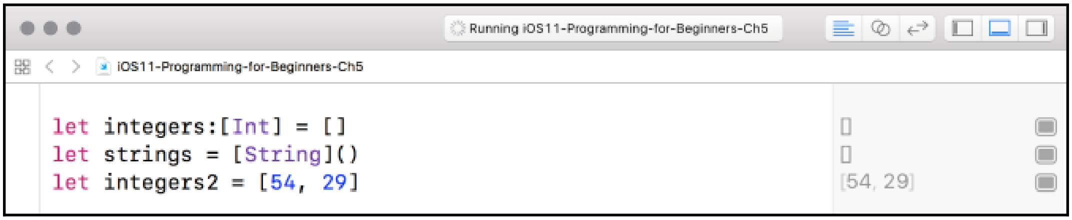
0	Florida
1	California
2	32
3	New York
4	99
5	true
6	9.0



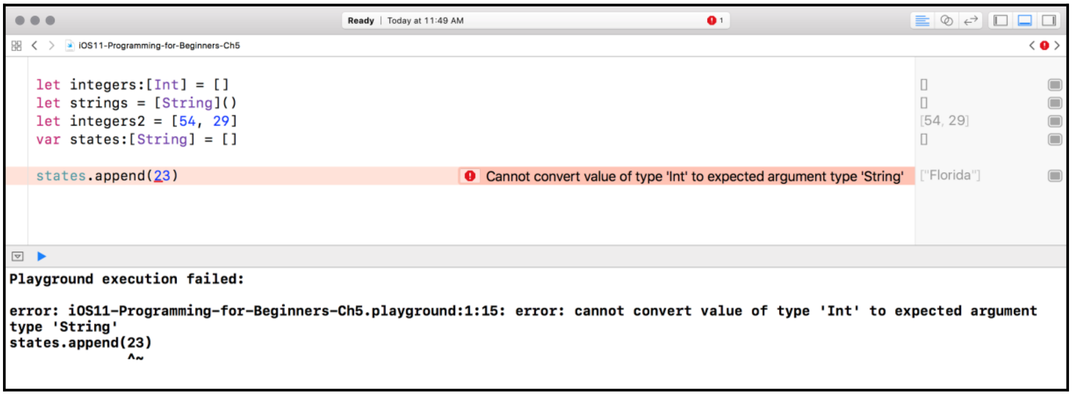
0	Florida
1	California
2	32
3	New York
4	99
5	true
6	9.0


Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5

```
let integers:[Int] = []
let strings = [String]()
```


Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5
[]
[]
[54, 29]

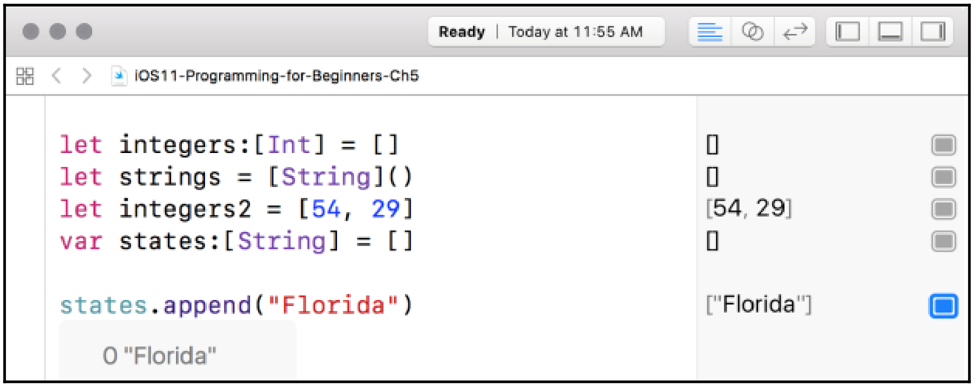
```
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
```


Ready | Today at 11:49 AM
1
iOS11-Programming-for-Beginners-Ch5
[]
[]
[54, 29]
[]
["Florida"]

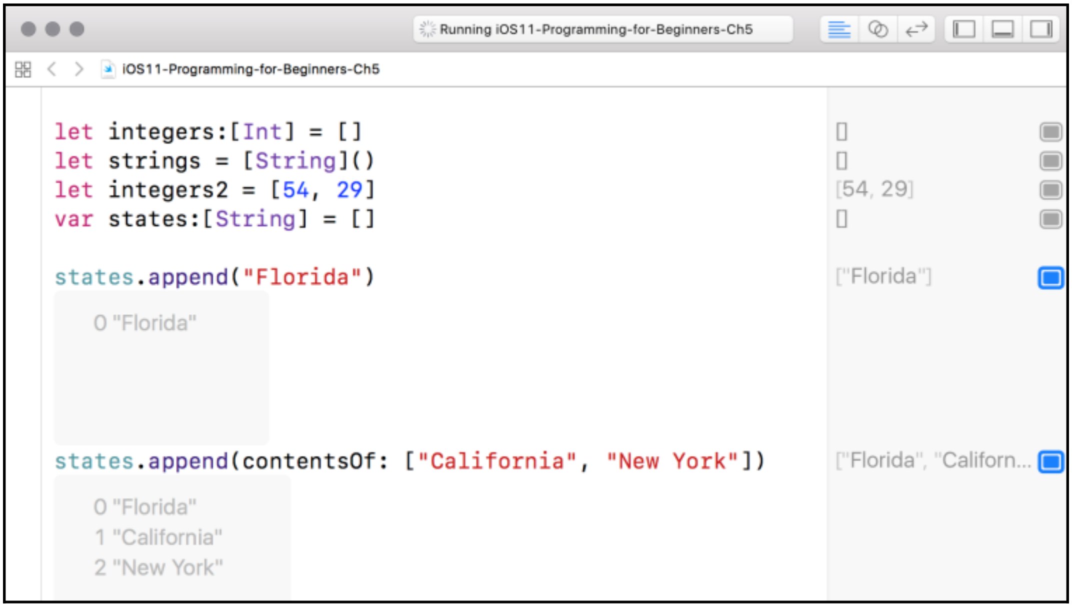
```
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []

states.append(23)
```

Cannot convert value of type 'Int' to expected argument type 'String'

```
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)
              ^~
```

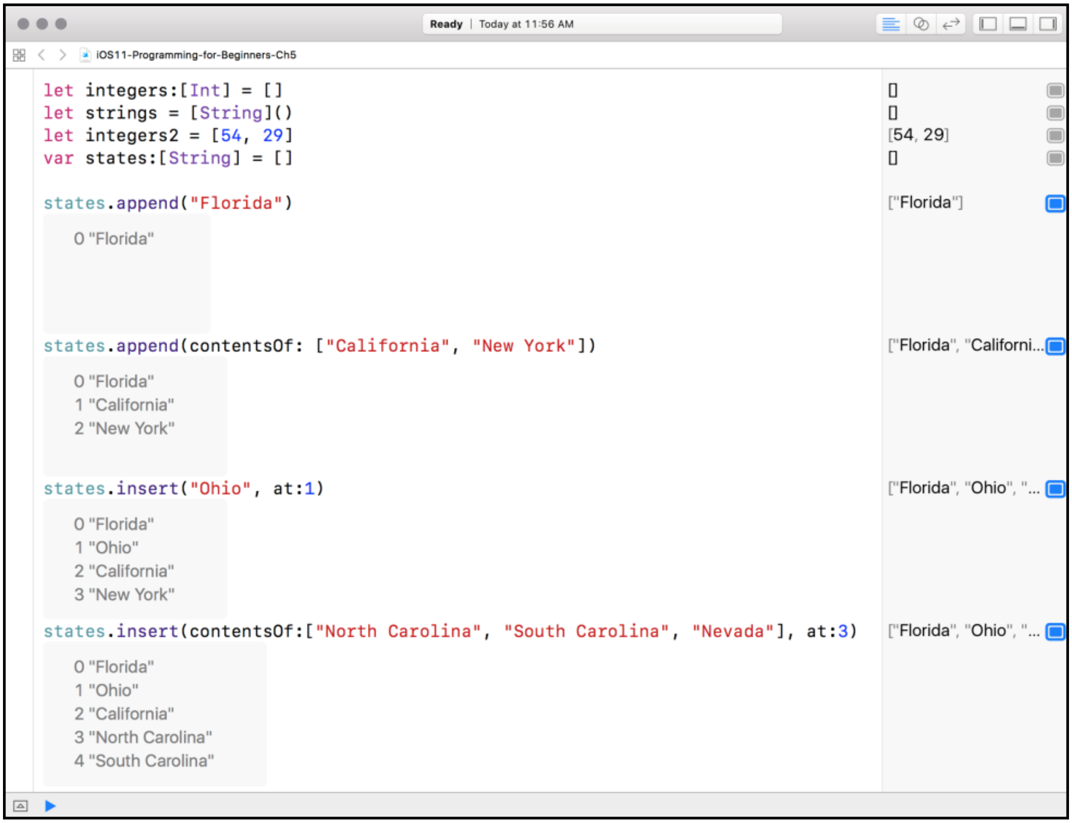

Ready | Today at 11:55 AM
iOS11-Programming-for-Beginners-Ch5
[]
[]
[54, 29]
[]
["Florida"]
0 "Florida"

```
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []

states.append("Florida")
```

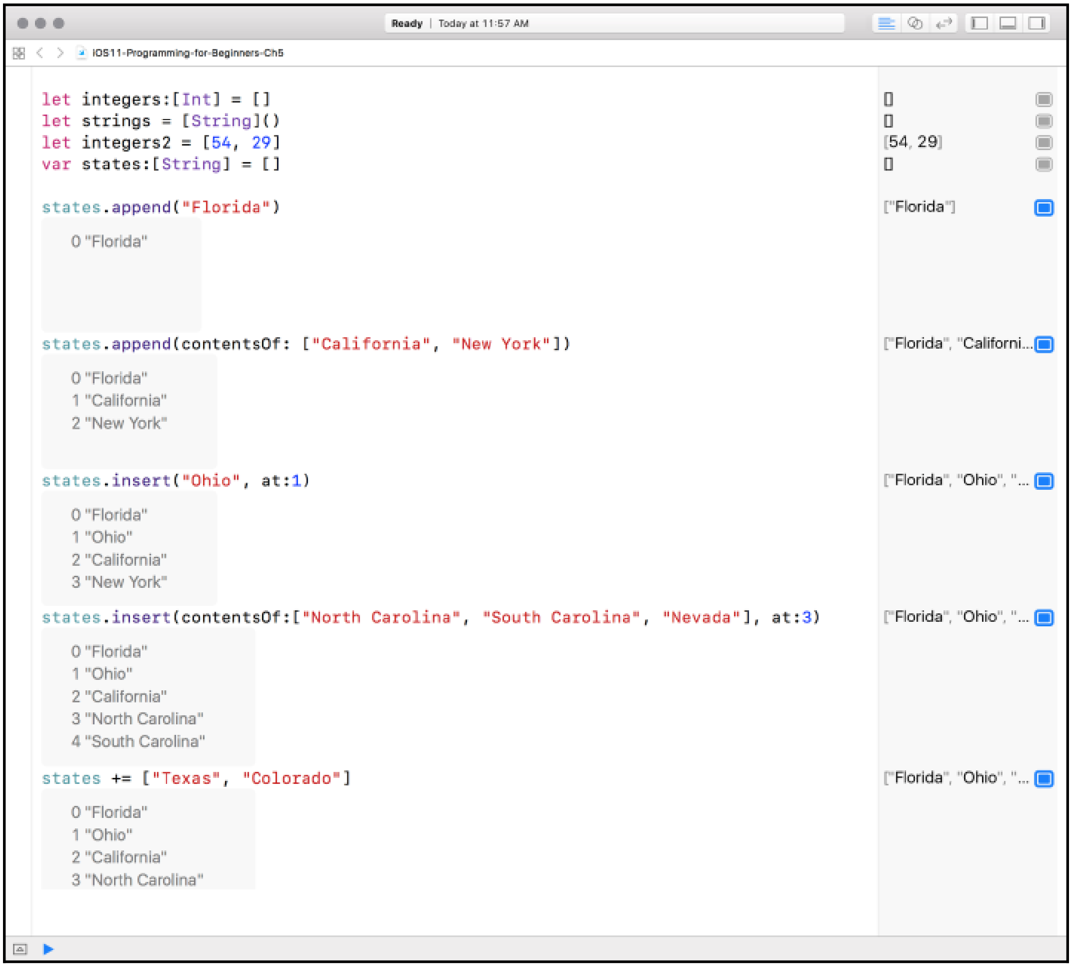


Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5

```
let integers:[Int] = []                                   []
let strings = [String]()                                  []
let integers2 = [54, 29]                                  [54, 29]
var states:[String] = []                                  []

states.append("Florida")                                  ["Florida"]
    0 "Florida"

states.append(contentsOf: ["California", "New York"])     ["Florida", "Californ...
    0 "Florida"
    1 "California"
    2 "New York"
```

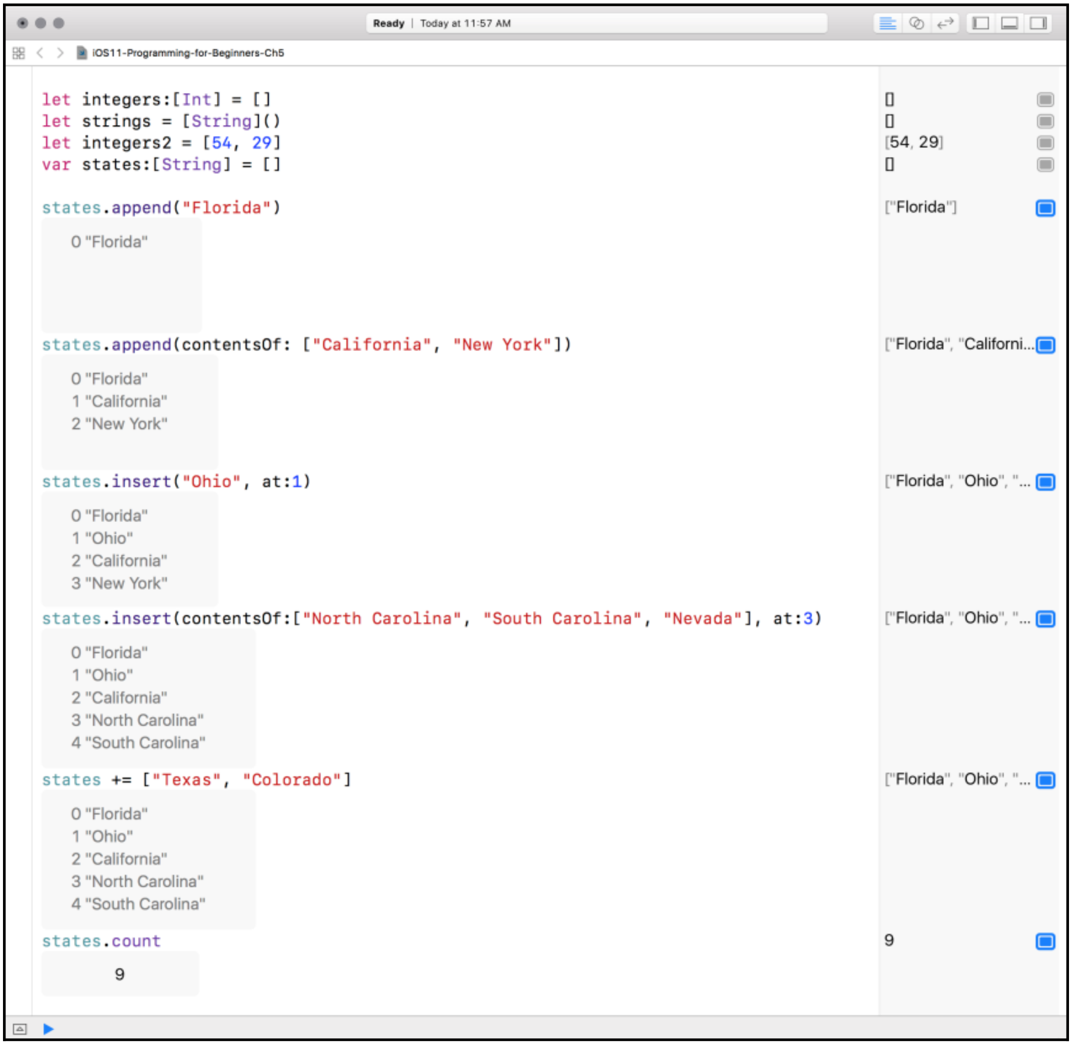

Ready | Today at 11:56 AM
iOS11-Programming-for-Beginners-Ch5

```
let integers:[Int] = []                                                    []
let strings = [String]()                                                   []
let integers2 = [54, 29]                                                   [54, 29]
var states:[String] = []                                                   []

states.append("Florida")                                                   ["Florida"]
    0 "Florida"

states.append(contentsOf: ["California", "New York"])                      ["Florida", "Californi...
    0 "Florida"
    1 "California"
    2 "New York"

states.insert("Ohio", at:1)                                                ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "New York"
states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)  ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "North Carolina"
    4 "South Carolina"
```



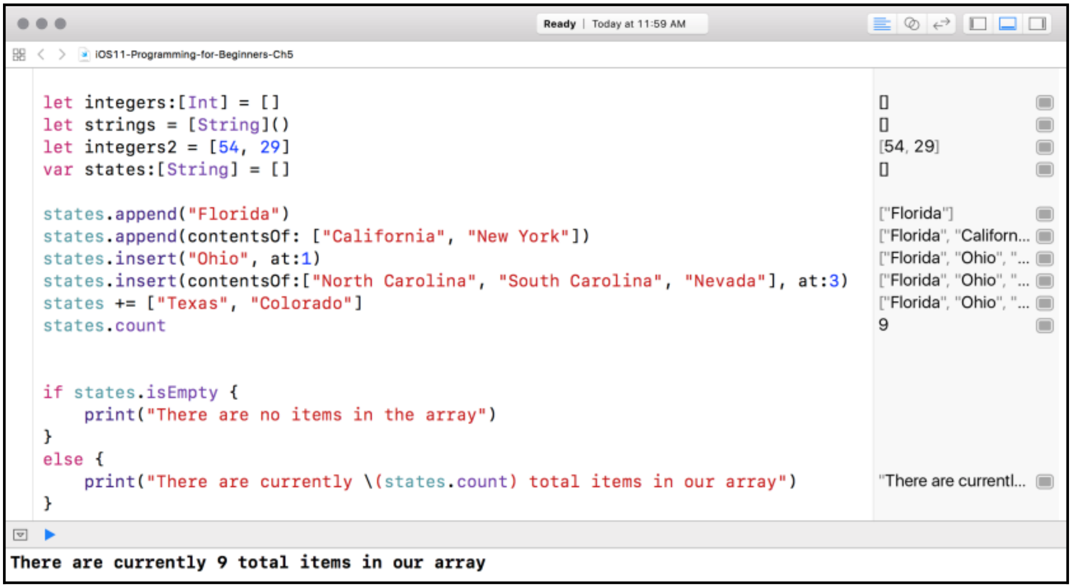
Ready | Today at 11:57 AM
iOS11-Programming-for-Beginners-Ch5

```
let integers:[Int] = []                                          []
let strings = [String]()                                         []
let integers2 = [54, 29]                                         [54, 29]
var states:[String] = []                                         []

states.append("Florida")                                         ["Florida"]
    0 "Florida"

states.append(contentsOf: ["California", "New York"])            ["Florida", "Californi...
    0 "Florida"
    1 "California"
    2 "New York"

states.insert("Ohio", at:1)                                      ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "New York"

states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)    ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "North Carolina"
    4 "South Carolina"

states += ["Texas", "Colorado"]                                  ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "North Carolina"
```



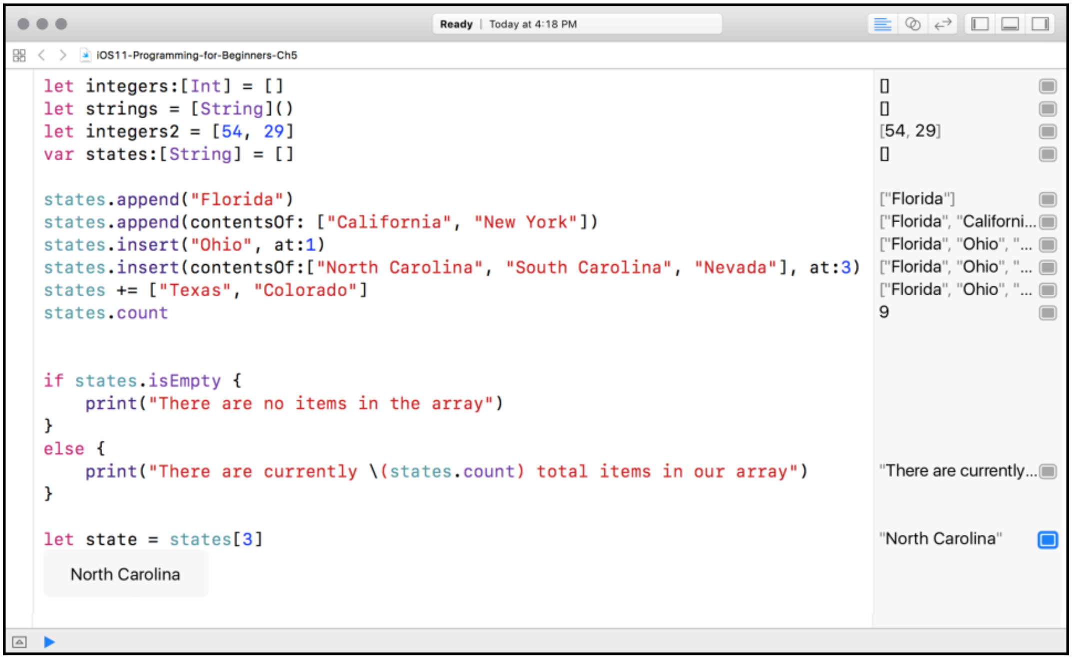
Ready | Today at 11:57 AM
iOS11-Programming-for-Beginners-Ch5

```
let integers:[Int] = []                                                           []
let strings = [String]()                                                          []
let integers2 = [54, 29]                                                          [54, 29]
var states:[String] = []                                                          []

states.append("Florida")                                                          ["Florida"]
    0 "Florida"

states.append(contentsOf: ["California", "New York"])                             ["Florida", "Californi...
    0 "Florida"
    1 "California"
    2 "New York"

states.insert("Ohio", at:1)                                                       ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "New York"

states.insert(contentsOf:["North Carolina", "South Carolina", "Nevada"], at:3)    ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "North Carolina"
    4 "South Carolina"

states += ["Texas", "Colorado"]                                                   ["Florida", "Ohio", "...
    0 "Florida"
    1 "Ohio"
    2 "California"
    3 "North Carolina"
    4 "South Carolina"

states.count                                                                      9
        9
```



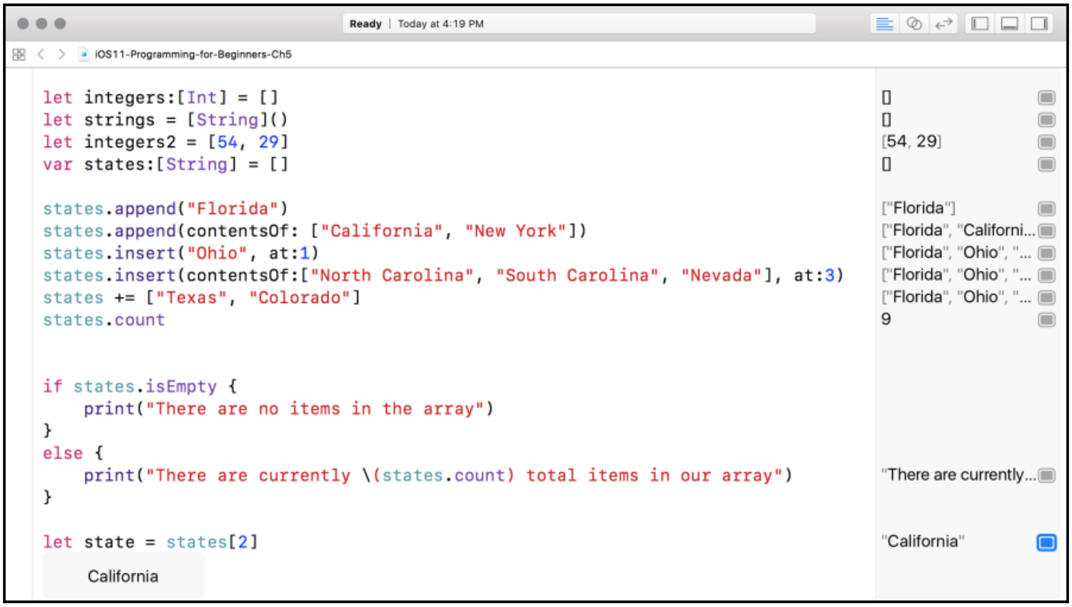
```swift
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []

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")
}
```

```
There are currently 9 total items in our array
```



```swift
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]
```

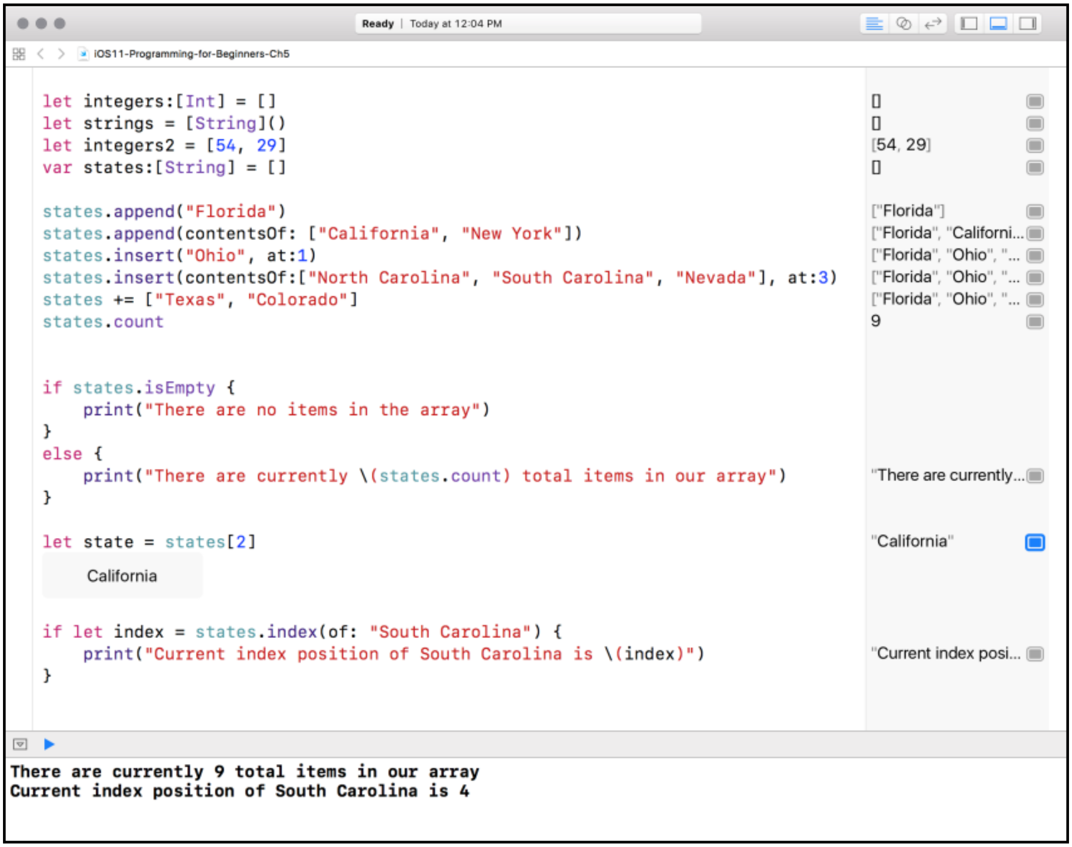
North Carolina



```swift
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]
```

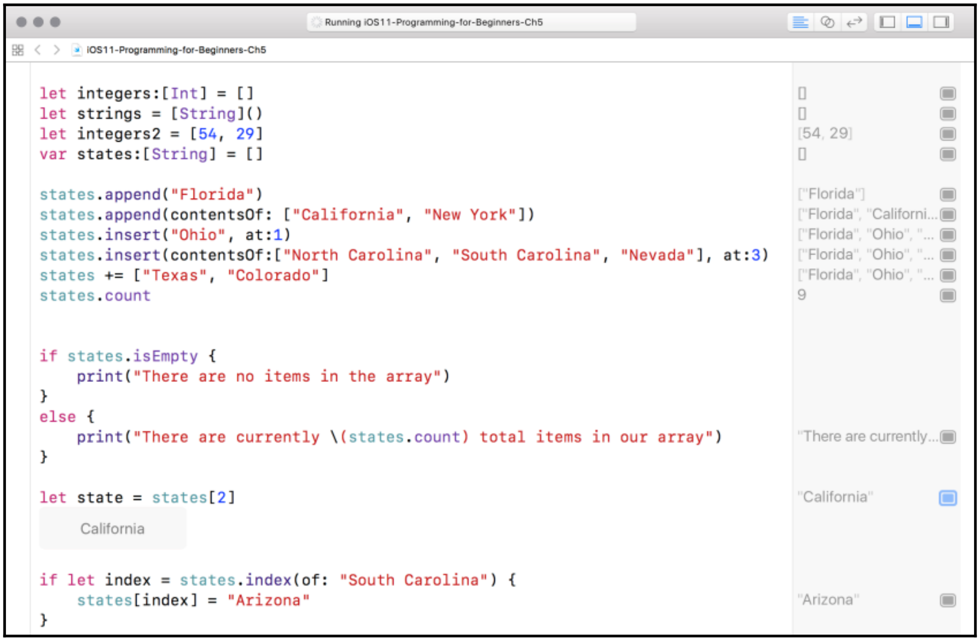
```
[]
[]
[54, 29]
[]

["Florida"]
["Florida", "Californi...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
["Florida", "Ohio", "...
9

"There are currently...

"California"
```

```
California
```

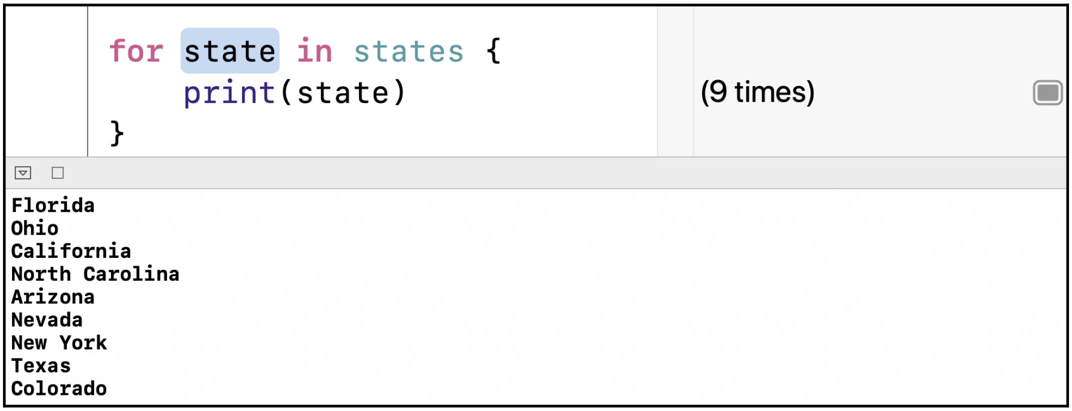


```
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []

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]

if let index = states.index(of: "South Carolina") {
    print("Current index position of South Carolina is \(index)")
}
```

```
There are currently 9 total items in our array
Current index position of South Carolina is 4
```

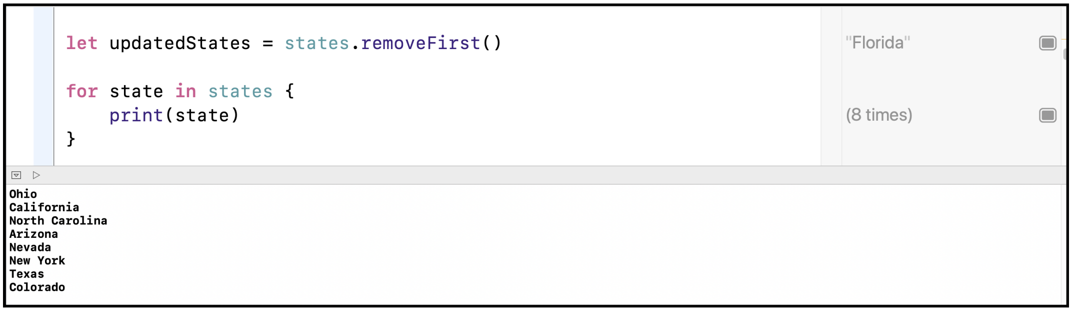


```
let integers:[Int] = []
let strings = [String]()
let integers2 = [54, 29]
var states:[String] = []

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]

if let index = states.index(of: "South Carolina") {
    states[index] = "Arizona"
}
```



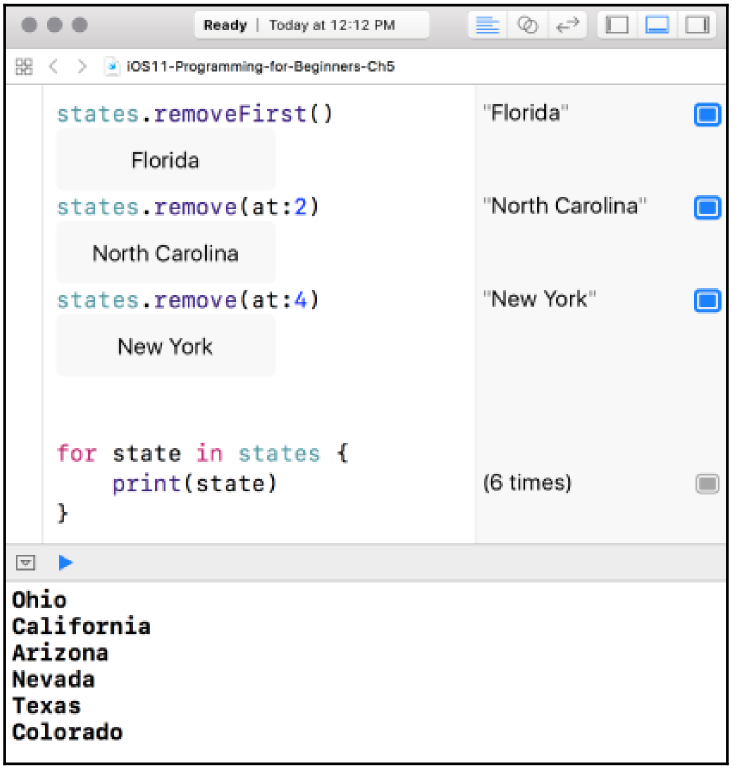
```
for state in states {
    print(state)
}
```

```
Florida
Ohio
California
North Carolina
Arizona
Nevada
New York
Texas
Colorado
```



```
let updatedStates = states.removeFirst()

for state in states {
    print(state)
}
```

```
Ohio
California
North Carolina
Arizona
Nevada
New York
Texas
Colorado
```

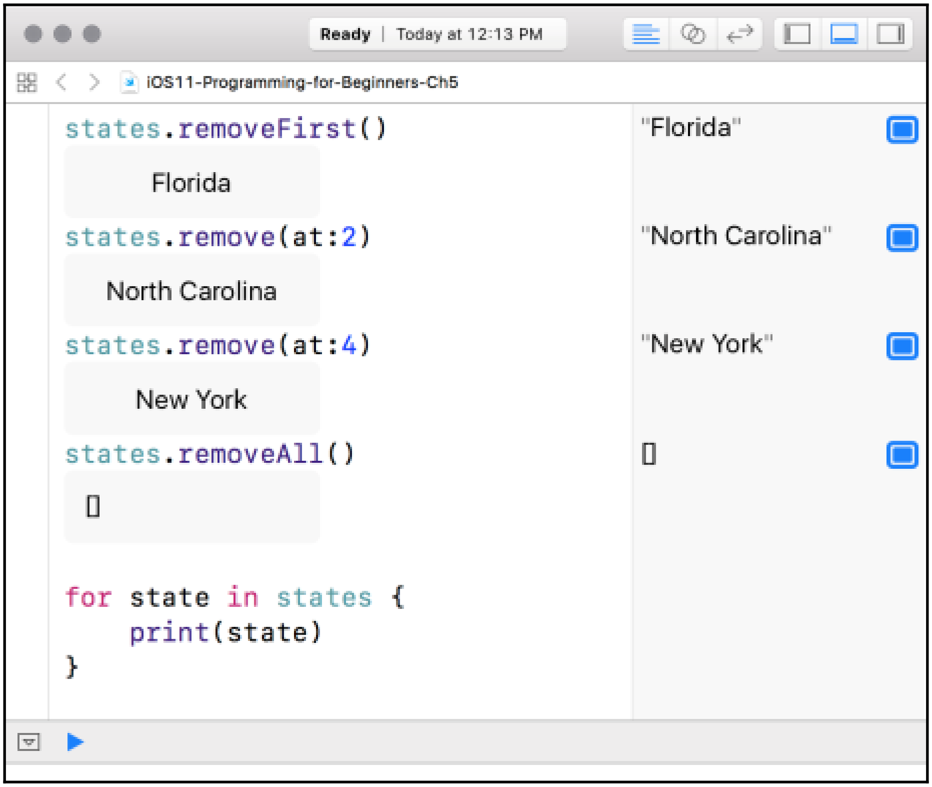


Ready | Today at 12:12 PM
iOS11-Programming-for-Beginners-Ch5

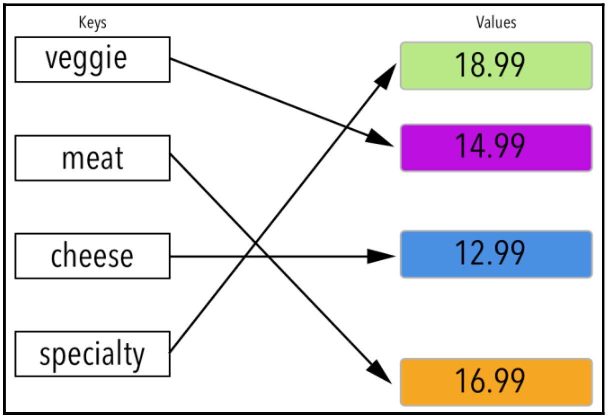
```
states.removeFirst()
    Florida
states.remove(at:2)
    North Carolina
states.remove(at:4)
    New York


for state in states {
    print(state)
}
```

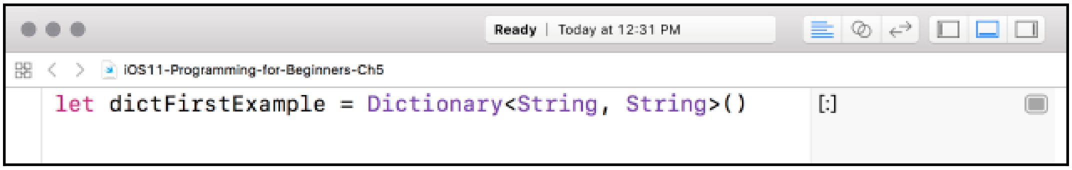
"Florida"
"North Carolina"
"New York"
(6 times)

```
Ohio
California
Arizona
Nevada
Texas
Colorado
```


Ready | Today at 12:13 PM
iOS11-Programming-for-Beginners-Ch5

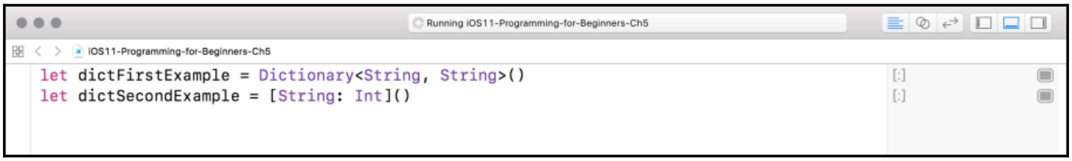
```
states.removeFirst()                    "Florida"
        Florida
states.remove(at:2)                     "North Carolina"
    North Carolina
states.remove(at:4)                     "New York"
       New York
states.removeAll()                      []
  []

for state in states {
    print(state)
}
```




Ready | Today at 12:31 PM
iOS11-Programming-for-Beginners-Ch5

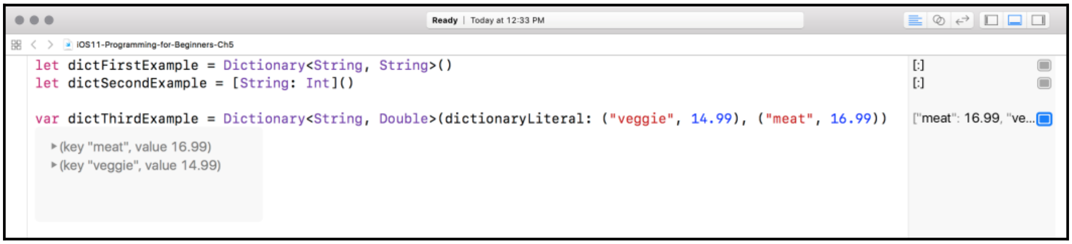
```swift
let dictFirstExample = Dictionary<String, String>()
```

```
[:]
```


Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5

```swift
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()
```

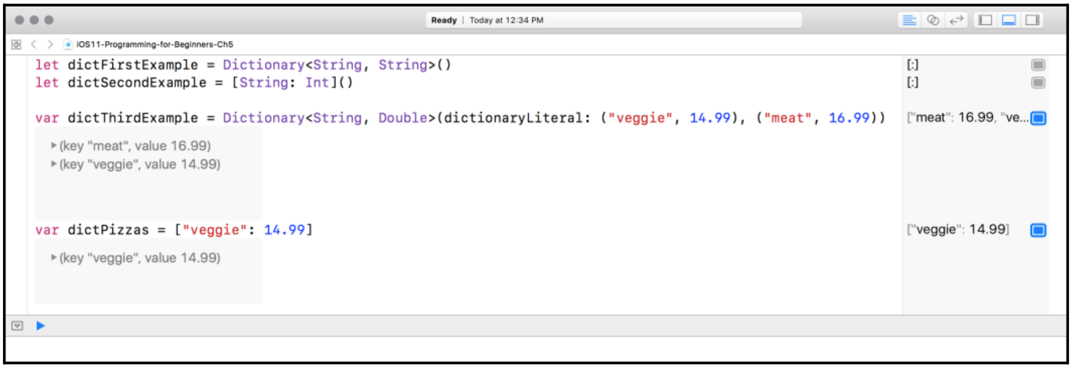
```
[:]
[:]
```


Ready | Today at 12:33 PM
iOS11-Programming-for-Beginners-Ch5

```swift
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()

var dictThirdExample = Dictionary<String, Double>(dictionaryLiteral: ("veggie", 14.99), ("meat", 16.99))
```

```
[:]
[:]

["meat": 16.99, "ve...
```

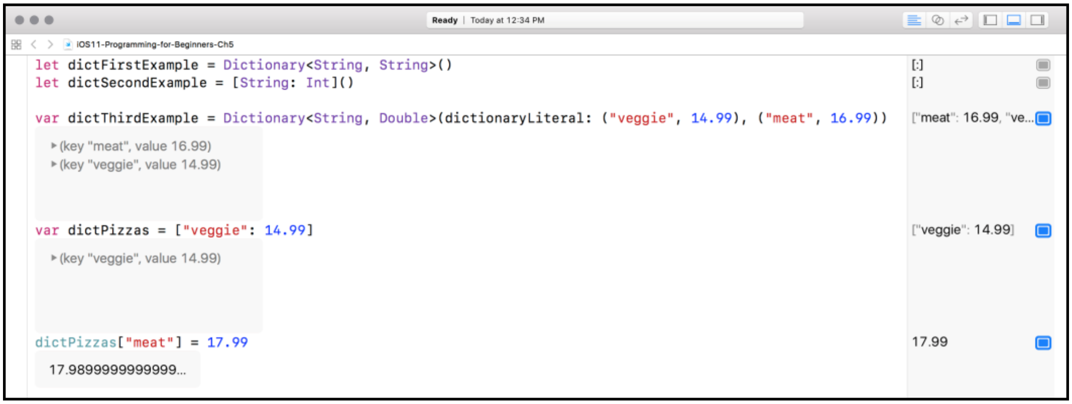
```
▸ (key "meat", value 16.99)
▸ (key "veggie", value 14.99)
```


Ready | Today at 12:34 PM
iOS11-Programming-for-Beginners-Ch5

```swift
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]
```

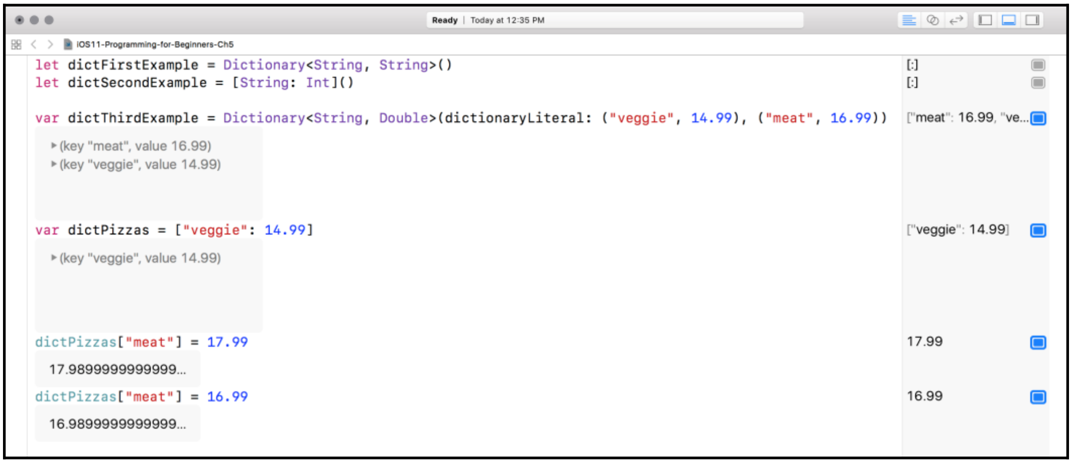
```
[:]
[:]

["meat": 16.99, "ve...

["veggie": 14.99]
```

```
▸ (key "meat", value 16.99)
▸ (key "veggie", value 14.99)

▸ (key "veggie", value 14.99)
```


Ready | Today at 12:34 PM
iOS11-Programming-for-Beginners-Ch5

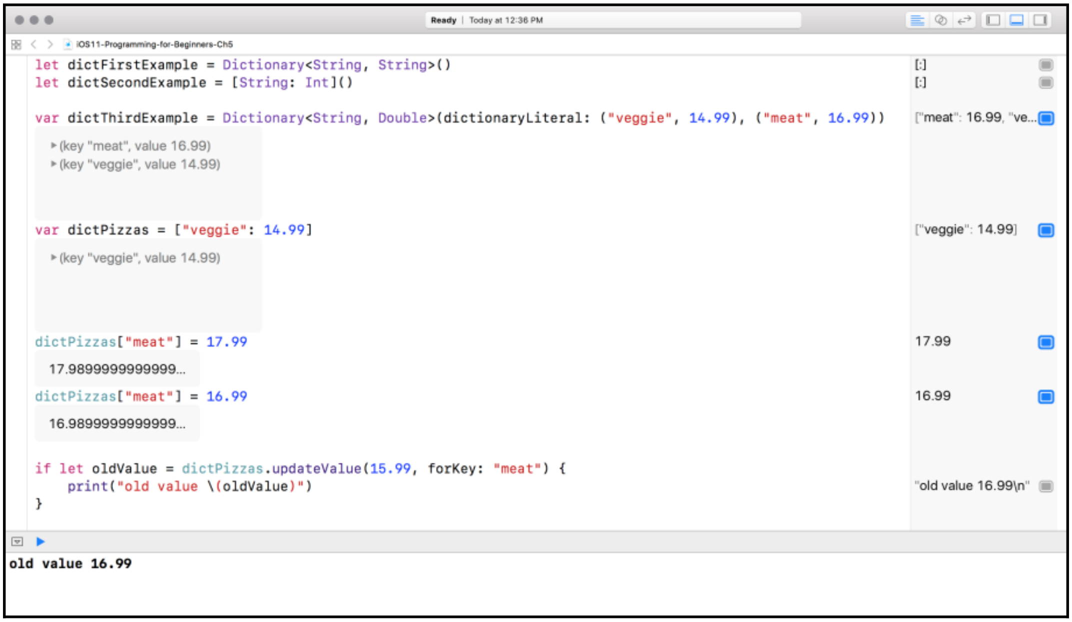
```
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()

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...
```

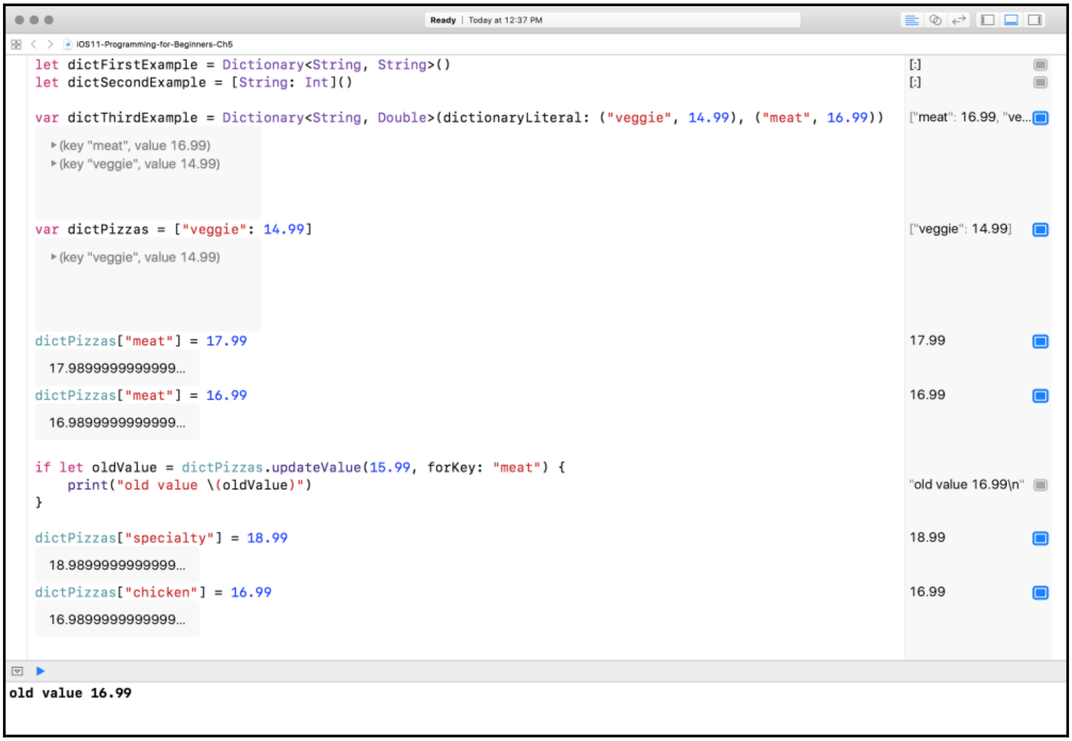
```
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
```


Ready | Today at 12:36 PM
iOS11-Programming-for-Beginners-Ch5
▸(key "meat", value 16.99)
▸(key "veggie", value 14.99)
▸(key "veggie", value 14.99)
17.9899999999999...
16.9899999999999...

```
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)")
}
```

```
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
```

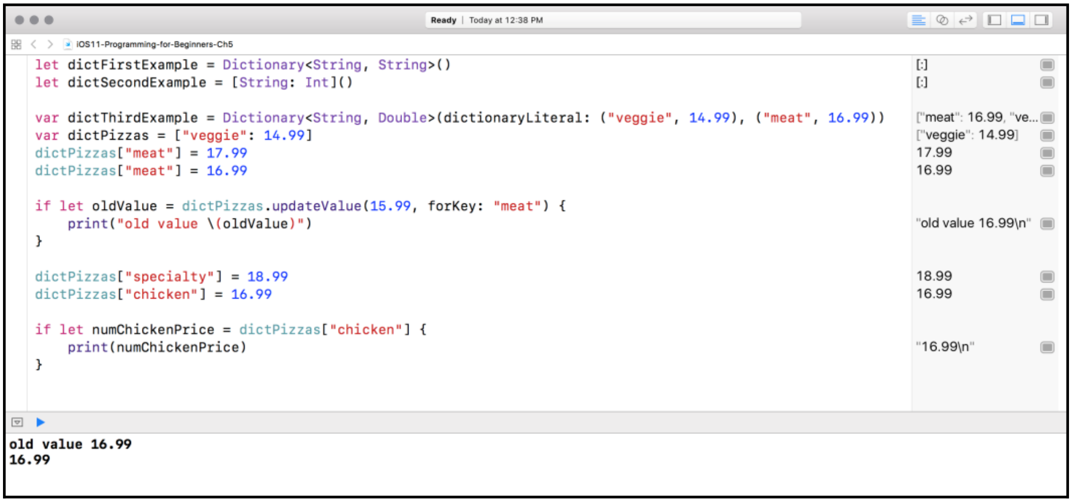
```
old value 16.99
```


Ready | Today at 12:37 PM
iOS11-Programming-for-Beginners-Ch5
▸(key "meat", value 16.99)
▸(key "veggie", value 14.99)
▸(key "veggie", value 14.99)
17.9899999999999...
16.9899999999999...
18.9899999999999...
16.9899999999999...

```
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
```

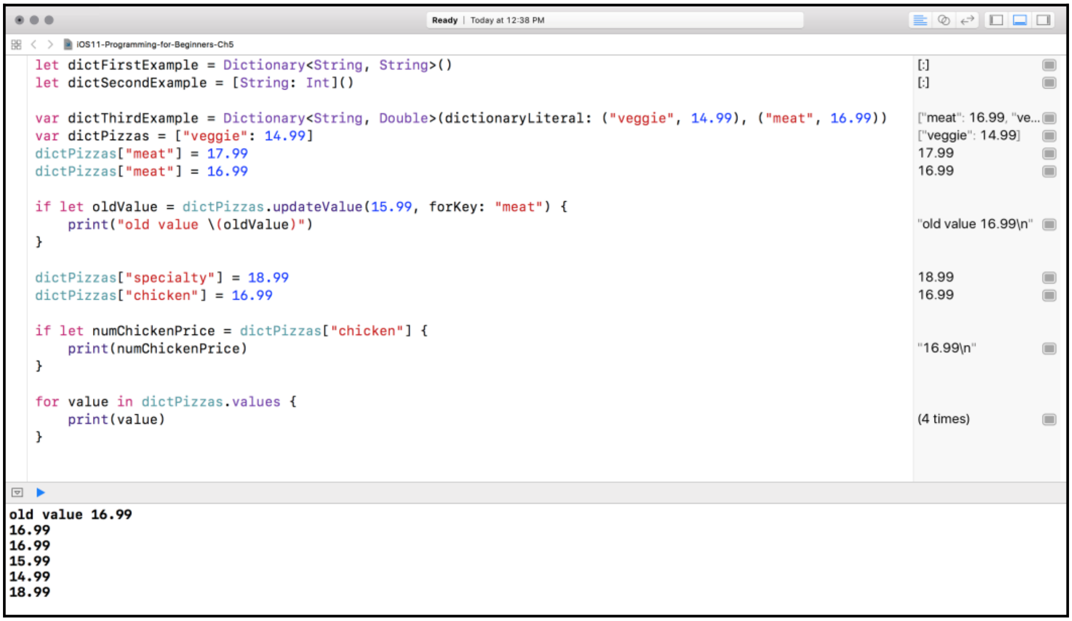
```
[:]
[:]
["meat": 16.99, "ve...
["veggie": 14.99]
17.99
16.99
"old value 16.99\n"
18.99
16.99
```

```
old value 16.99
```



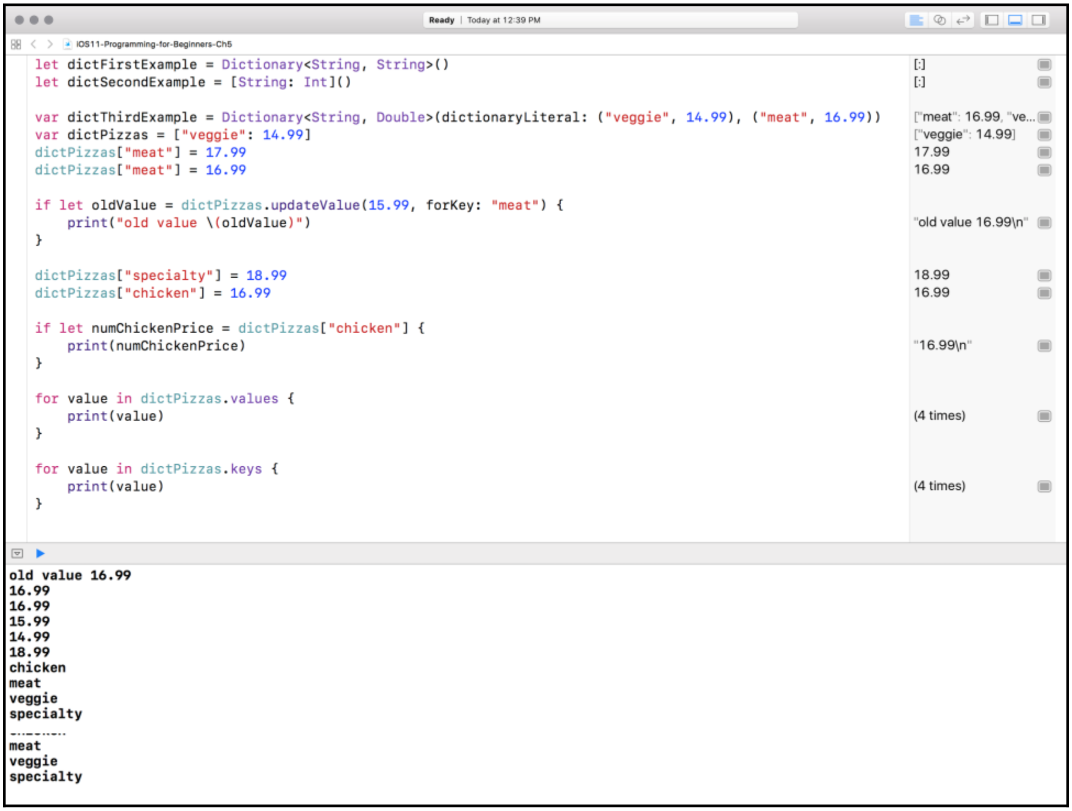
```
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()

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)
}
```

```
old value 16.99
16.99
```



```
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)
}
```

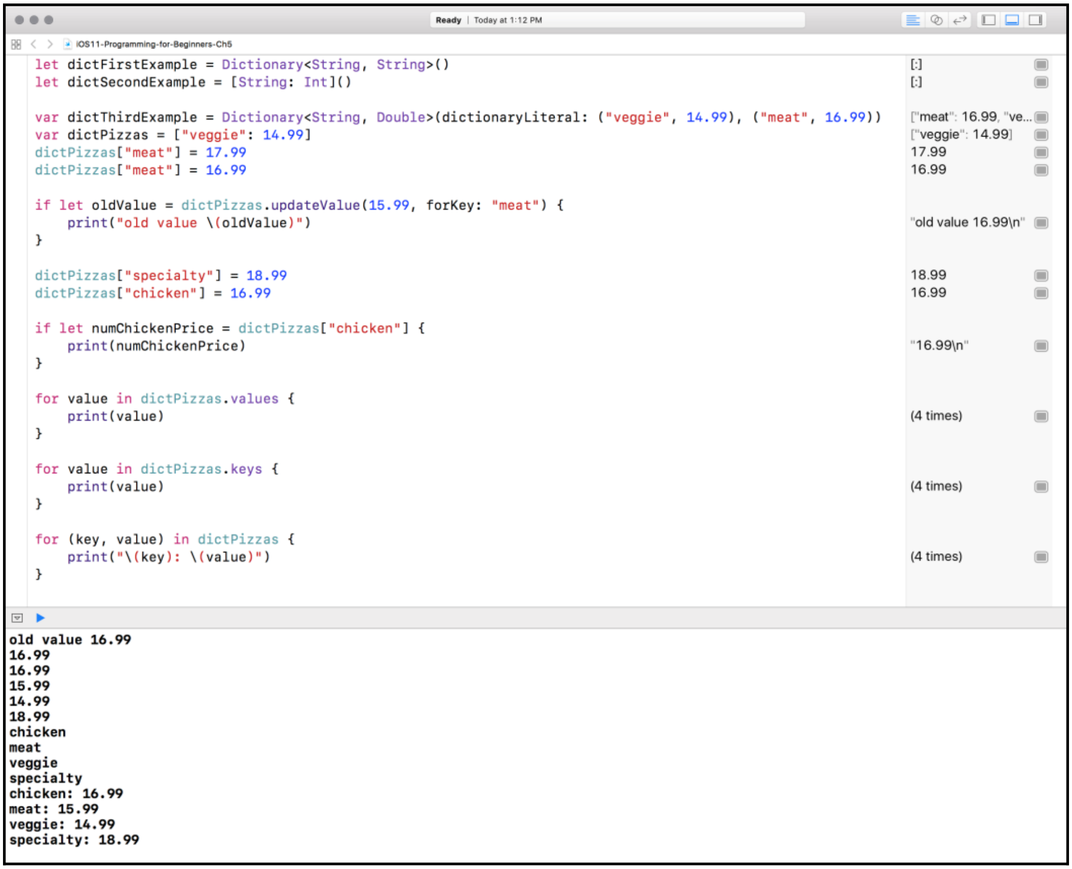
```
old value 16.99
16.99
16.99
15.99
14.99
18.99
```


Ready | Today at 12:39 PM
iOS11-Programming-for-Beginners-Ch5

```swift
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)
}
```

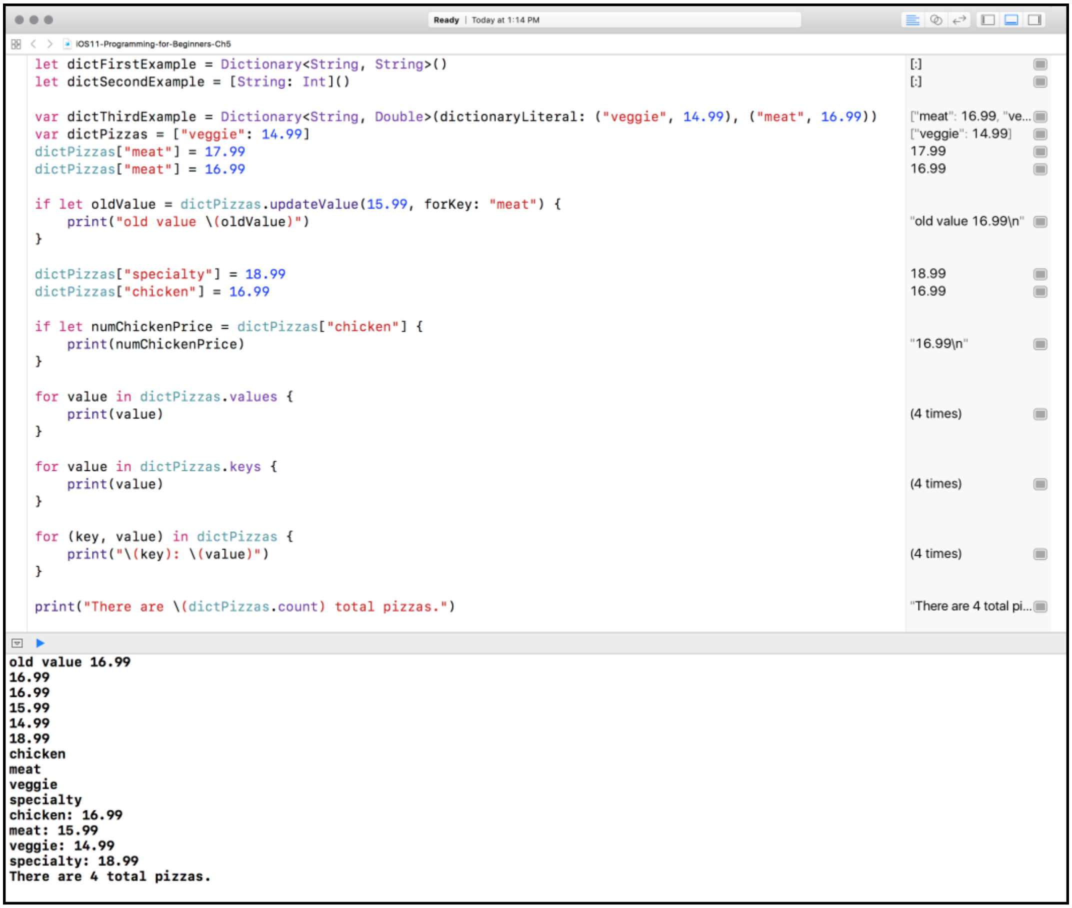
```
[:]
[:]

["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
chicken
meat
veggie
specialty
```



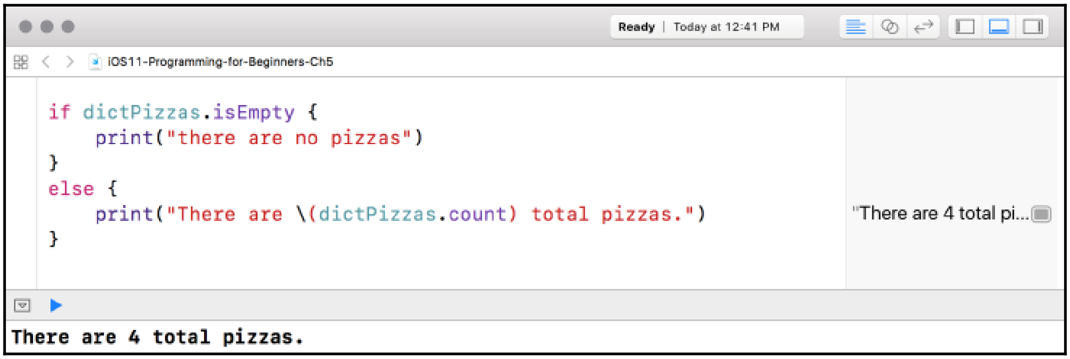
```
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()

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)")
}
```

```
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
```



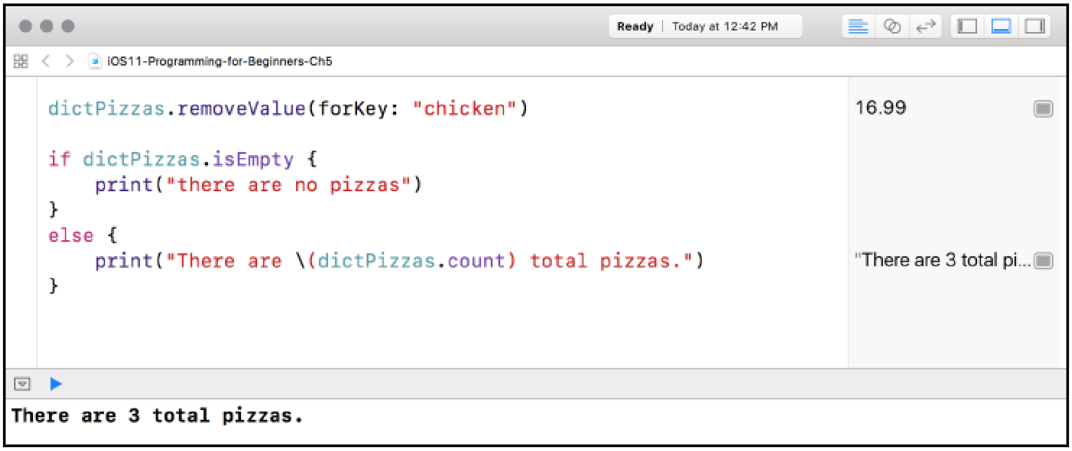
```swift
let dictFirstExample = Dictionary<String, String>()
let dictSecondExample = [String: Int]()

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.")
```

```
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
"There are 4 total pi...

```swift
if dictPizzas.isEmpty {
    print("there are no pizzas")
}
else {
    print("There are \(dictPizzas.count) total pizzas.")
}
```

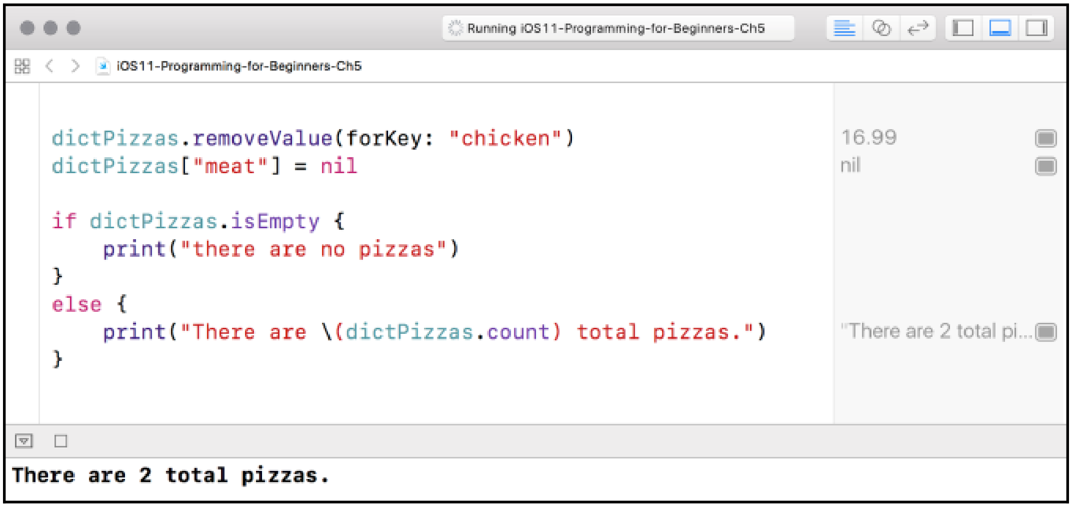
```
There are 4 total pizzas.
```


Ready | Today at 12:42 PM
iOS11-Programming-for-Beginners-Ch5

```swift
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...
```

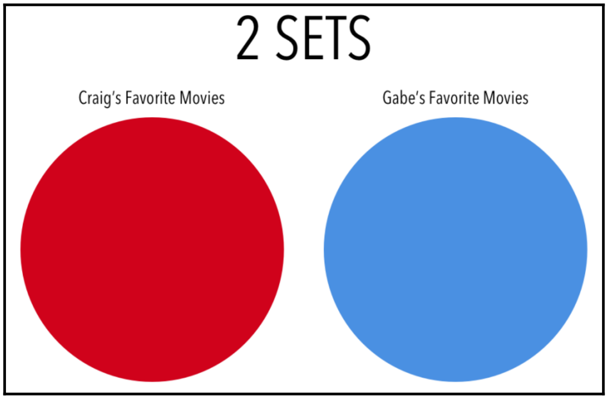
```
There are 3 total pizzas.
```


Running iOS11-Programming-for-Beginners-Ch5
iOS11-Programming-for-Beginners-Ch5

```swift
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...
```

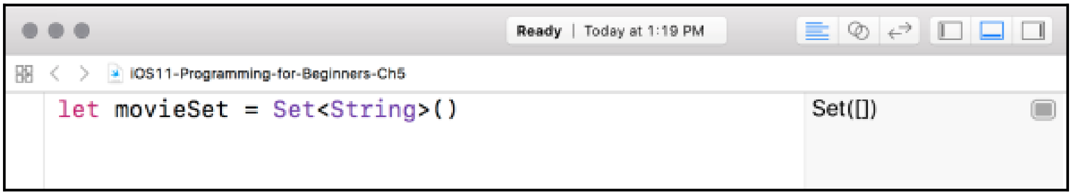
```
There are 2 total pizzas.
```



2 SETS

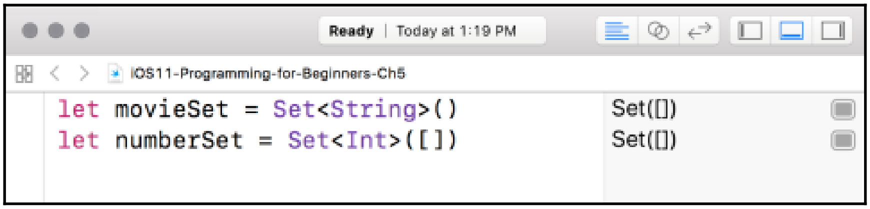
Craig's Favorite Movies

Gabe's Favorite Movies



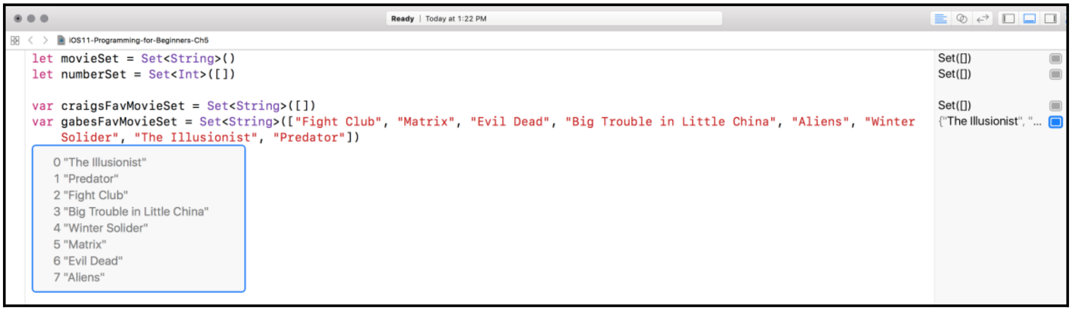
```
let movieSet = Set<String>()
```

```
Set([])
```



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

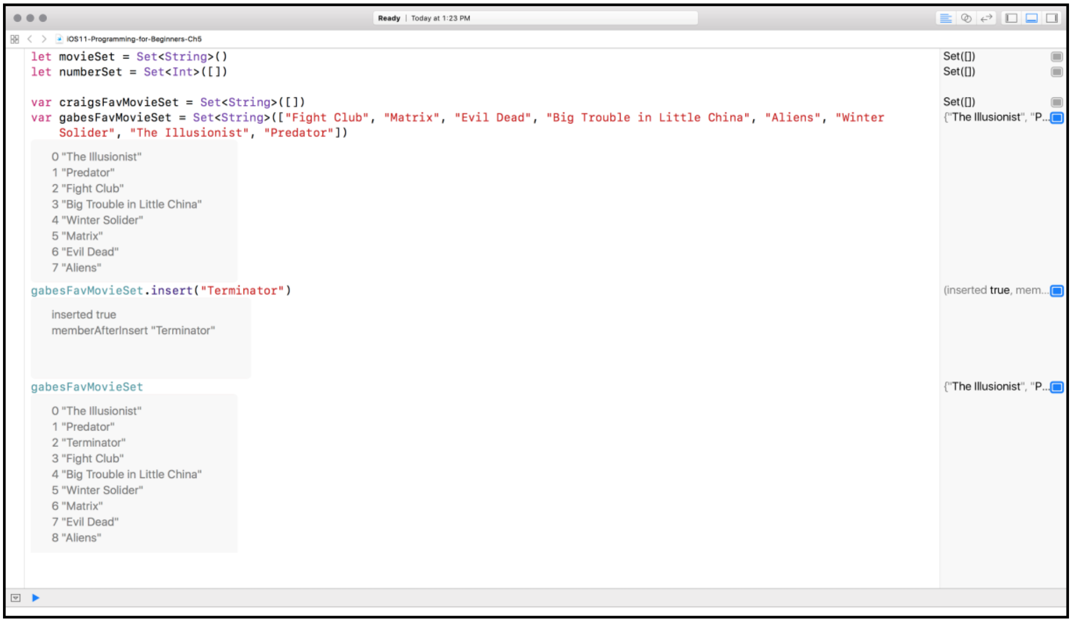
```
Set([])
Set([])
```



```
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"])
```

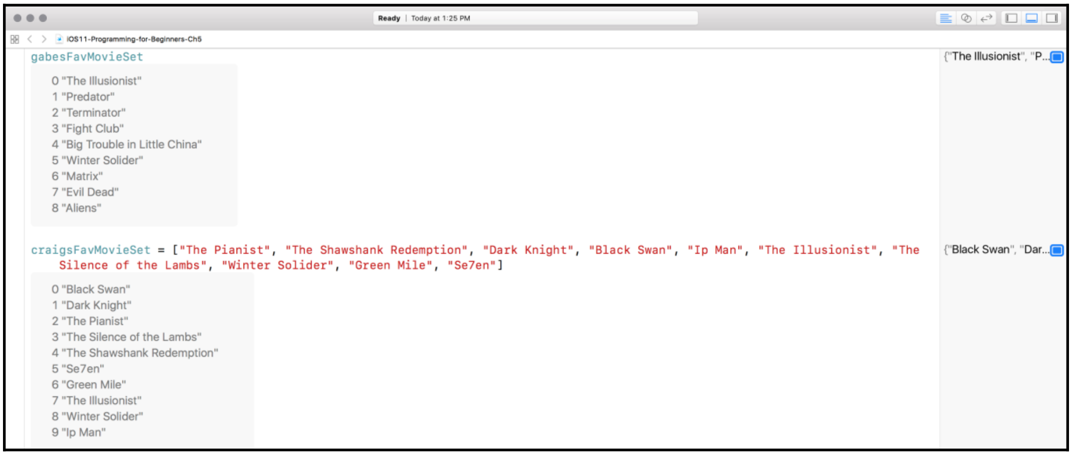
```
Set([])
Set([])

Set([])
{"The Illusionist", "...
```

```
0 "The Illusionist"
1 "Predator"
2 "Fight Club"
3 "Big Trouble in Little China"
4 "Winter Solider"
5 "Matrix"
6 "Evil Dead"
7 "Aliens"
```



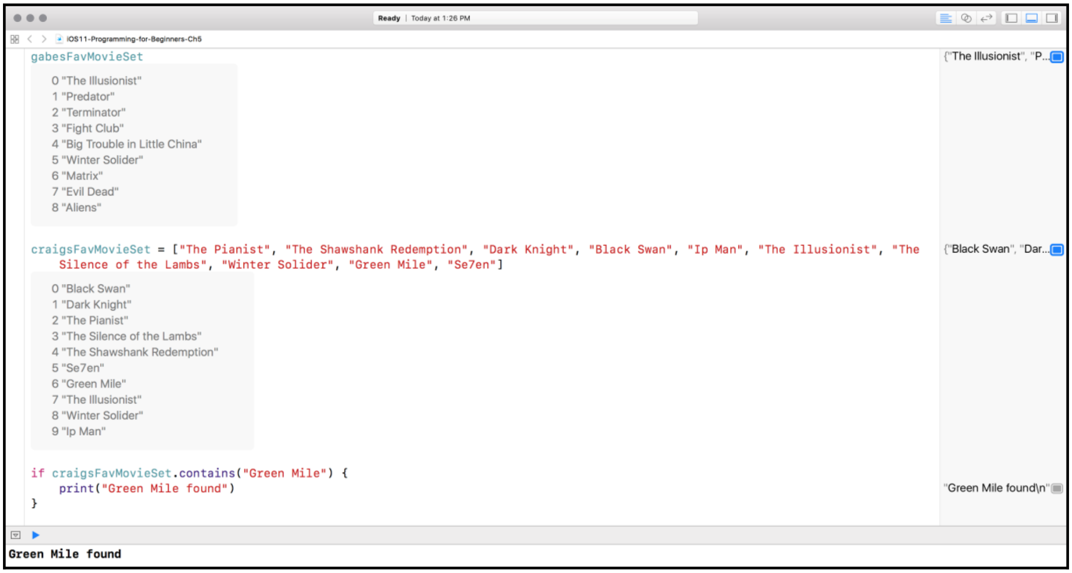
```
let movieSet = Set<String>()
let numberSet = Set<Int>([])

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...
```



```
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"

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"
```

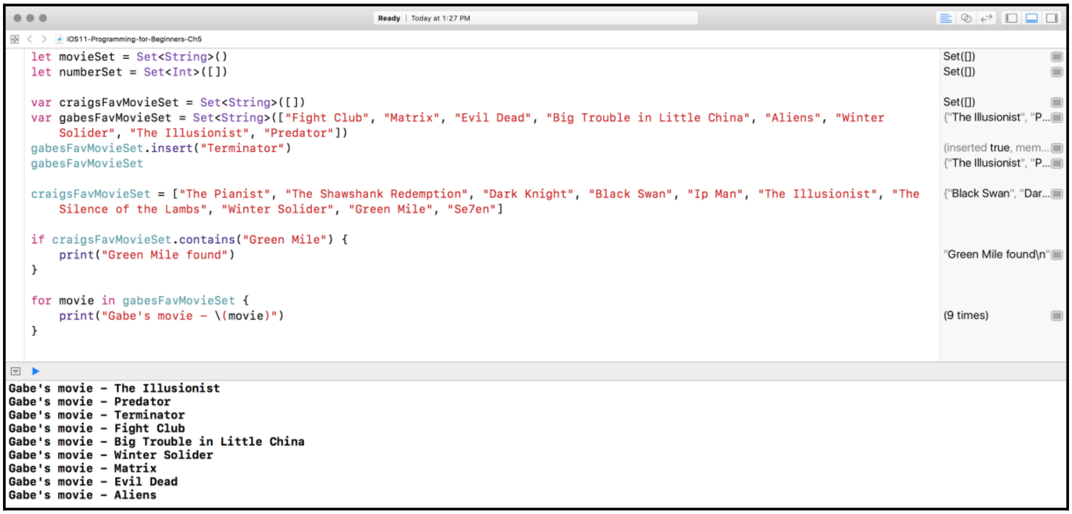
```
{"The Illusionist", "P...

{"Black Swan", "Dar...
```



```
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"

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"

if craigsFavMovieSet.contains("Green Mile") {
    print("Green Mile found")
}
```

```
{"The Illusionist", "P...
{"Black Swan", "Dar...
"Green Mile found\n"
```

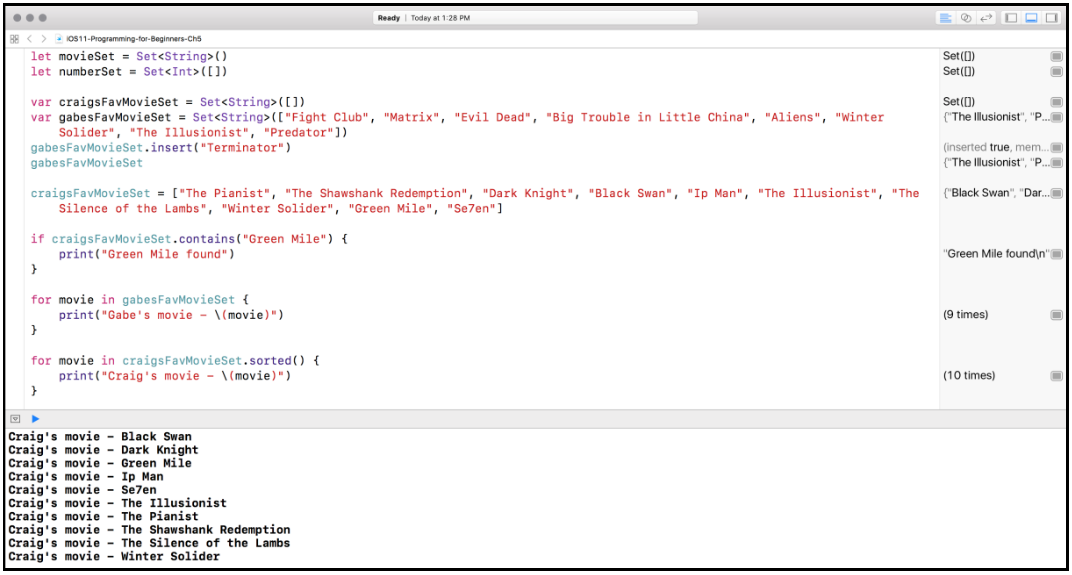
```
Green Mile found
```



```
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)
```

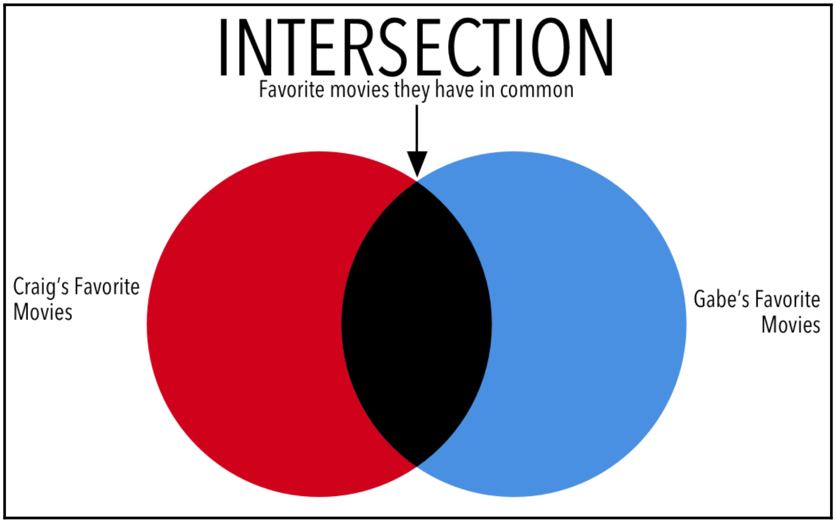
```
Gabe's movie - The Illusionist
Gabe's movie - Predator
Gabe's movie - Terminator
Gabe's movie - Fight Club
Gabe's movie - Big Trouble in Little China
Gabe's movie - Winter Solider
Gabe's movie - Matrix
Gabe's movie - Evil Dead
Gabe's movie - Aliens
```

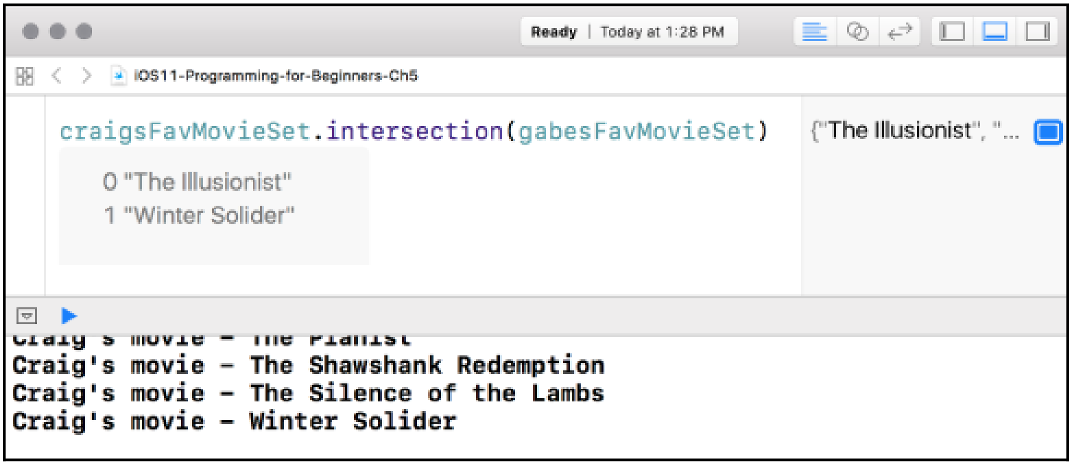


```
let movieSet = Set<String>()
let numberSet = Set<Int>([])

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
```

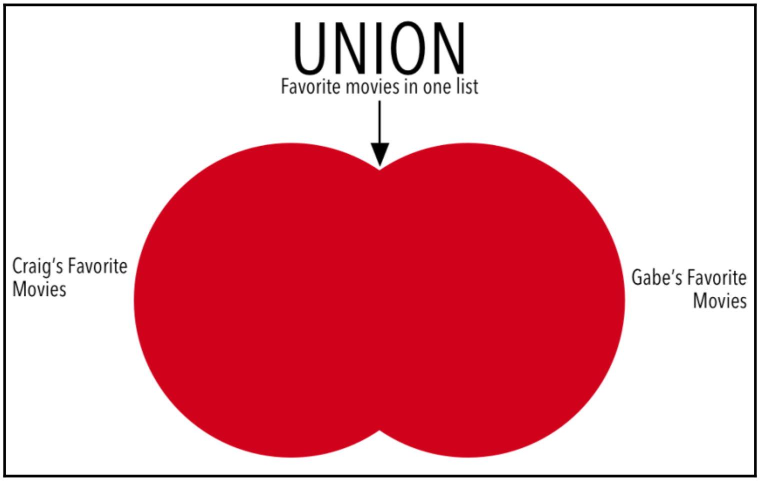


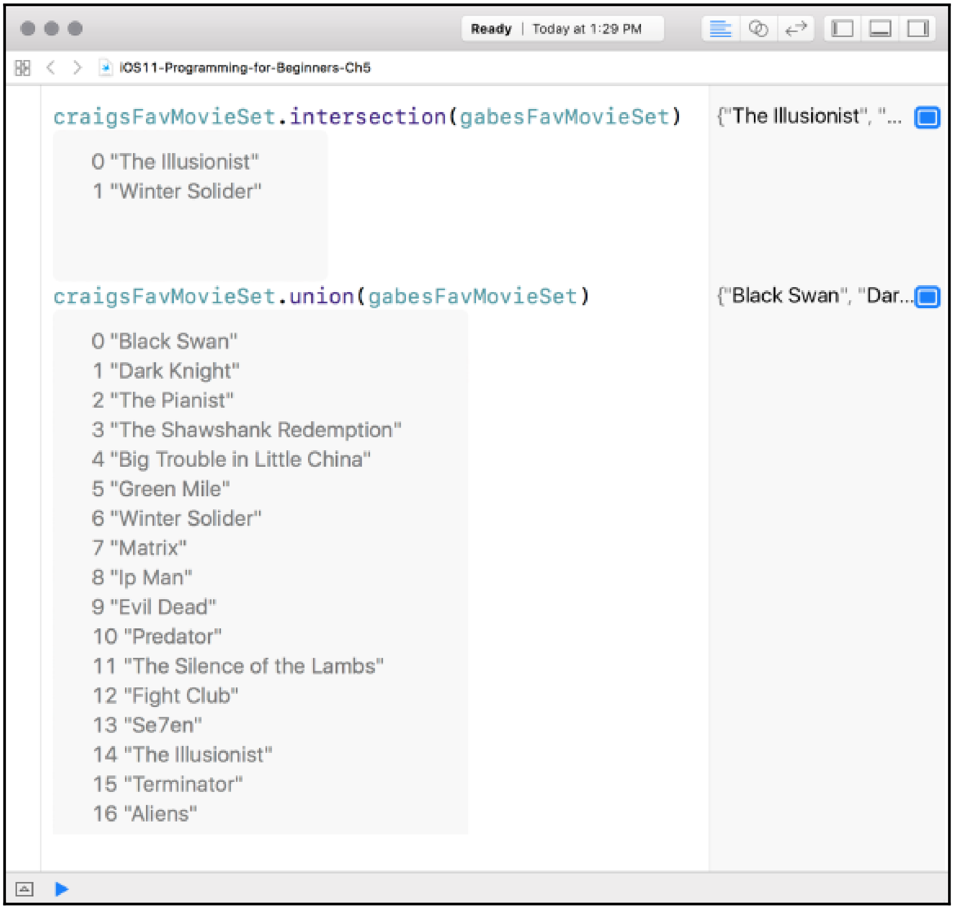


```
craigsFavMovieSet.intersection(gabesFavMovieSet)
    0 "The Illusionist"
    1 "Winter Solider"
```

```
{"The Illusionist", "...
```

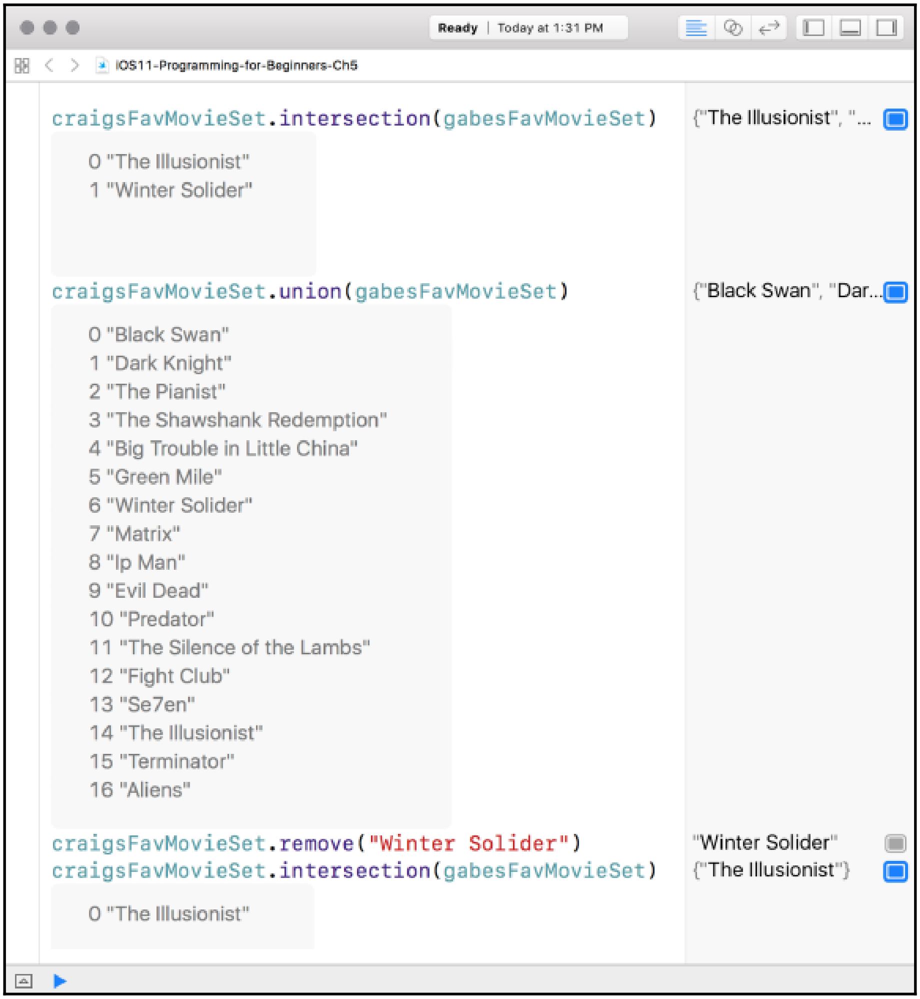
```
Craig's movie - The Pianist
Craig's movie - The Shawshank Redemption
Craig's movie - The Silence of the Lambs
Craig's movie - Winter Solider
```



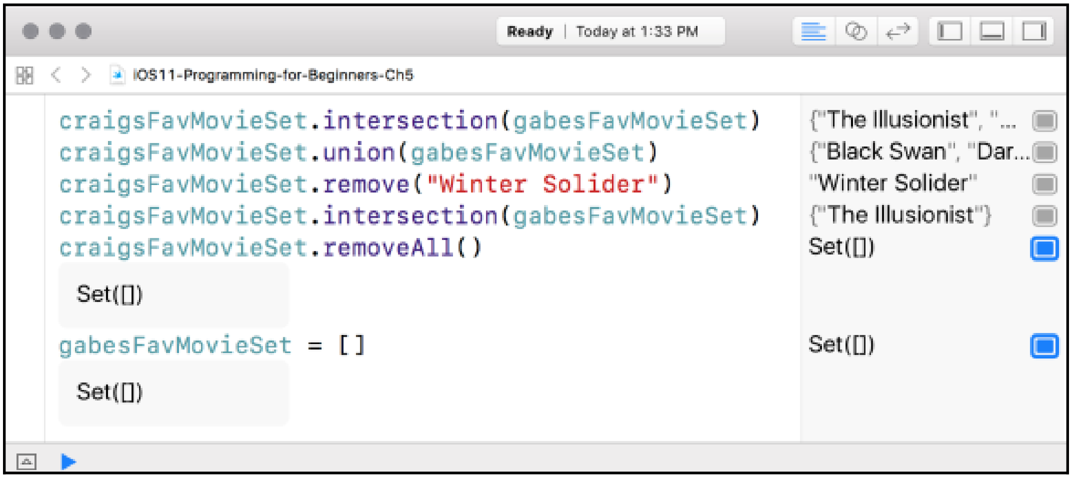


```
craigsFavMovieSet.intersection(gabesFavMovieSet)
    0 "The Illusionist"
    1 "Winter Solider"
craigsFavMovieSet.union(gabesFavMovieSet)
    0 "Black Swan"
    1 "Dark Knight"
    2 "The Pianist"
    3 "The Shawshank Redemption"
    4 "Big Trouble in Little China"
    5 "Green Mile"
    6 "Winter 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"
```

```
{"The Illusionist", "...
{"Black Swan", "Dar...
```

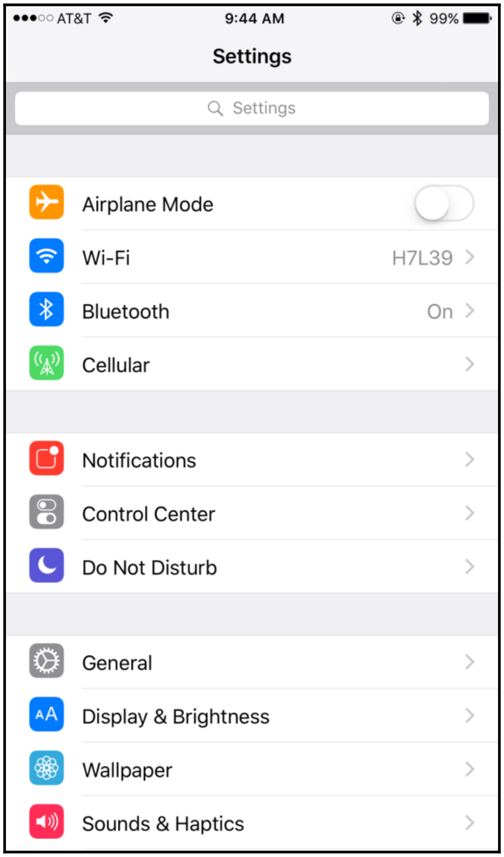


```
craigsFavMovieSet.intersection(gabesFavMovieSet)      {"The Illusionist", "...
    0 "The Illusionist"
    1 "Winter Solider"

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"
```



```
craigsFavMovieSet.intersection(gabesFavMovieSet)    {"The Illusionist", "...
craigsFavMovieSet.union(gabesFavMovieSet)           {"Black Swan", "Dar...
craigsFavMovieSet.remove("Winter Solider")          "Winter Solider"
craigsFavMovieSet.intersection(gabesFavMovieSet)    {"The Illusionist"}
craigsFavMovieSet.removeAll()                       Set([])
  Set([])
gabesFavMovieSet = []                               Set([])
  Set([])
```

Chapter 6: Starting the UI Setup



●●●○○ AT&T 9:44 AM 99%

Settings

Settings

Airplane Mode ○

Wi-Fi H7L39 >

Bluetooth On >

Cellular >

Notifications >

Control Center >

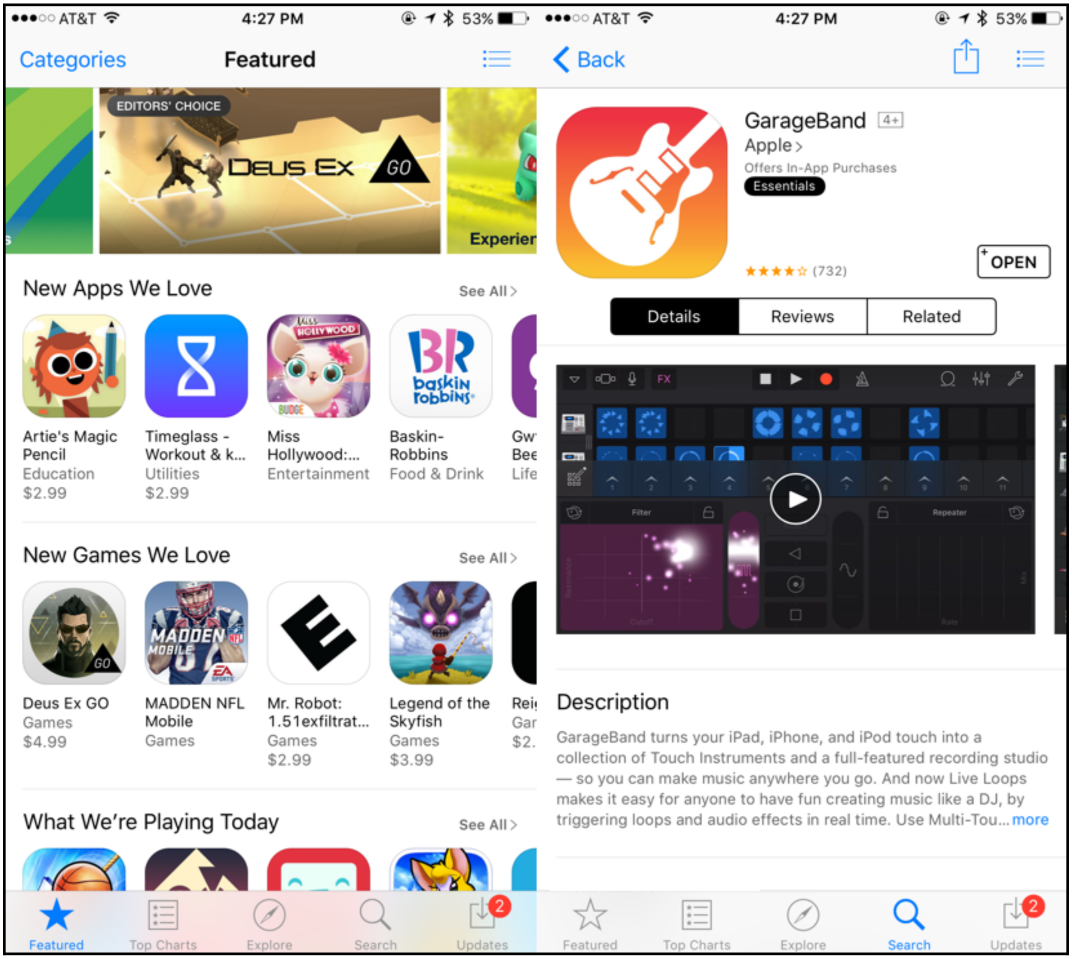
Do Not Disturb >

General >

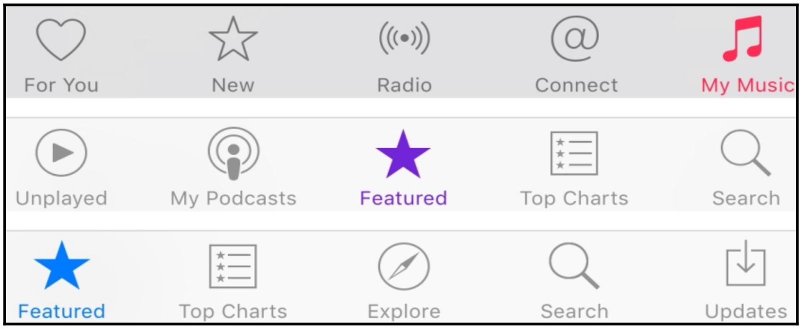
Display & Brightness >

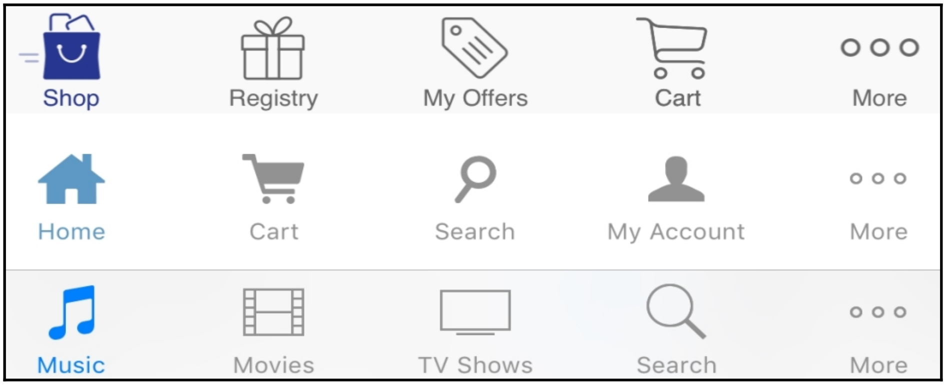
Wallpaper >

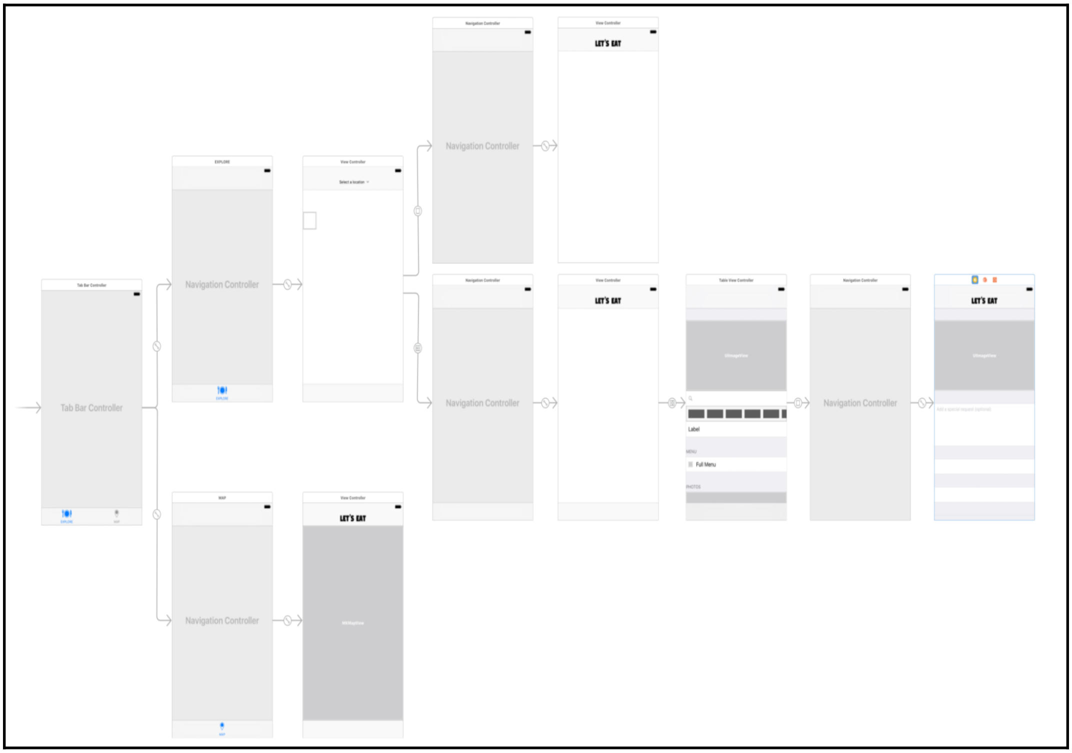
Sounds & Haptics >

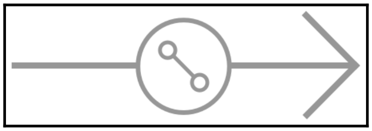


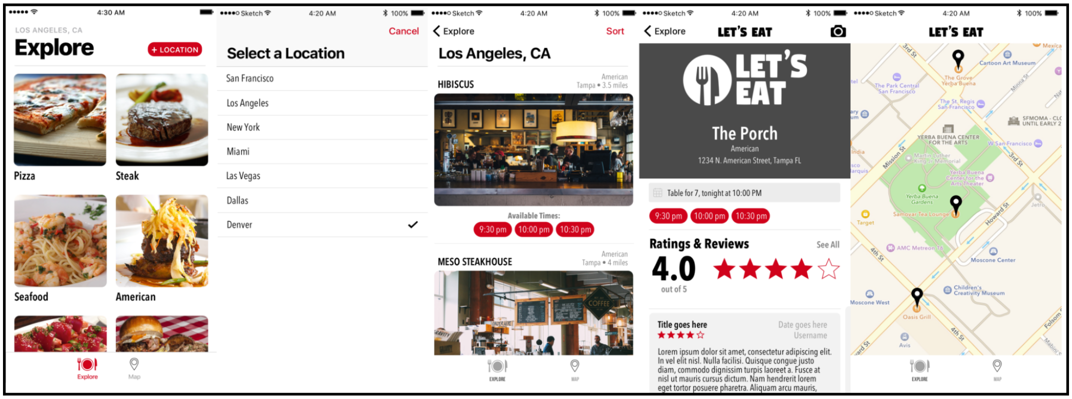
GarageBand turns your iPad, iPhone, and iPod touch into a collection of Touch Instruments and a full-featured recording studio — so you can make music anywhere you go. And now Live Loops makes it easy for anyone to have fun creating music like a DJ, by triggering loops and audio effects in real time. Use Multi-Tou... more



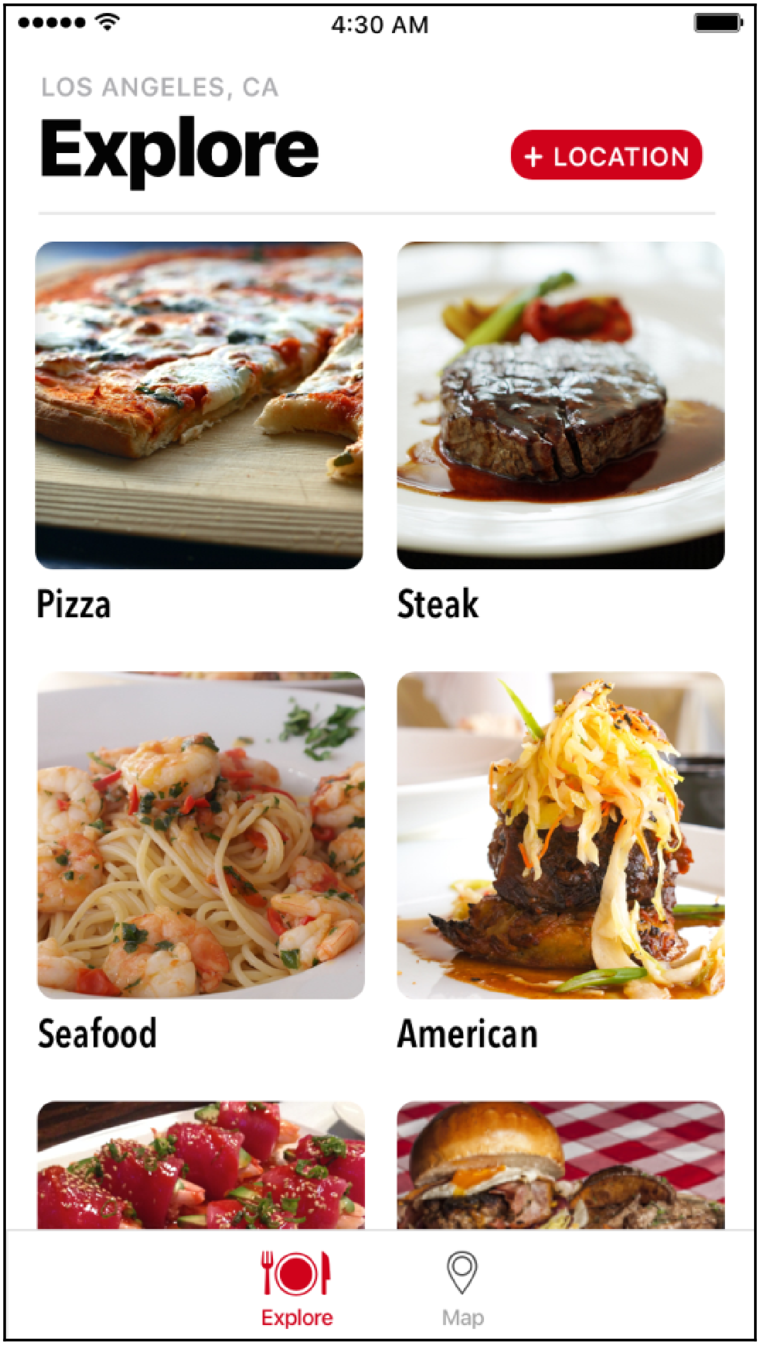

Shop
Registry
My Offers
Cart
More
Home
Cart
Search
My Account
More
Music
Movies
TV Shows
Search
More

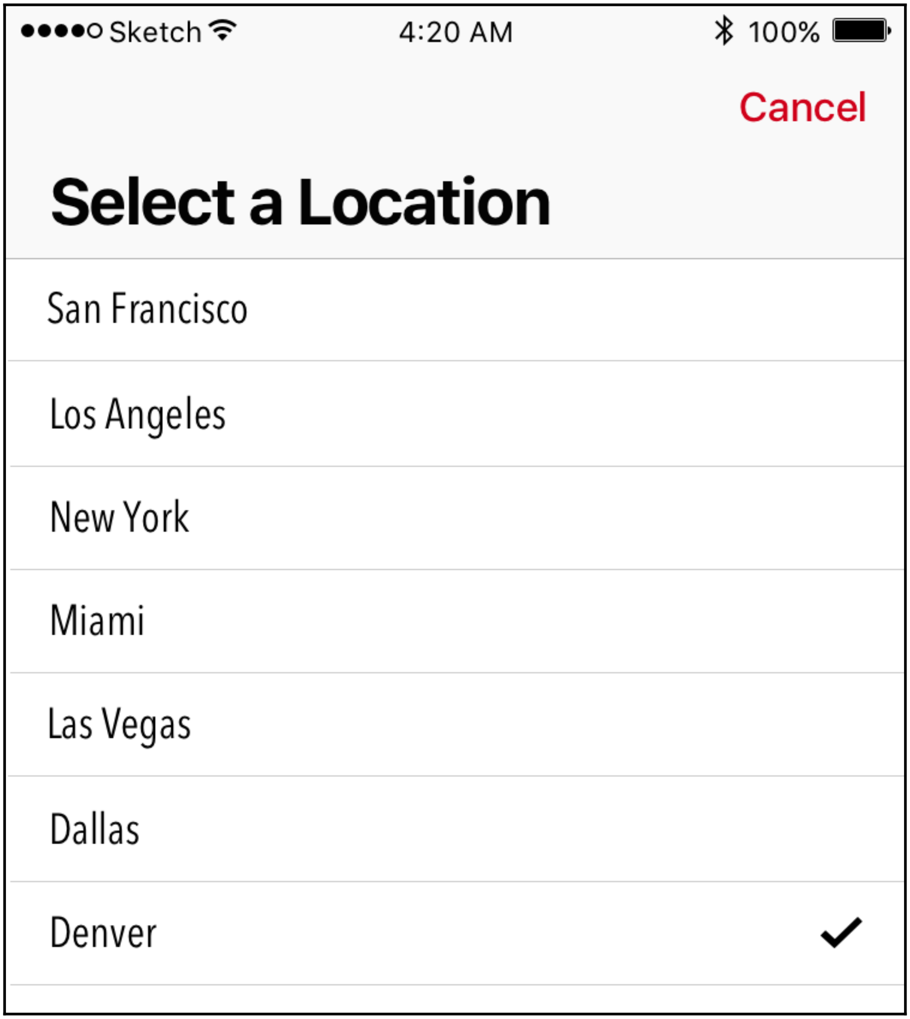

Tab Bar Controller
Navigation Controller
Navigation Controller
Navigation Controller
Navigation Controller
Navigation Controller
Navigation Controller
LET'S EAT
LET'S EAT
LET'S EAT
LET'S EAT
Label
Full Menu





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,


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



Select a Location

San Francisco

Los Angeles

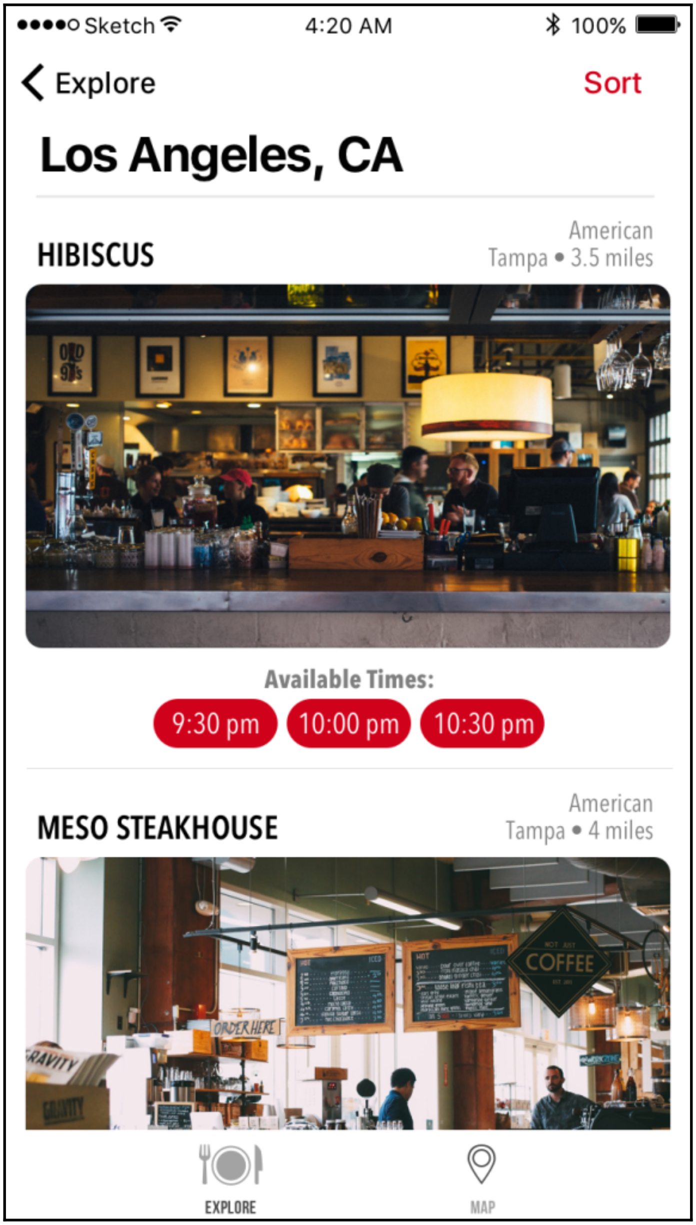
New York

Miami

Las Vegas

Dallas

Denver ✓



Los Angeles, CA

HIBISCUS

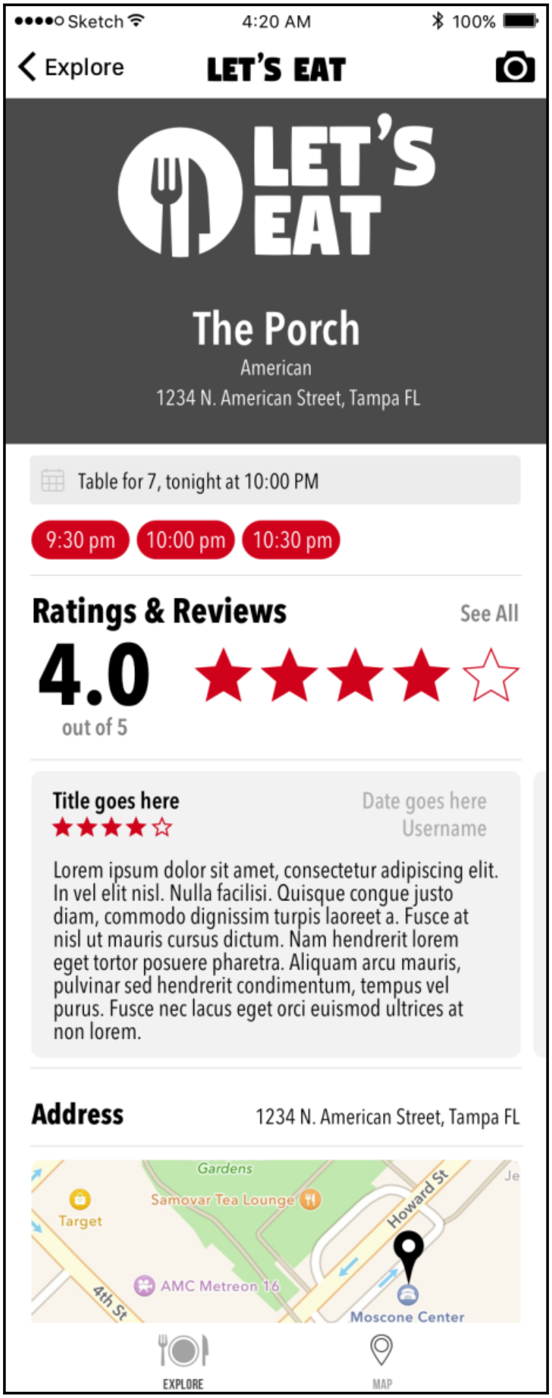
American
Tampa • 3.5 miles

Available Times:

9:30 pm 10:00 pm 10:30 pm

MESO STEAKHOUSE

American
Tampa • 4 miles



The Porch

American

1234 N. American Street, Tampa FL

Table for 7, tonight at 10:00 PM

Ratings & Reviews

4.0 ★★★★☆

out of 5

Title goes here

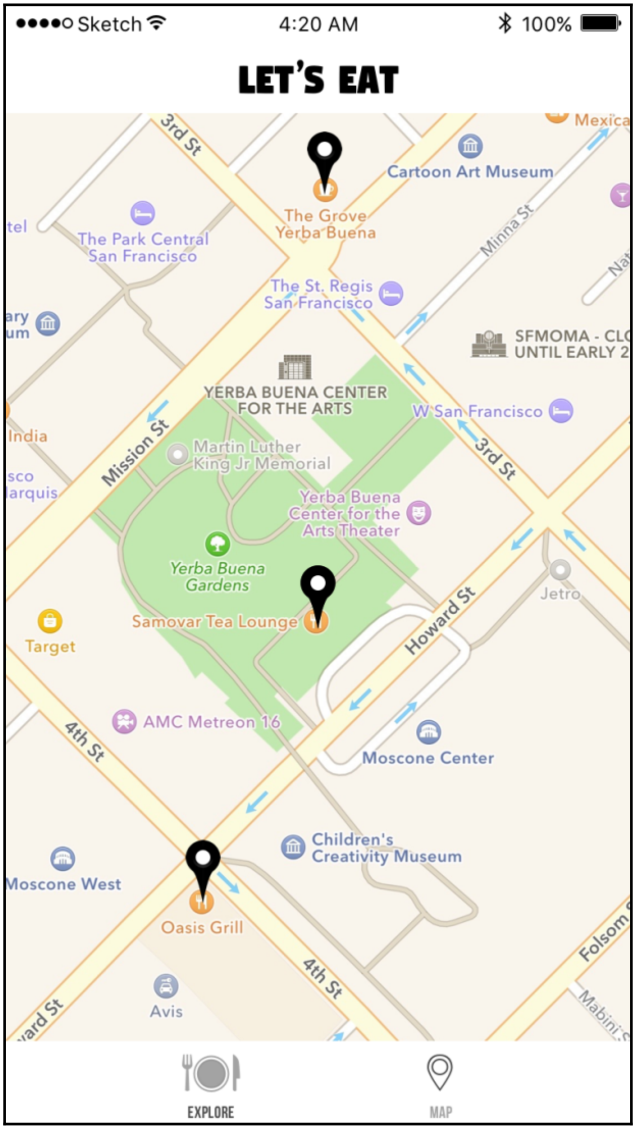
★★★★☆

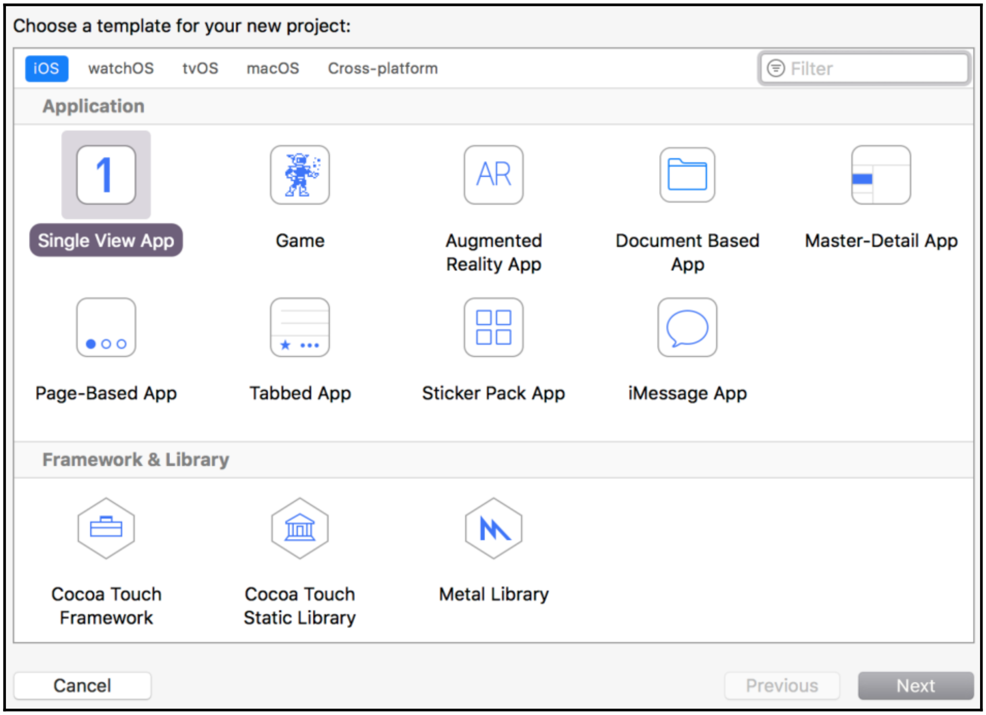
Date goes here

Username

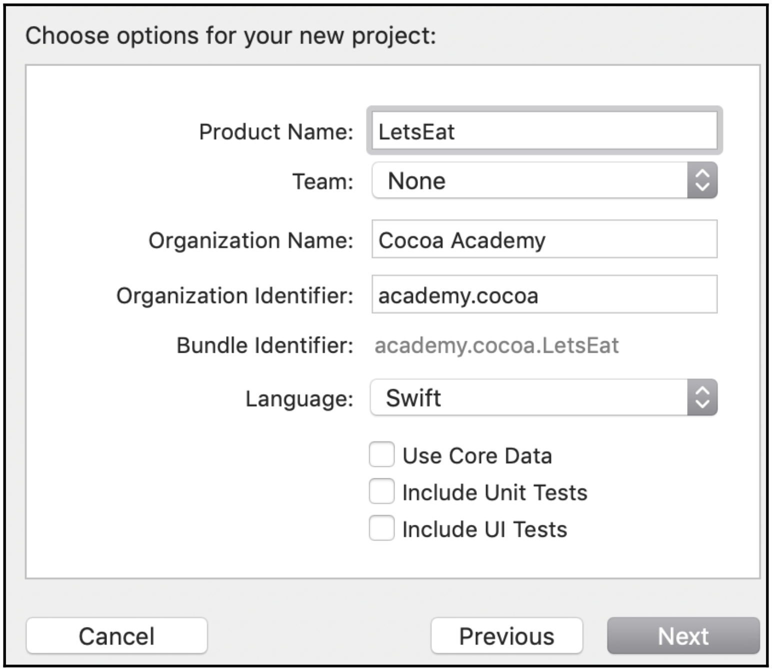
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vel elit nisl. Nulla facilisi. Quisque congue justo diam, commodo dignissim turpis laoreet a. Fusce at nisl ut mauris cursus dictum. Nam hendrerit lorem eget tortor posuere pharetra. Aliquam arcu mauris, pulvinar sed hendrerit condimentum, tempus vel purus. Fusce nec lacus eget orci euismod ultrices at non lorem.

Address 1234 N. American Street, Tampa FL


●●●●○ Sketch
4:20 AM
100%
LET'S EAT
3rd St
Cartoon Art Museum
Mexica
The Grove
Yerba Buena
tel
The Park Central
San Francisco
Minna St
Nat
The St. Regis
San Francisco
ary
um
SFMOMA - CLO
UNTIL EARLY 2
YERBA BUENA CENTER
FOR THE ARTS
W San Francisco
Mission St
India
Martin Luther
King Jr Memorial
3rd St
sco
arquis
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
Folsom
4th St
Avis
Mabini
ard St
EXPLORE
MAP



Choose a template for your new project:



Choose options for your new project:

Product Name: LetsEat

Team: None

Organization Name: Cocoa Academy

Organization Identifier: academy.cocoa

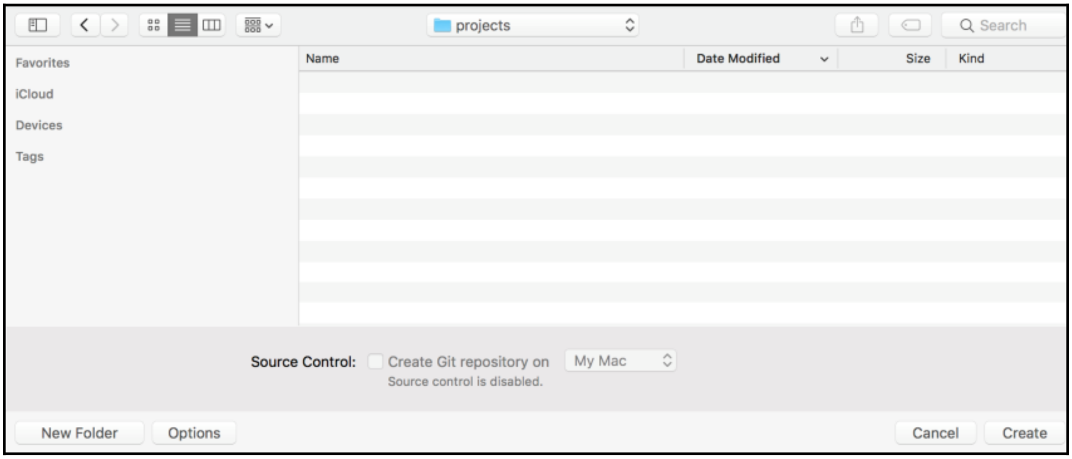
Bundle Identifier: academy.cocoa.LetsEat

Language: Swift

□ Use Core Data

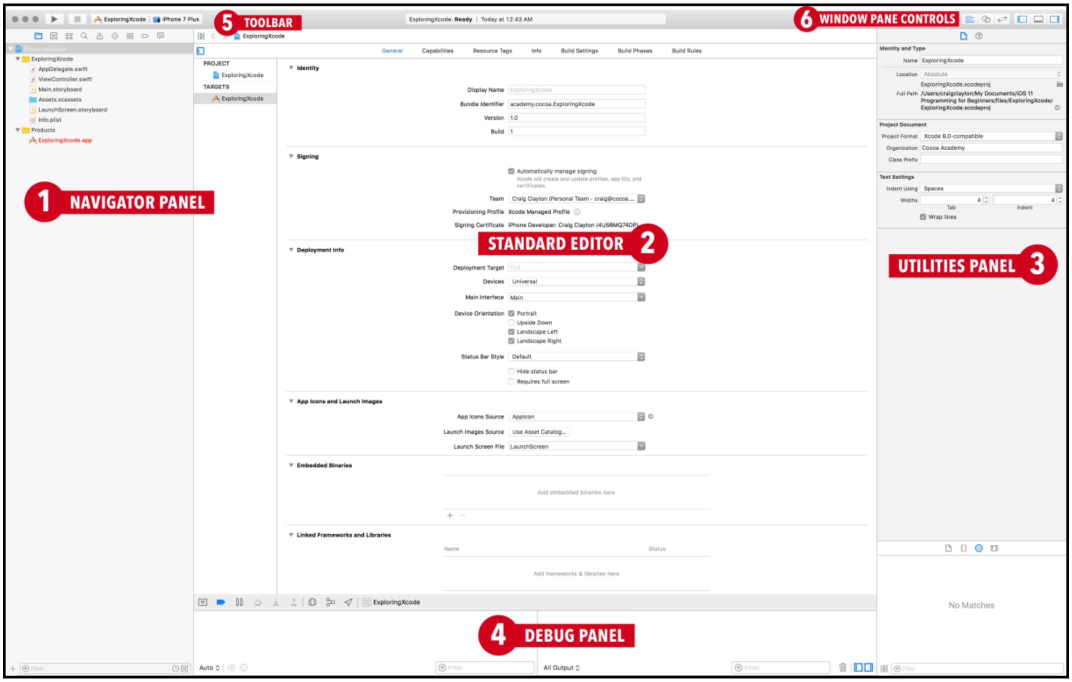
□ Include Unit Tests

□ Include UI Tests



Source Control: □ Create Git repository on My Mac

Source control is disabled.



Identity

Display Name: ExploringXcode
Bundle Identifier: academy.cocoa.ExploringXcode
Version: 1.0
Build: 1

Signing

☑ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team: Craig Clayton (Personal Team - craig@cocoa....
Provisioning Profile: Xcode Managed Profile
Signing Certificate: iPhone Developer: Craig Clayton (4U5BMQ74DP)

Deployment Info

Deployment Target: 11.0
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

Linked Frameworks and Libraries

Identity and Type

Name: ExploringXcode
Location: Absolute
ExploringXcode.xcodeproj
Full Path: /Users/craigclayton/My Documents/iOS 11 Programming for Beginners/files/ExploringXcode/ExploringXcode.xcodeproj

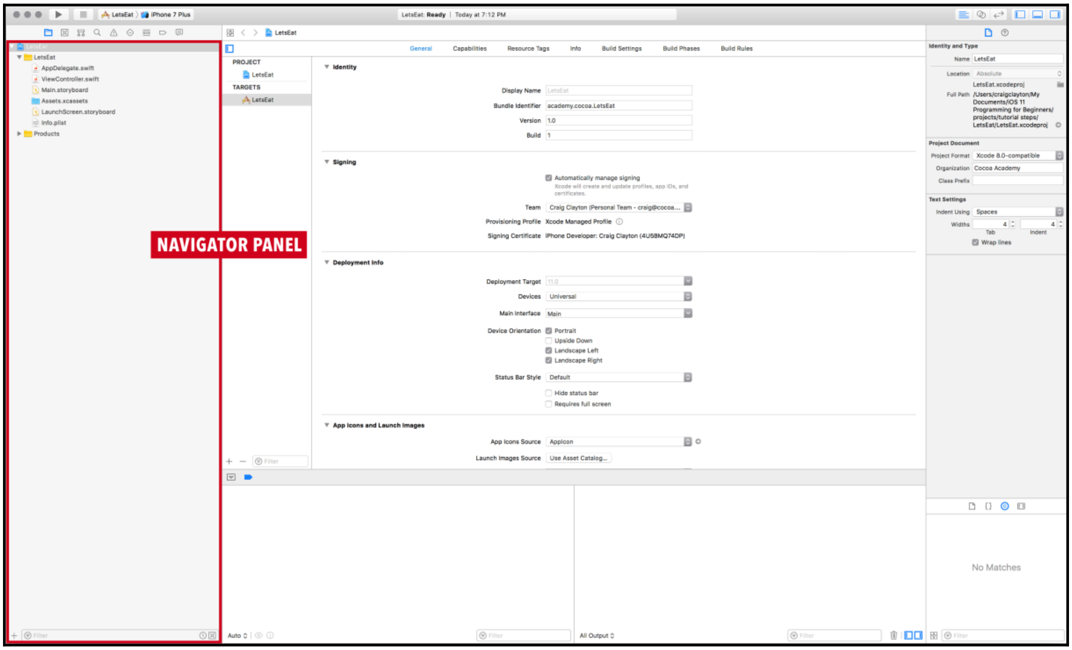
Project Document

Project Format: Xcode 8.0-compatible
Organization: Cocoa Academy
Class Prefix:

Text Settings

Indent Using: Spaces
Widths: 4 Tab, 4 Indent
☑ Wrap lines

Chapter 7: Setting Up the Basic Structure



Identity

Display Name: LetsEat
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 (Personal Team - craig@cocoa....
Provisioning Profile: Xcode Managed Profile
Signing Certificate: iPhone Developer: Craig Clayton (4U5BMQ74DP)

Deployment Info

Deployment Target: 11.0
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...

Identity and Type

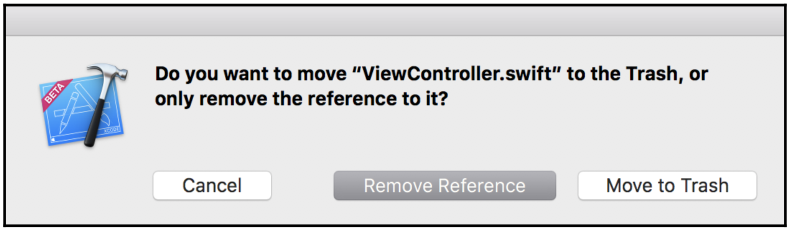
Name: LetsEat
Location: Absolute
LetsEat.xcodeproj
Full Path: /Users/craigclayton/My Documents/iOS 11 Programming for Beginners/projects/tutorial steps/LetsEat/LetsEat.xcodeproj

Project Document

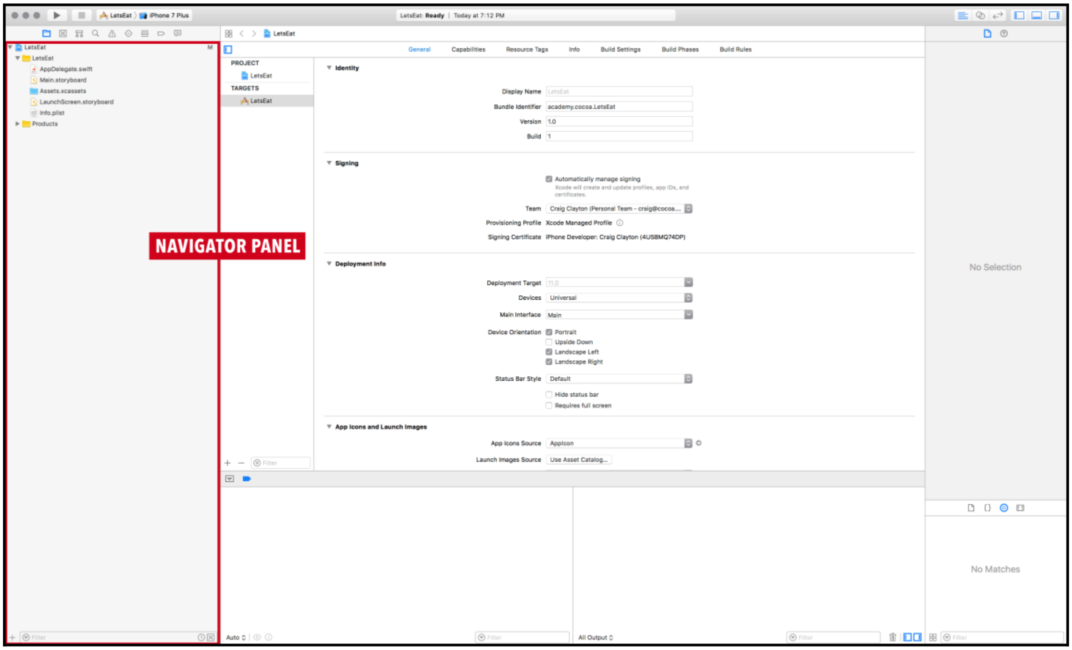
Project Format: Xcode 8.0-compatible
Organization: Cocoa Academy
Class Prefix:

Text Settings

Indent Using: Spaces
Widths: 4 (Tab) 4 (Indent)
☑ Wrap lines



Do you want to move "ViewController.swift" to the Trash, or only remove the reference to it?



Identity

Display Name LetsEat

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 (Personal Team - craig@cocoa....

Provisioning Profile Xcode Managed Profile

Signing Certificate iPhone Developer: Craig Clayton (4U5BMQ74DP)

Deployment Info

Deployment Target 11.0

Devices Universal

Main Interface Main

Device Orientation ☑ Portrait
☐ Upside Down
☑ Landscape Left
☑ Landscape Right

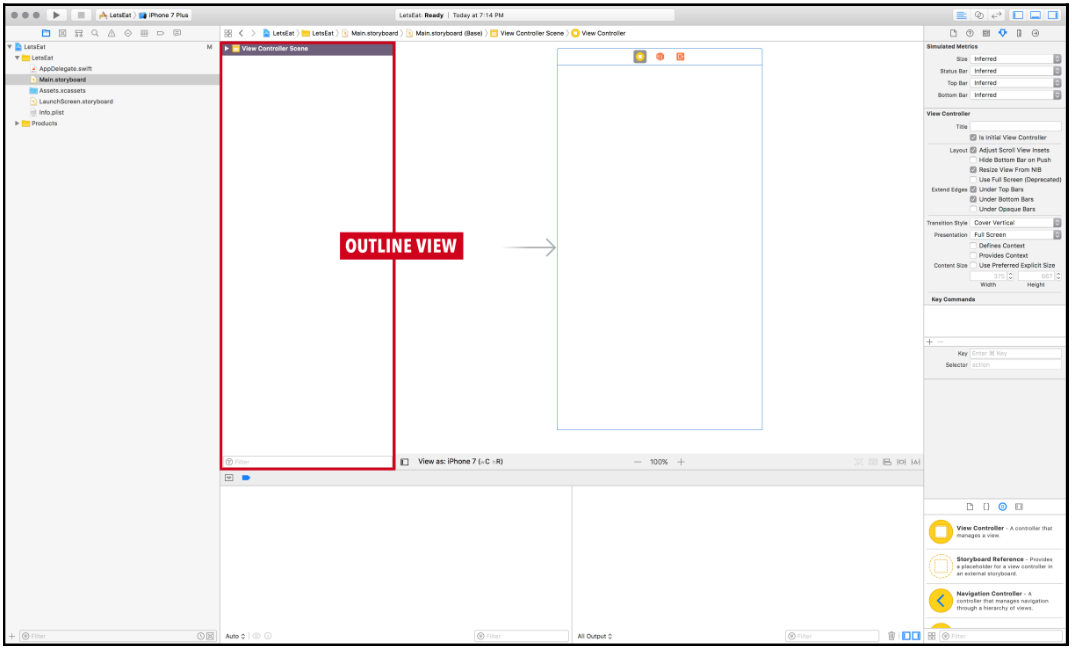
Status Bar Style Default

☐ Hide status bar
☐ Requires full screen

App Icons and Launch Images

App Icons Source AppIcon

Launch Images Source Use Asset Catalog...



Simulated Metrics

Size Inferred

Status Bar Inferred

Top Bar Inferred

Bottom Bar Inferred

View Controller

Title

☑ Is Initial View Controller

Layout ☑ Adjust Scroll View Insets
☐ Hide Bottom Bar on Push
☑ Resize View From NIB
☐ Use Full Screen (Deprecated)

Extend Edges ☑ Under Top Bars
☑ Under Bottom Bars
☐ Under Opaque Bars

Transition Style Cover Vertical

Presentation Full Screen

☐ Defines Context
☐ Provides Context

Content Size ☐ Use Preferred Explicit Size
375 667
Width Height

Key Commands

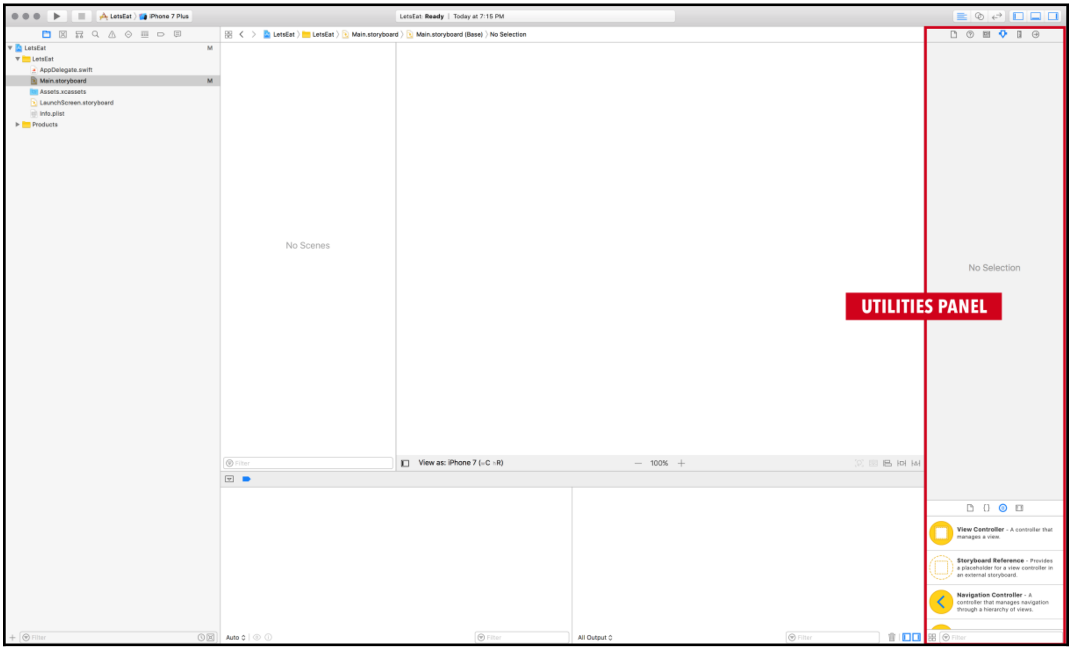
Key Enter ⌘ Key

Selector action

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.

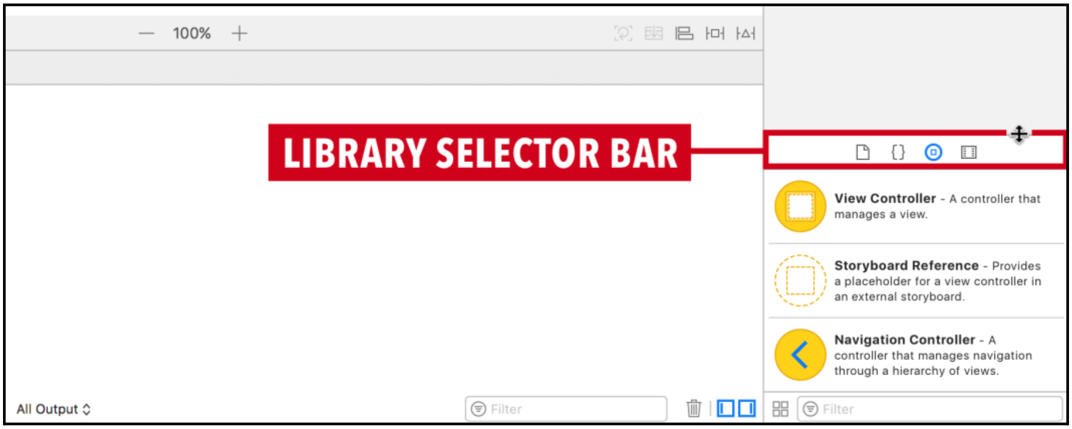


UTILITIES PANEL

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.

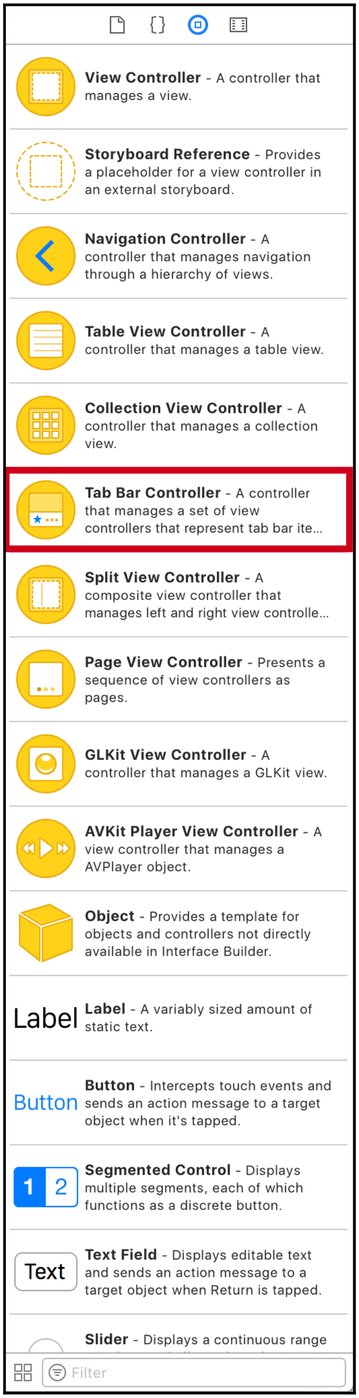


LIBRARY SELECTOR BAR

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.



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.

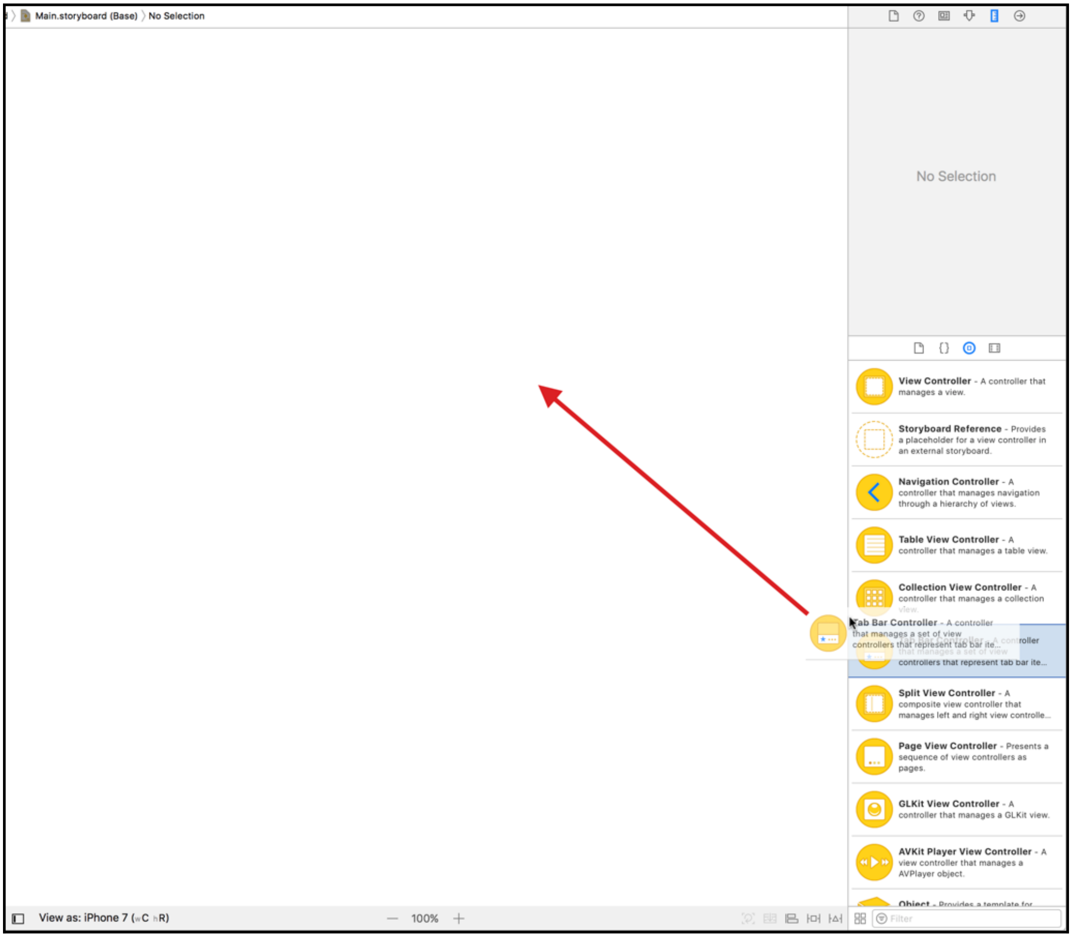
Label - A variably sized amount of static text.

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

Slider - Displays a continuous range



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...

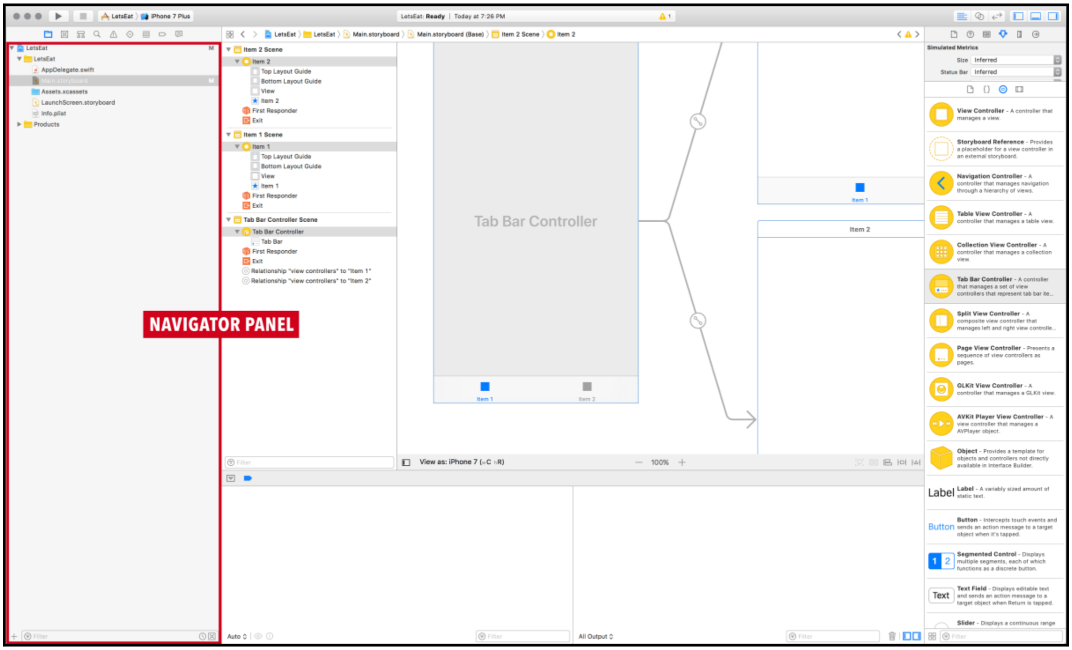
Split View Controller - A composite view controller that manages left and right view controlle...

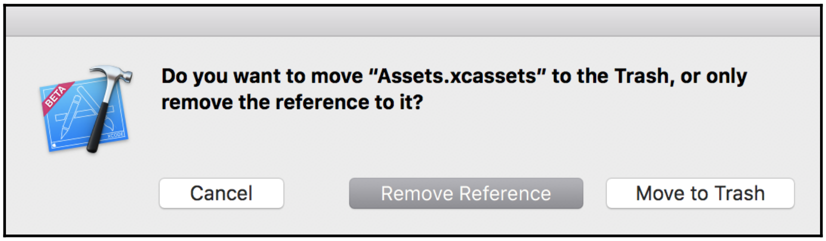
Page View Controller - Presents a sequence of view controllers as pages.

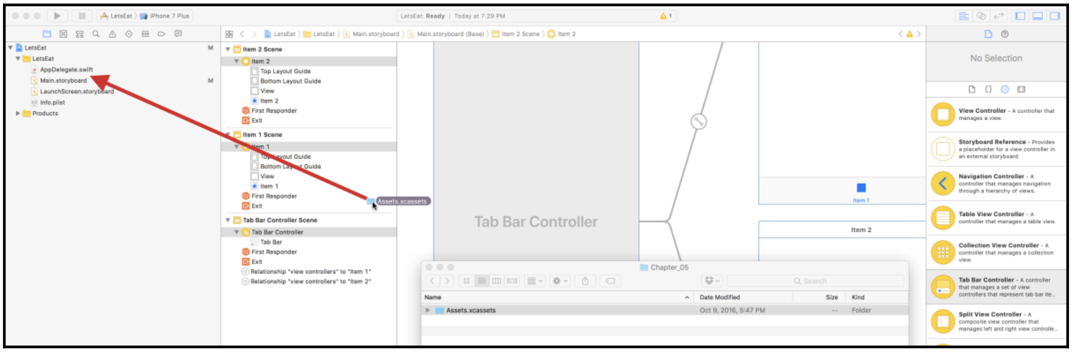
GLKit View Controller - A controller that manages a GLKit view.

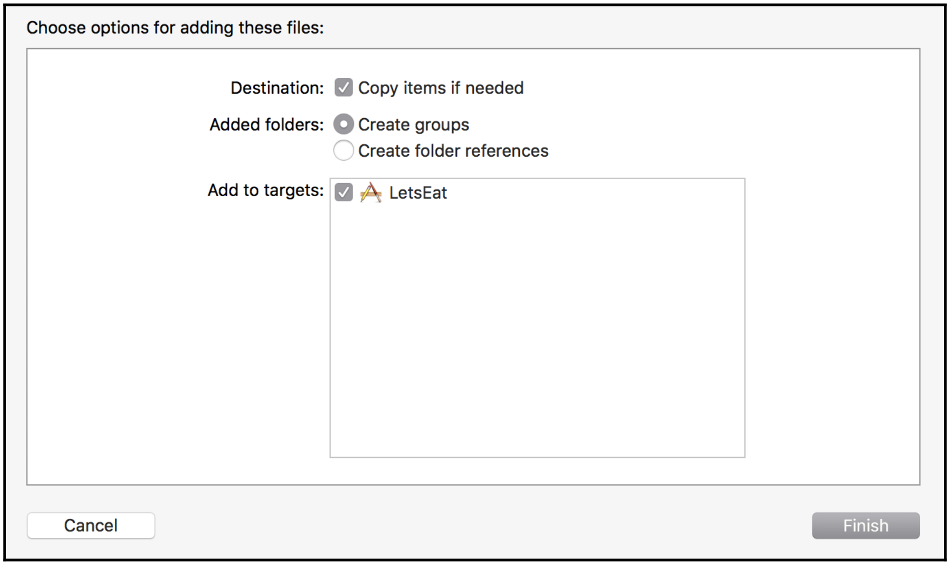
AVKit Player View Controller - A view controller that manages a AVPlayer object.

Object - Provides a template for




Do you want to move “Assets.xcassets” to the Trash, or only remove the reference to it?





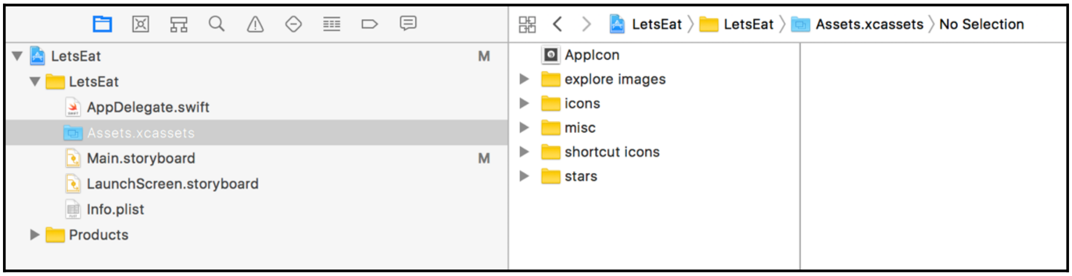
Choose options for adding these files:

Destination: ☑ Copy items if needed

Added folders: ● Create groups

○ Create folder references

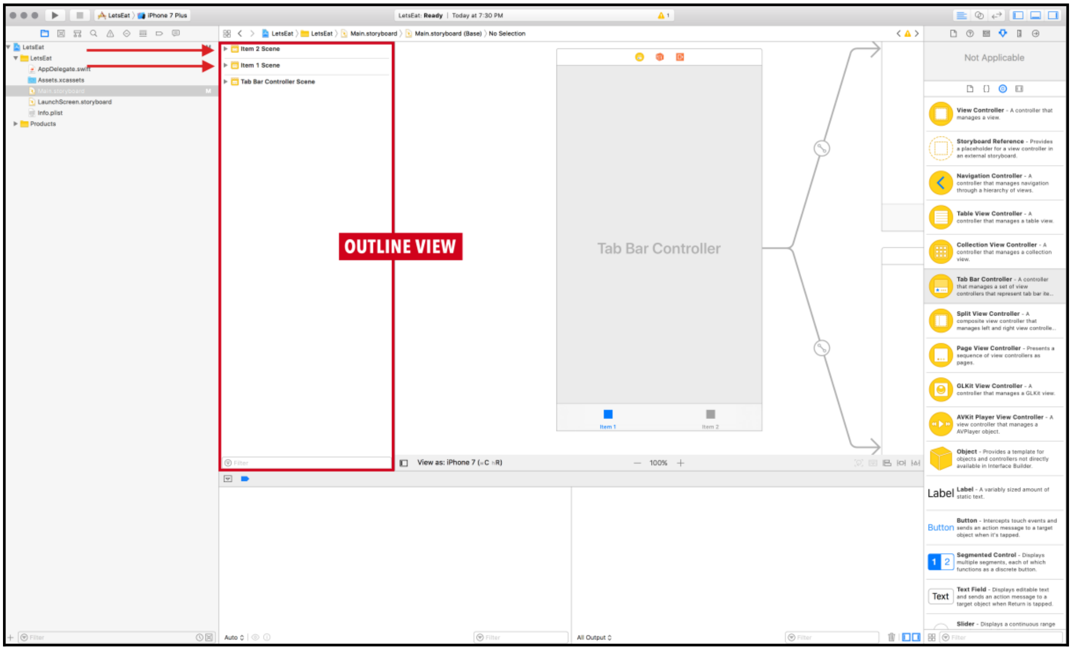
Add to targets: ☑ LetsEat

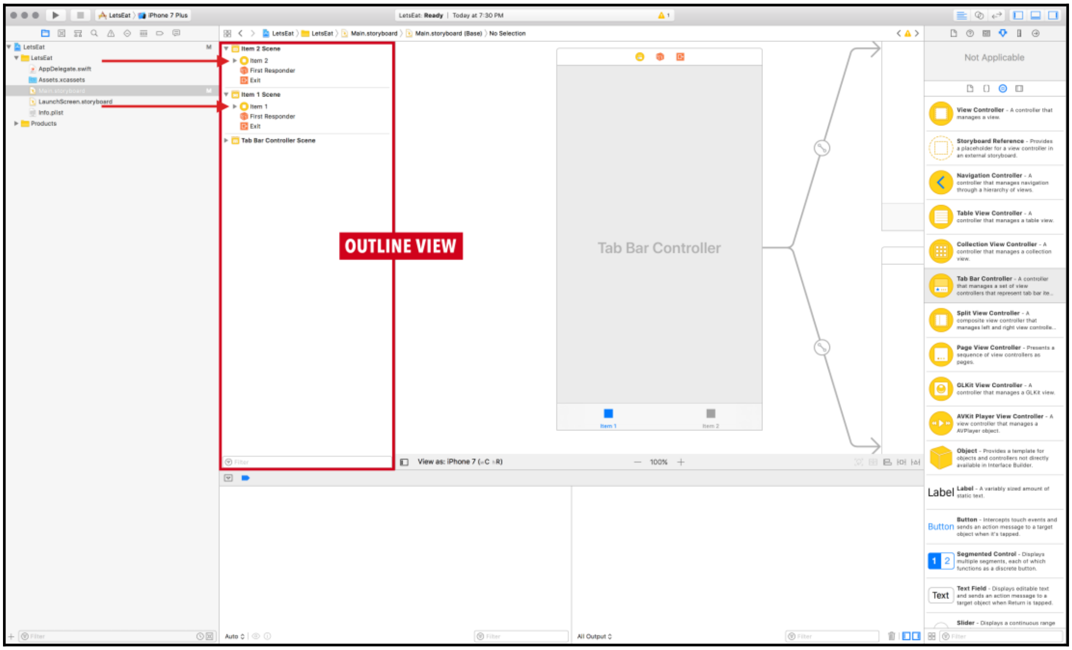


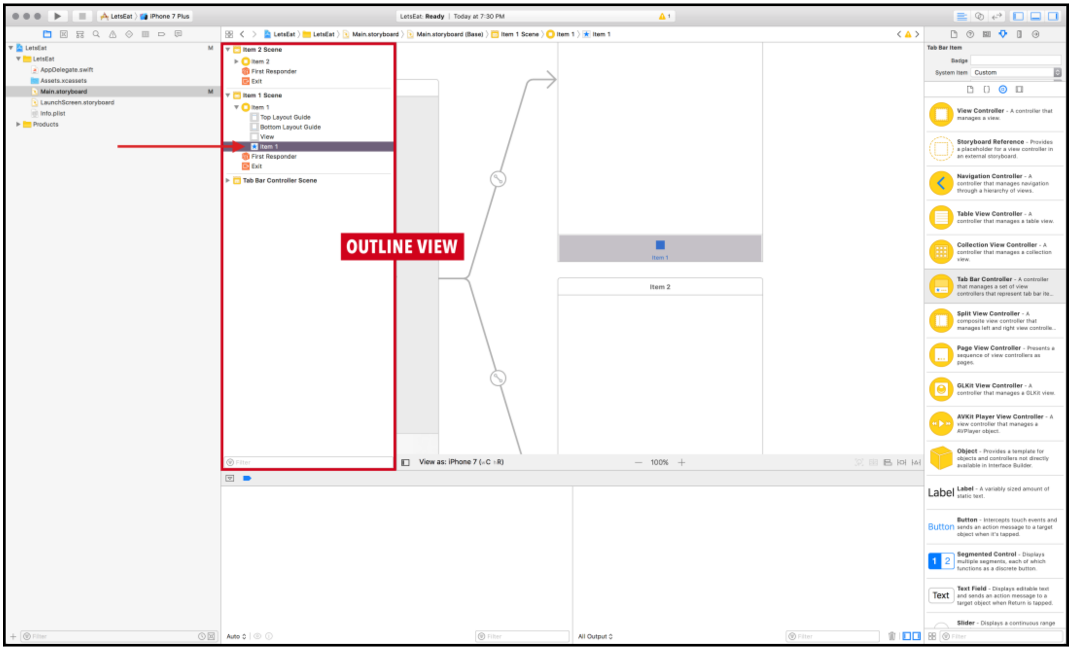
```
▼ LetsEat                          M
  ▼ LetsEat
      AppDelegate.swift
      Assets.xcassets
      Main.storyboard              M
      LaunchScreen.storyboard
      Info.plist
  ▶ Products
```

LetsEat 〉 LetsEat 〉 Assets.xcassets 〉 No Selection

```
  AppIcon
▶ explore images
▶ icons
▶ misc
▶ shortcut icons
▶ stars
```

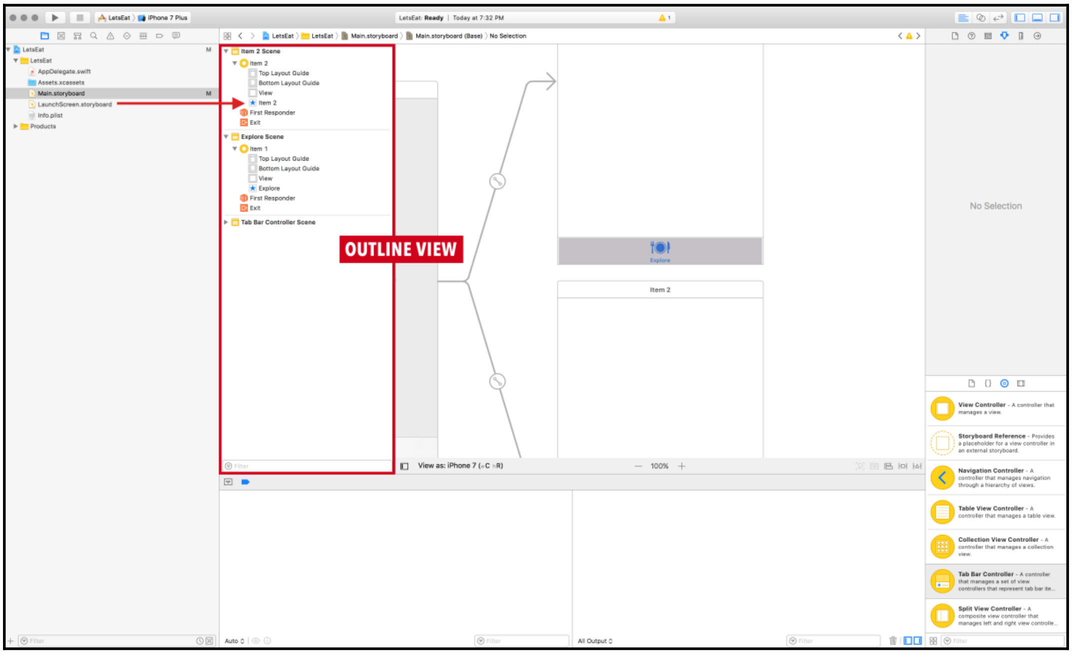

OUTLINE VIEW
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
LaunchScreen.storyboard
Info.plist
Products
Item 2 Scene
Item 1 Scene
Tab Bar Controller Scene
Tab Bar Controller
Item 1
Item 2
View as: iPhone 7 (wC hR)
100%
Not Applicable
View Controller - A controller that manages a view.
Storyboard 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 - A variably sized amount of static text.
Button - Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control - Displays multiple segments, each of which functions as a discrete button.
Text Field - Displays editable text and sends an action message to a target object when Return is tapped.
Slider - Displays a continuous range


OUTLINE VIEW
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
LaunchScreen.storyboard
Info.plist
Products
Item 2 Scene
Item 2
First Responder
Exit
Item 1 Scene
Item 1
First Responder
Exit
Tab Bar Controller Scene
Tab Bar Controller
Item 1
Item 2
View as: iPhone 7 (wC hR)
100%
Not Applicable



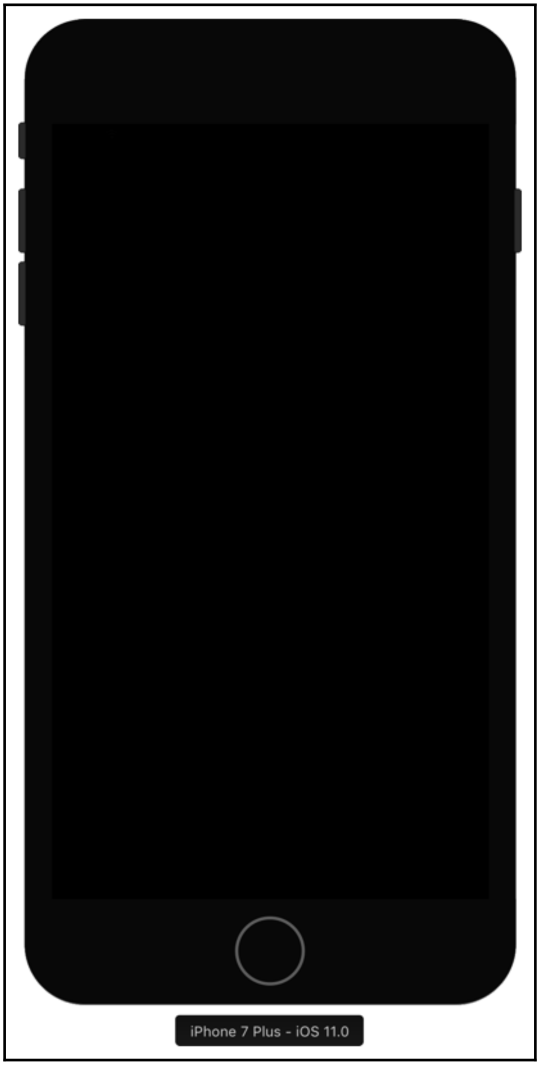
OUTLINE VIEW

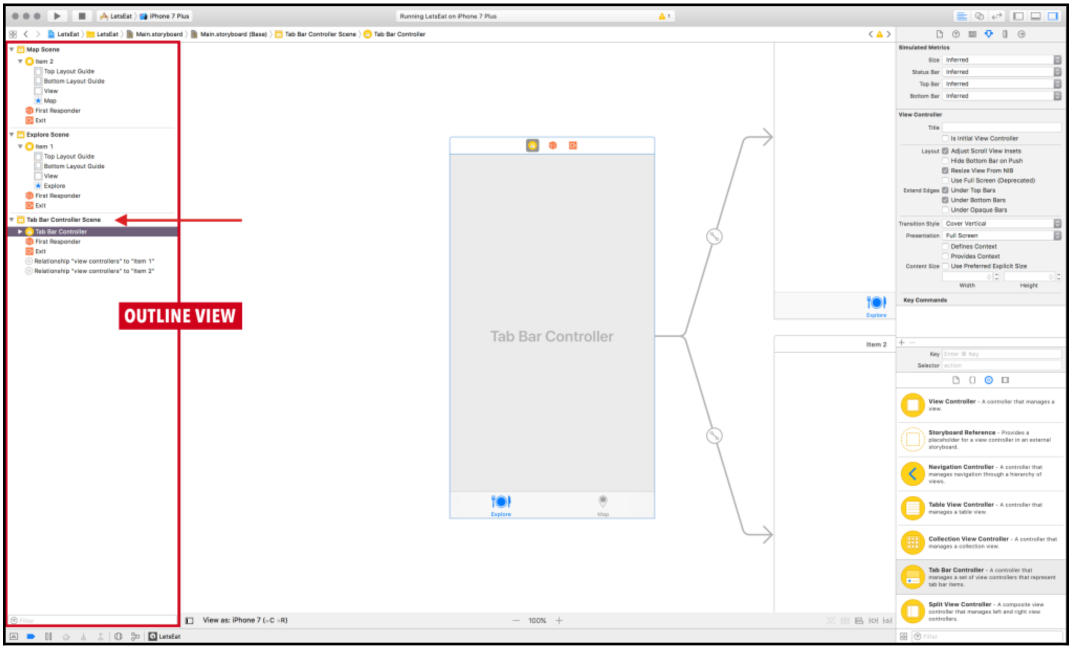
LetsEat
iPhone 7 Plus
LetsEat: Ready | Today at 7:30 PM
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
LaunchScreen.storyboard
Info.plist
Products
Item 2 Scene
Item 2
First Responder
Exit
Item 1 Scene
Item 1
Top Layout Guide
Bottom Layout Guide
View
Item 1
First Responder
Exit
Tab Bar Controller Scene
Item 1
Item 2
View as: iPhone 7
100%
Tab Bar Item
Badge
System Item
Custom
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 - A variably sized amount of static text.
Button - Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control - Displays multiple segments, each of which functions as a discrete button.
Text Field - Displays editable text and sends an action message to a target object when Return is tapped.
Slider - Displays a continuous range
Auto
All Output
Filter



OUTLINE VIEW

LetsEat
iPhone 7 Plus
LetsEat: Ready | Today at 7:32 PM
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard
LaunchScreen.storyboard
Info.plist
Products
Main.storyboard (Base)
No Selection
Item 2 Scene
Item 2
Top Layout Guide
Bottom Layout Guide
View
Item 2
First Responder
Exit
Explore Scene
Item 1
Top Layout Guide
Bottom Layout Guide
View
Explore
First Responder
Exit
Tab Bar Controller Scene
Explore
Item 2
View as: iPhone 7
100%
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...
Auto
All Output
Filter


iPhone 7 Plus - iOS 11.0



Simulated Metrics

Size: Inferred
Status Bar: Inferred
Top Bar: Inferred
Bottom Bar: Inferred

View Controller

Title:
□ Is Initial View Controller

Layout: ☑ Adjust Scroll View Insets
□ Hide Bottom Bar on Push
☑ Resize View From NIB
□ Use Full Screen (Deprecated)

Extend Edges: ☑ Under Top Bars
☑ Under Bottom Bars
□ Under Opaque Bars

Transition Style: Cover Vertical
Presentation: Full Screen
□ Defines Context
□ Provides Context

Content Size: □ Use Preferred Explicit Size
Width 0 Height 0

Key Commands

Key: Enter ⌘ Key
Selector: action

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

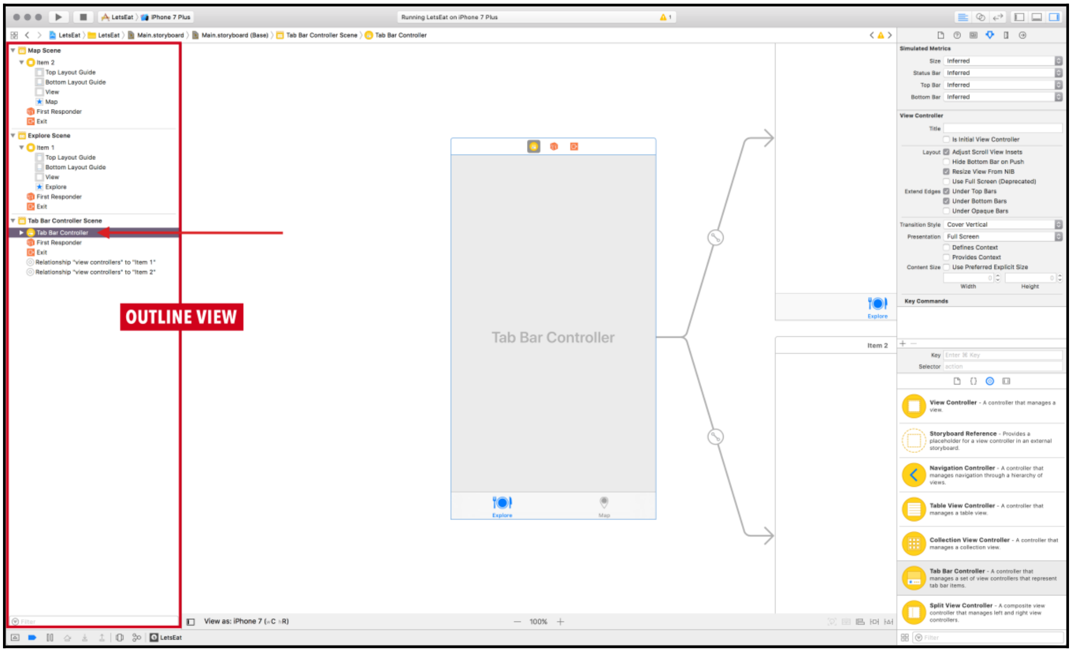
Navigation Controller - A controller that manages navigation through a hierarchy of views.

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.



Simulated Metrics

Size: Inferred
Status Bar: Inferred
Top Bar: Inferred
Bottom Bar: Inferred

View Controller

Title:
□ Is Initial View Controller

Layout: ☑ Adjust Scroll View Insets
□ Hide Bottom Bar on Push
☑ Resize View From NIB
□ Use Full Screen (Deprecated)

Extend Edges: ☑ Under Top Bars
☑ Under Bottom Bars
□ Under Opaque Bars

Transition Style: Cover Vertical
Presentation: Full Screen
□ Defines Context
□ Provides Context

Content Size: □ Use Preferred Explicit Size
Width 0 Height 0

Key Commands

Key: Enter ⌘ Key
Selector: action

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

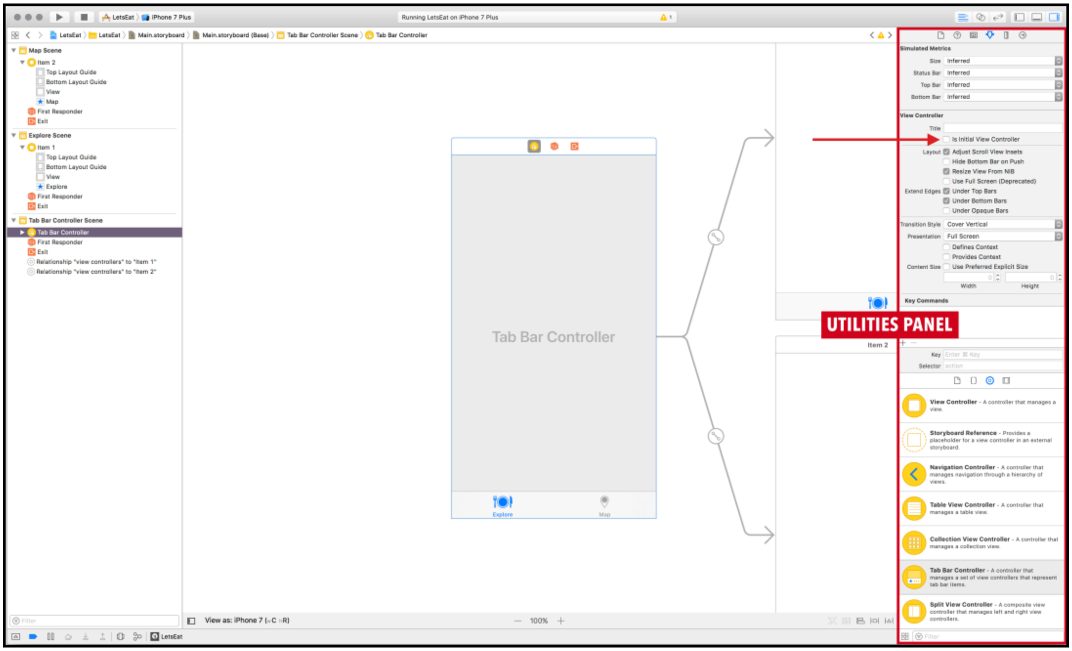
Navigation Controller - A controller that manages navigation through a hierarchy of views.

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.



Simulated Metrics

Size: Inferred
Status Bar: Inferred
Top Bar: Inferred
Bottom Bar: Inferred

View Controller

Title:
□ Is Initial View Controller
Layout: ☑ Adjust Scroll View Insets
□ Hide Bottom Bar on Push
☑ Resize View From NIB
□ Use Full Screen (Deprecated)
Extend Edges: ☑ Under Top Bars
☑ Under Bottom Bars
□ Under Opaque Bars
Transition Style: Cover Vertical
Presentation: Full Screen
□ Defines Context
□ Provides Context
Content Size: □ Use Preferred Explicit Size
Width: 0 Height: 0

Key Commands

Key: Enter ⌘ Key
Selector: action

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

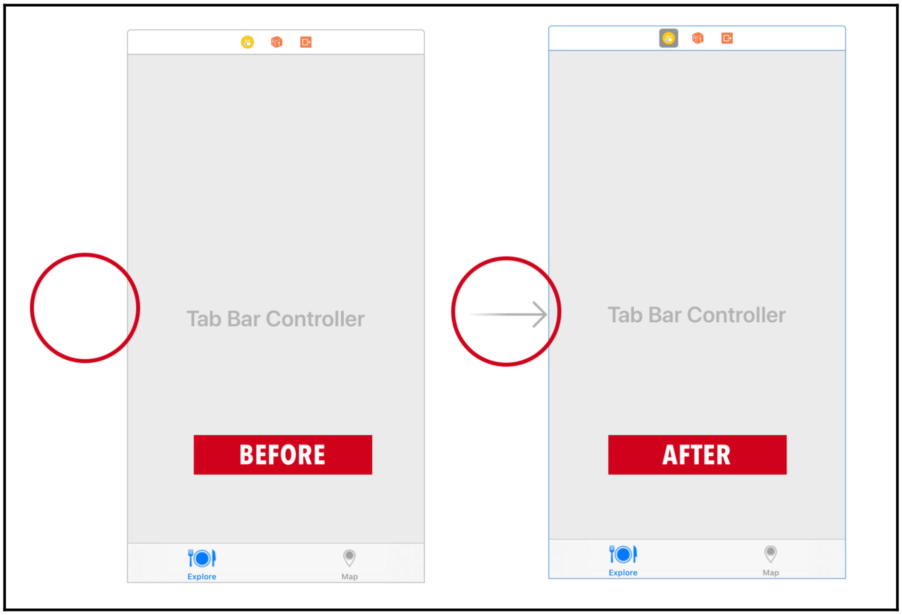
Navigation Controller - A controller that manages navigation through a hierarchy of views.

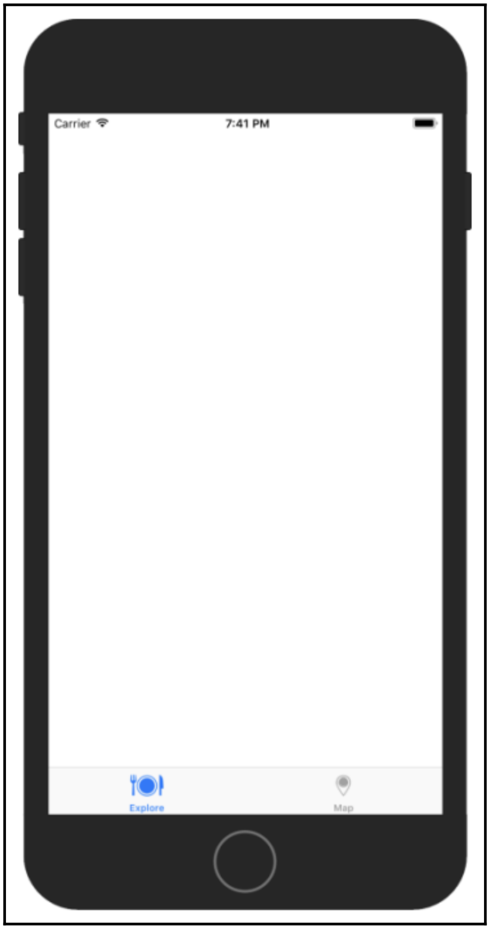
Table View Controller - A controller that manages a table view.

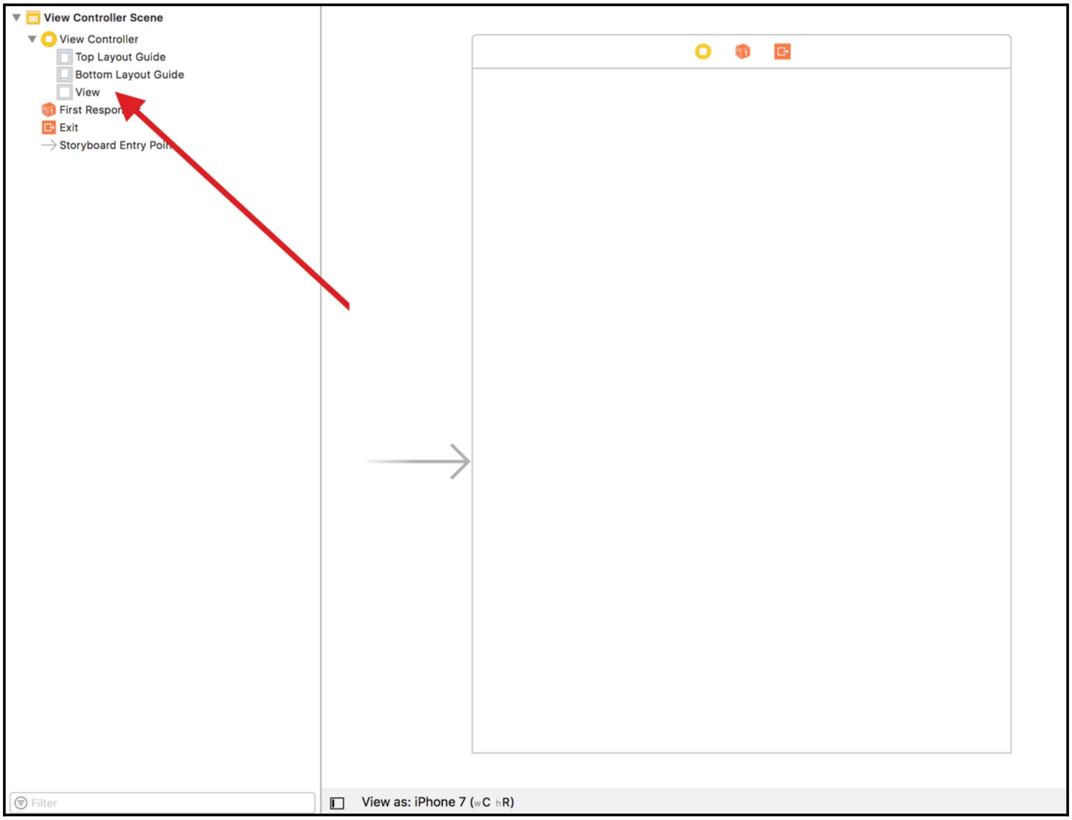
Collection View Controller - A controller that manages a collection view.

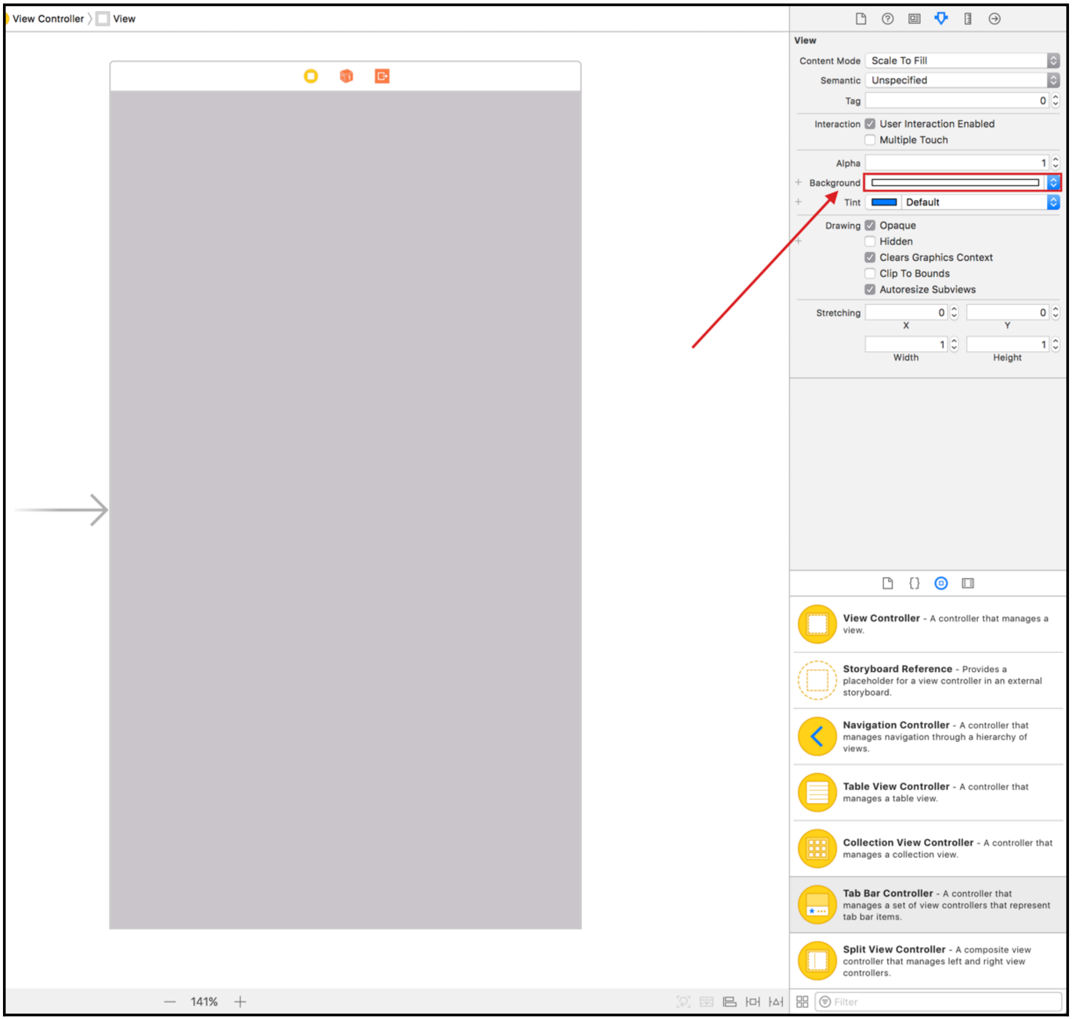
Tab Bar Controller - A controller that manages a set of view controllers that represent tab bar items.

Split View Controller - A composite view controller that manages left and right view controllers.




Carrier
7:41 PM
Explore
Map


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



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: X 0, Y 0
Width 1, Height 1

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

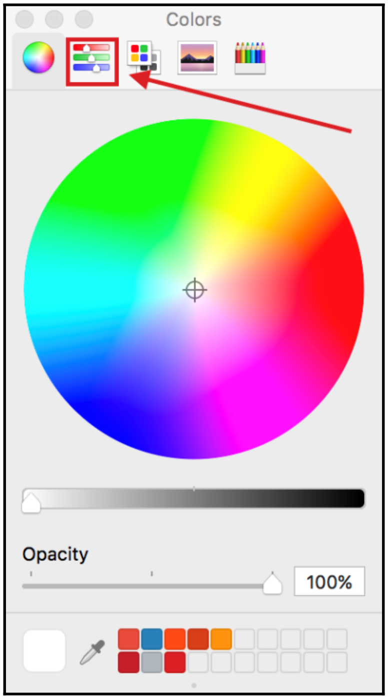
Navigation Controller - A controller that manages navigation through a hierarchy of views.

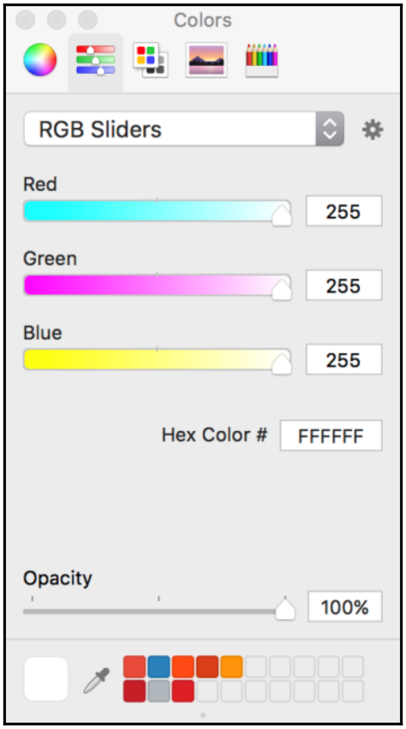
Table View Controller - A controller that manages a table view.

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.


Colors
Opacity
100%



Colors

RGB Sliders

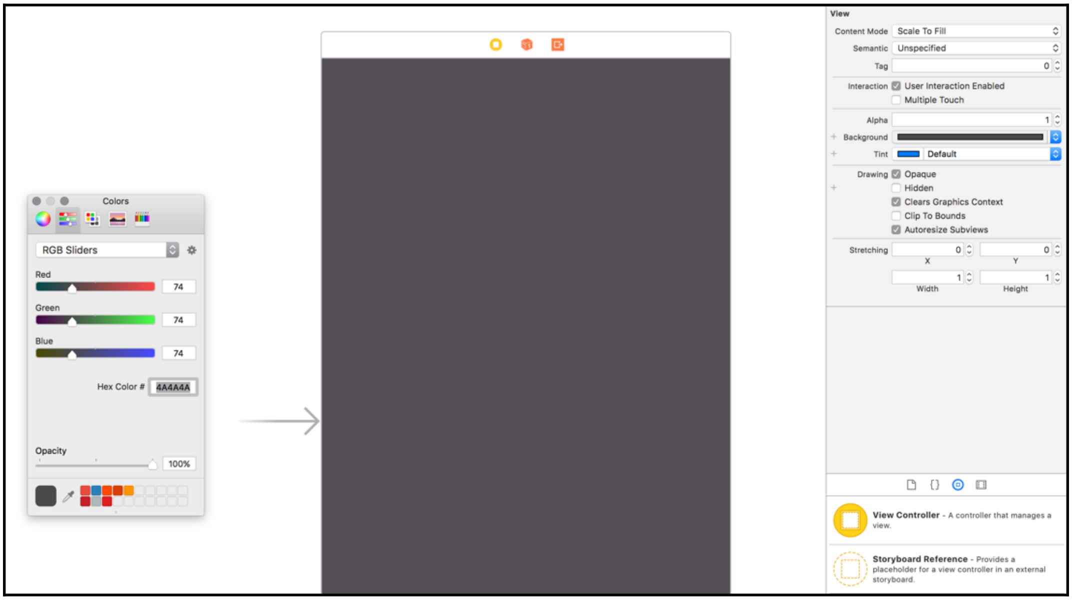
Red 255

Green 255

Blue 255

Hex Color # FFFFFF

Opacity 100%



Colors

RGB Sliders

Red 74

Green 74

Blue 74

Hex Color # 4A4A4A

Opacity 100%

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

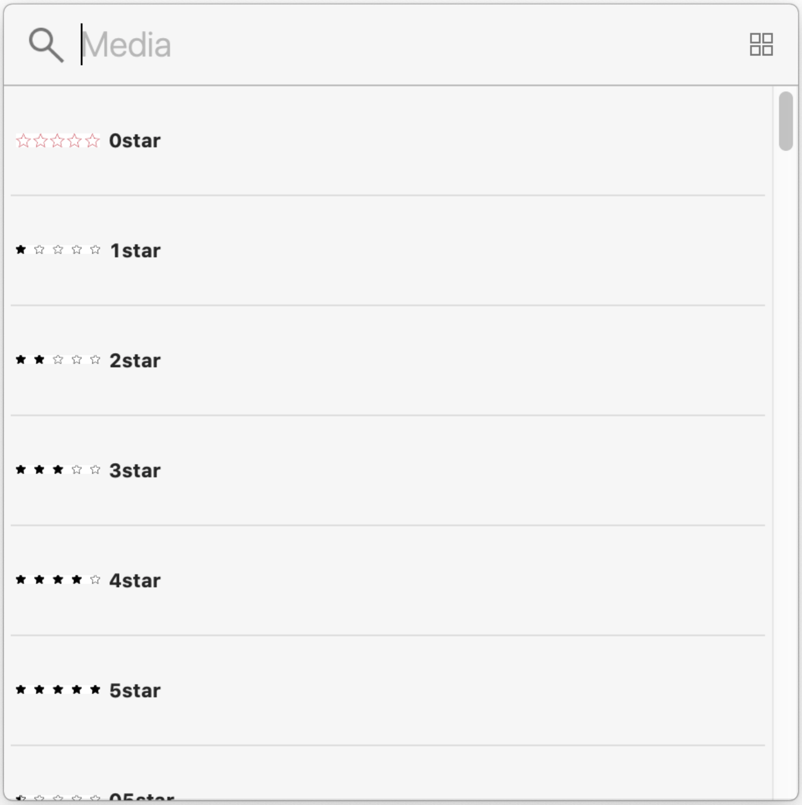
☑ Autoresize Subviews

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.



Media

☆☆☆☆☆ 0star

★☆☆☆☆ 1star

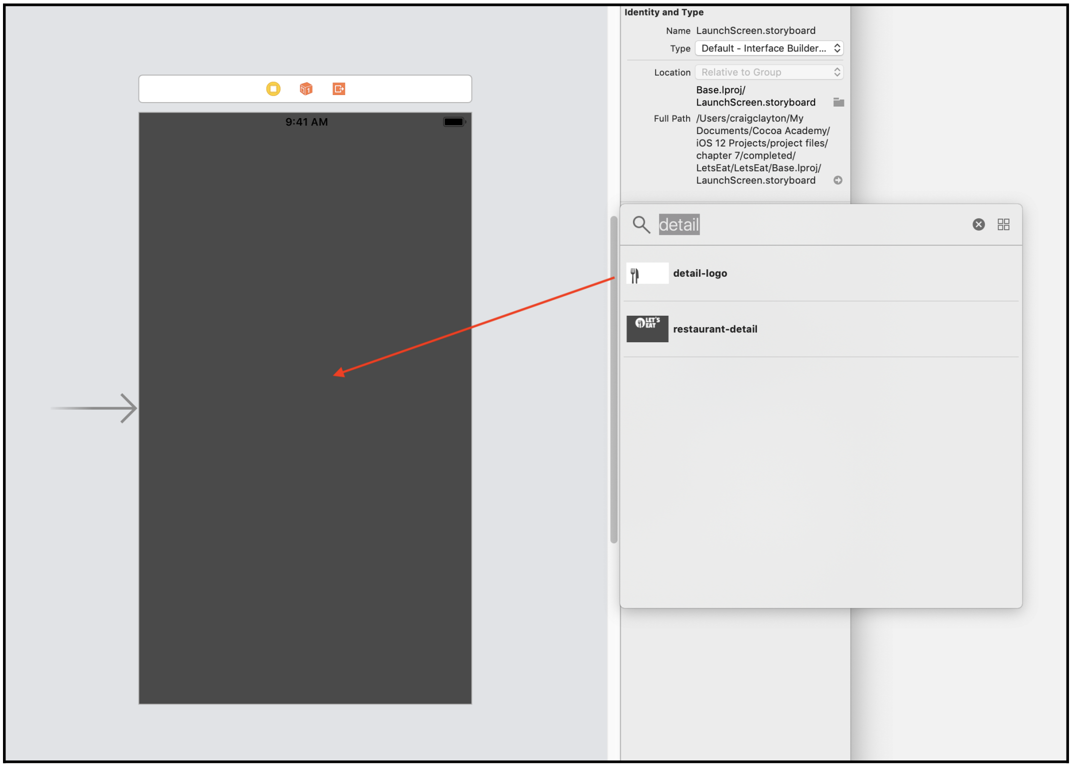
★★☆☆☆ 2star

★★★☆☆ 3star

★★★★☆ 4star

★★★★★ 5star

★☆☆☆☆ 05star



Identity and Type

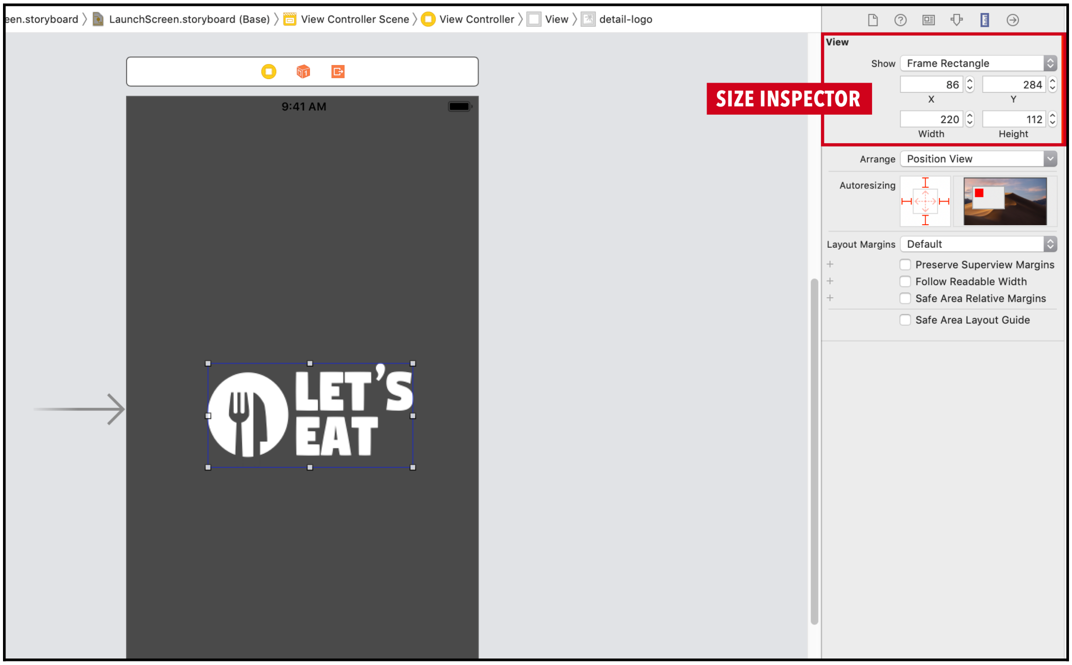
Name LaunchScreen.storyboard

Type Default - Interface Builder...

Location Relative to Group

Base.lproj/
LaunchScreen.storyboard

Full Path /Users/craigclayton/My Documents/Cocoa Academy/iOS 12 Projects/project files/chapter 7/completed/LetsEat/LetsEat/Base.lproj/LaunchScreen.storyboard



een.storyboard 〉 LaunchScreen.storyboard (Base) 〉 View Controller Scene 〉 View Controller 〉 View 〉 detail-logo

View

Show: Frame Rectangle

X: 86 Y: 284

Width: 220 Height: 112

Arrange: Position View

Autoresizing

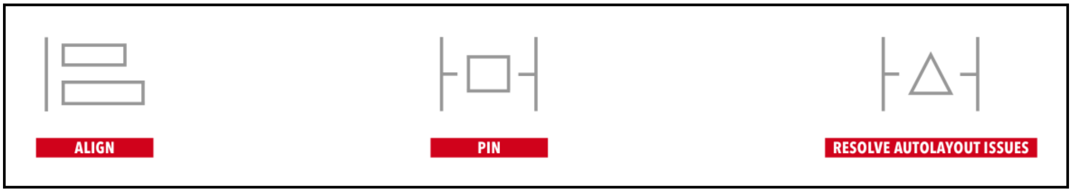
Layout Margins: Default

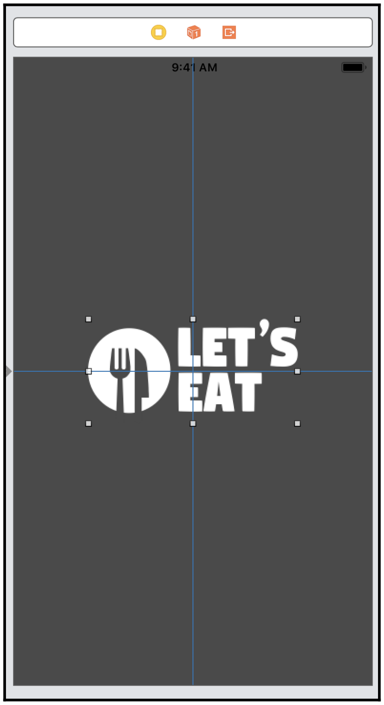
□ Preserve Superview Margins

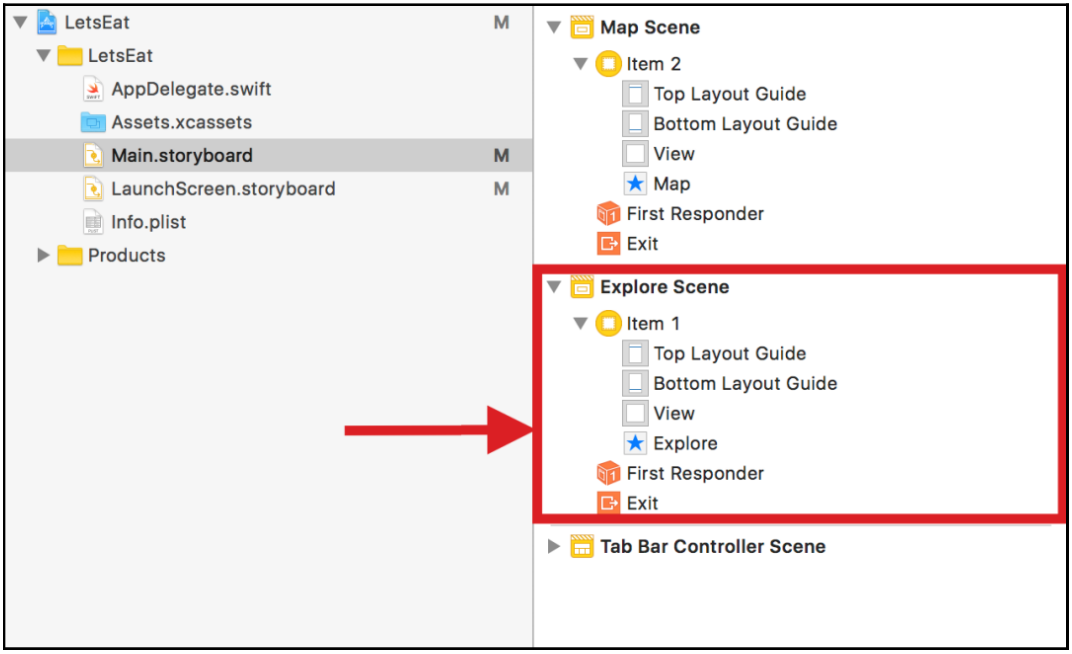
□ Follow Readable Width

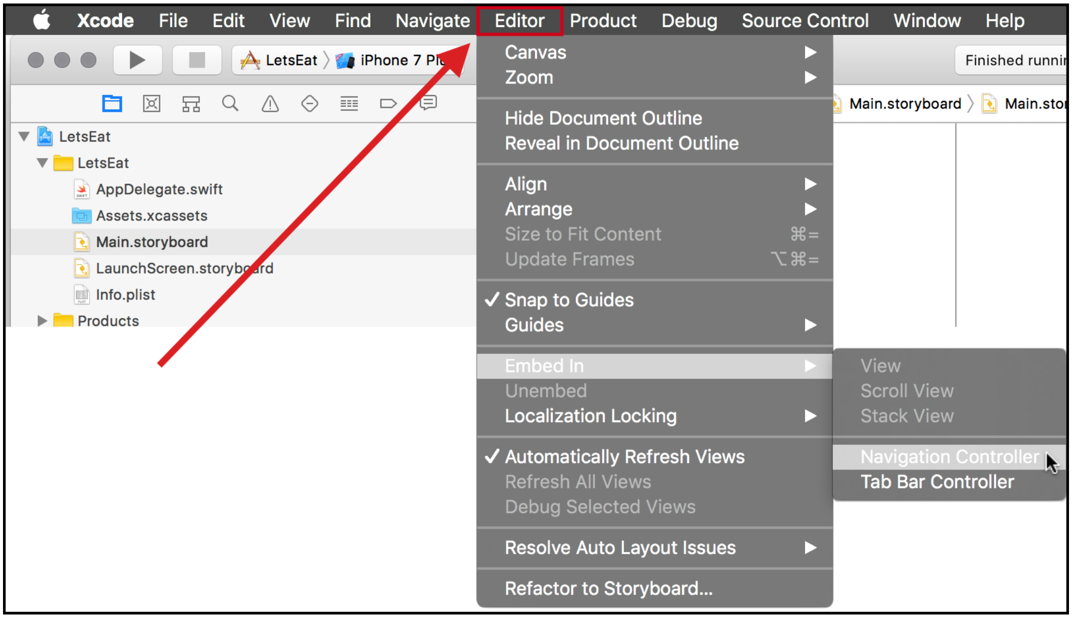
□ Safe Area Relative Margins

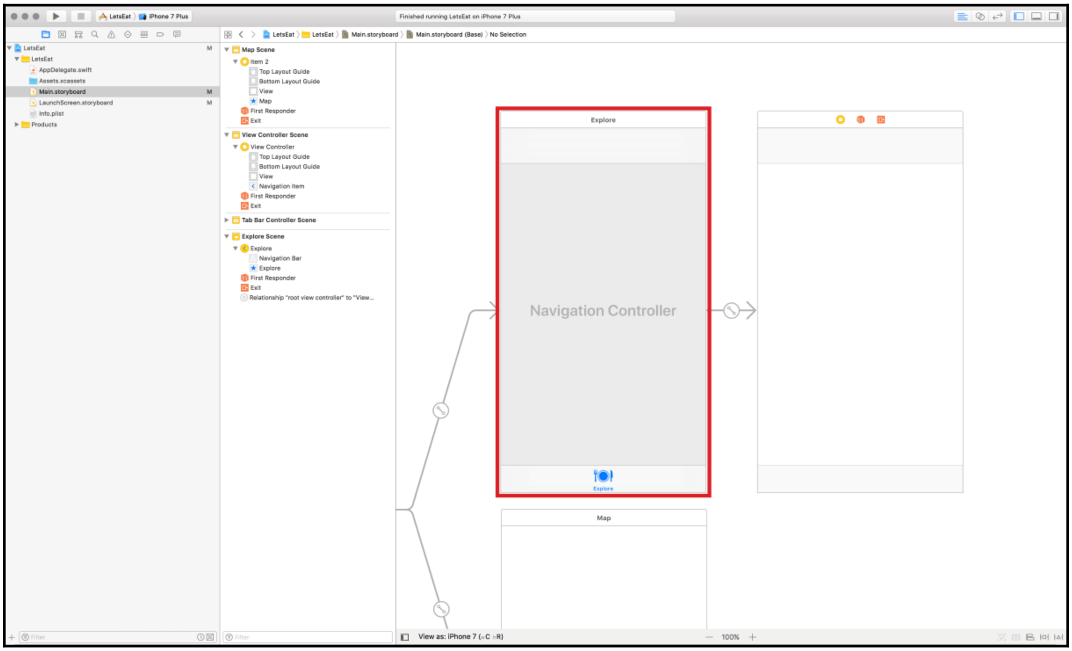
□ Safe Area Layout Guide

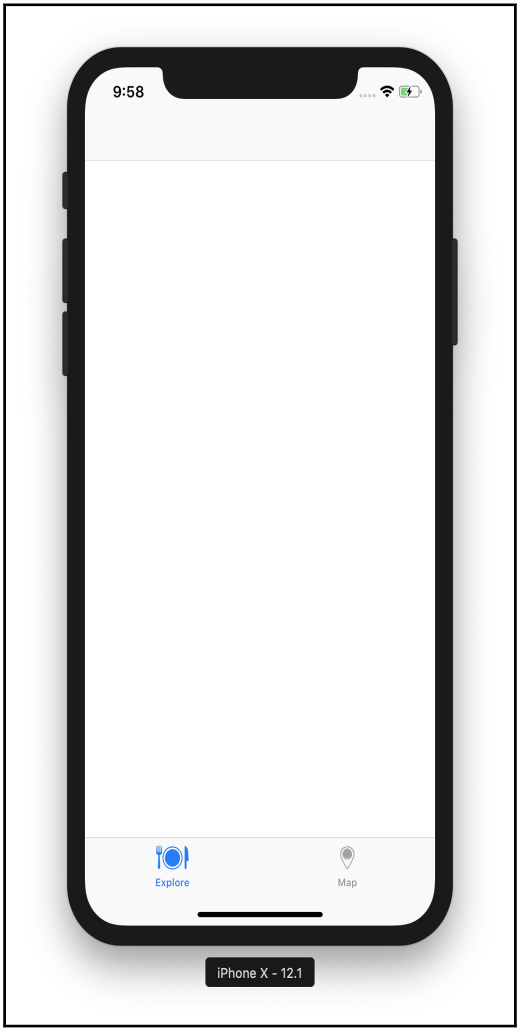



9:41 AM
LET'S EAT

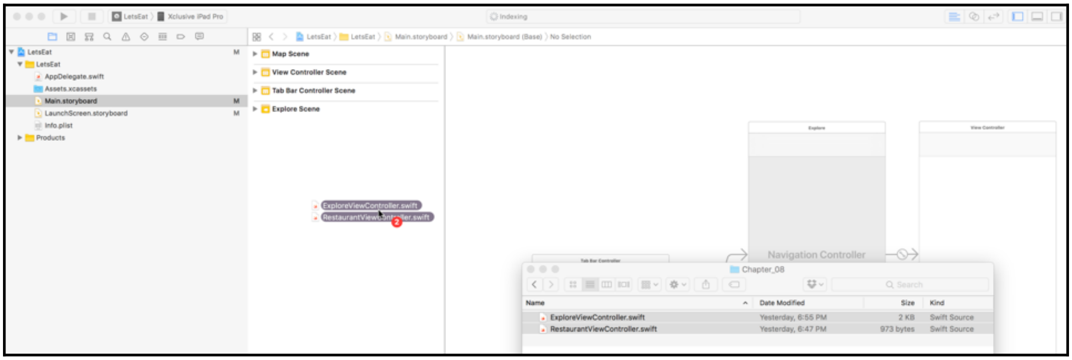

LetsEat M
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard M
LaunchScreen.storyboard M
Info.plist
Products
Map Scene
Item 2
Top Layout Guide
Bottom Layout Guide
View
Map
First Responder
Exit
Explore Scene
Item 1
Top Layout Guide
Bottom Layout Guide
View
Explore
First Responder
Exit
Tab Bar Controller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


Xcode
File
Edit
View
Find
Navigate
Editor
Product
Debug
Source Control
Window
Help
LetsEat
iPhone 7 Plu
Finished runnin
Main.storyboard
Main.stor
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

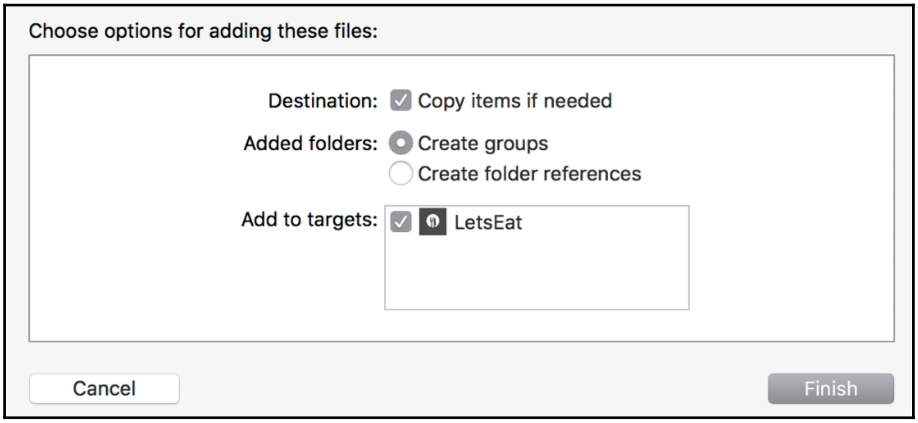

LetsEat
iPhone 7 Plus
Finished running LetsEat on iPhone 7 Plus
LetsEat
LetsEat
Main.storyboard
Main.storyboard (Base)
No Selection
LetsEat M
LetsEat
AppDelegate.swift
Assets.xcassets
Main.storyboard M
LaunchScreen.storyboard M
Info.plist
Products
Map Scene
Item 2
Top Layout Guide
Bottom Layout Guide
View
Map
First Responder
Exit
View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
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
Navigation Controller
Explore
Map
View as: iPhone 7 (wC hR)
100%


9:58
Explore
Map
iPhone X - 12.1

Chapter 8: Building Our App Structure in Storyboard



Name	Date Modified	Size	Kind
ExploreViewController.swift	Yesterday, 6:55 PM	2 KB	Swift Source
RestaurantViewController.swift	Yesterday, 6:47 PM	973 bytes	Swift Source

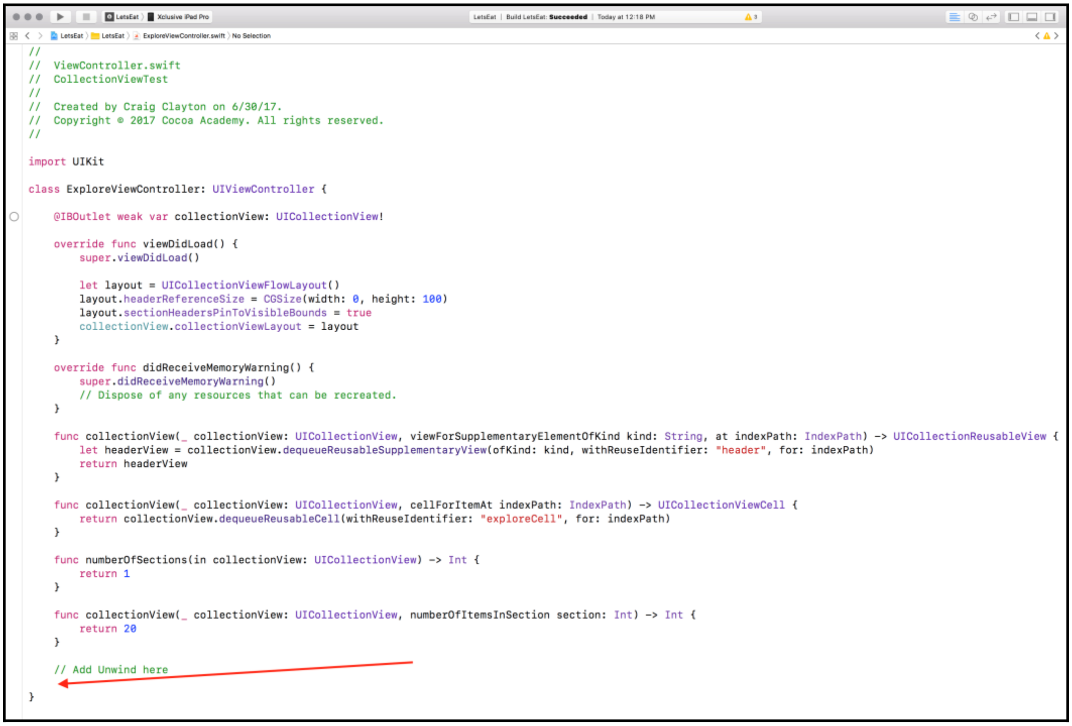


Choose options for adding these files:

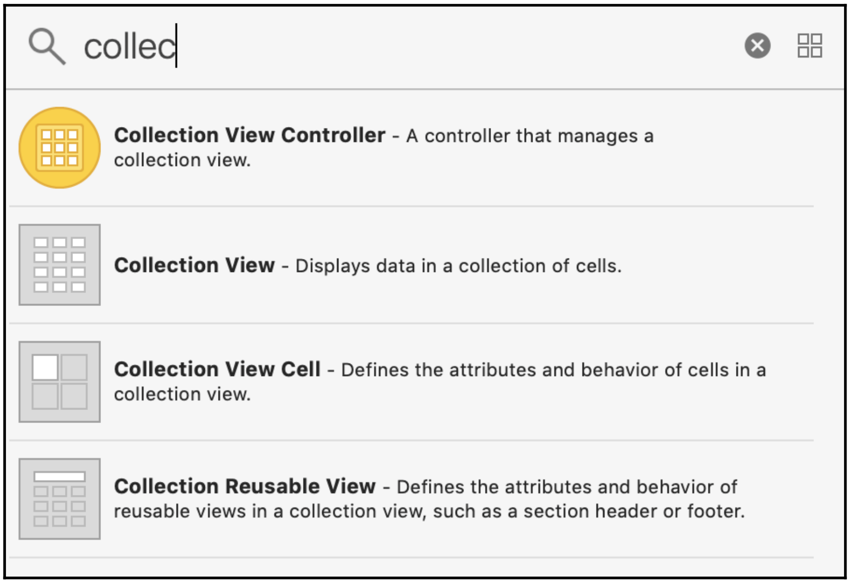
Destination: ☑ Copy items if needed

Added folders: ● Create groups
○ Create folder references

Add to targets: ☑ LetsEat


LetsEat
Xclusive iPad Pro
LetsEat | Build LetsEat: Succeeded | Today at 12:18 PM
LetsEat 〉 LetsEat 〉 ExploreViewController.swift 〉 No Selection

```
//
//  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

}
```

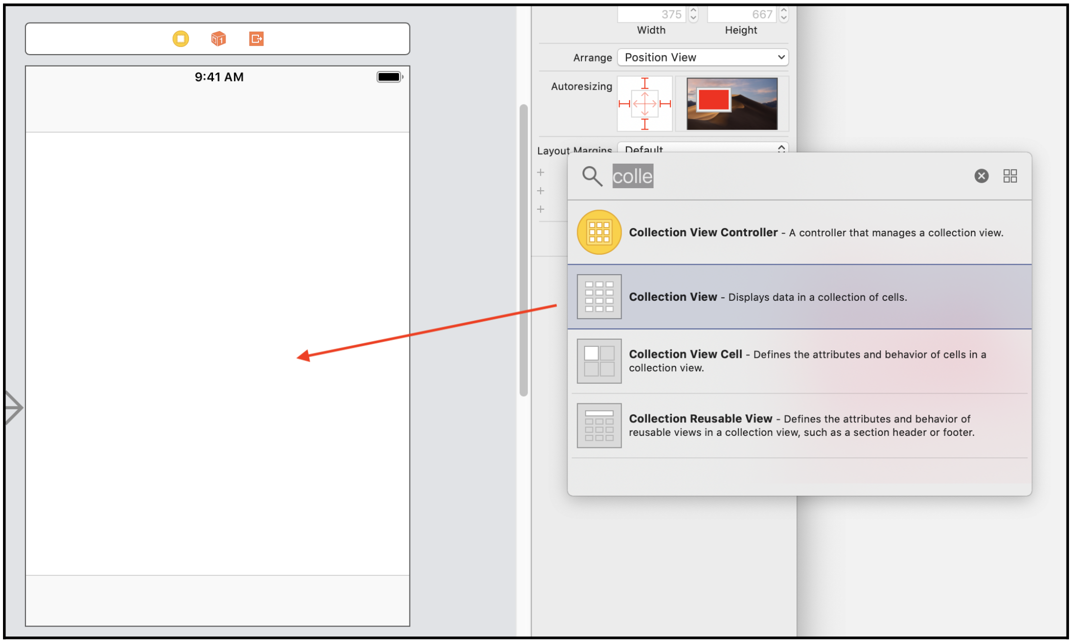


Collection View Controller - A controller that manages a collection view.

Collection View - Displays data in a collection of cells.

Collection View Cell - Defines the attributes and behavior of cells in a collection view.

Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a section header or footer.



375 Width

667 Height

Arrange: Position View

Layout Margins: Default

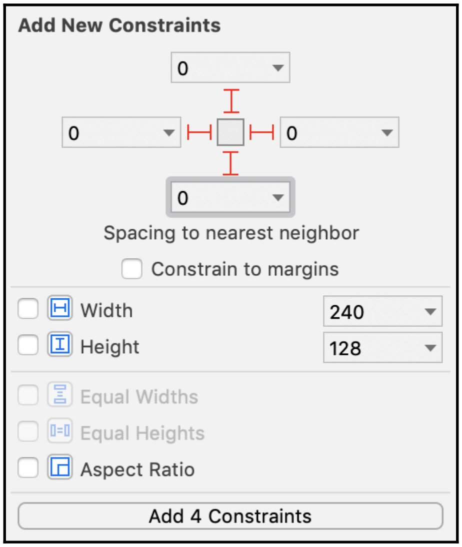
colle

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.

Collection Reusable View - Defines the attributes and behavior of reusable views in a collection view, such as a section header or footer.


Add New Constraints

0

0 0

0

Spacing to nearest neighbor

□ Constrain to margins

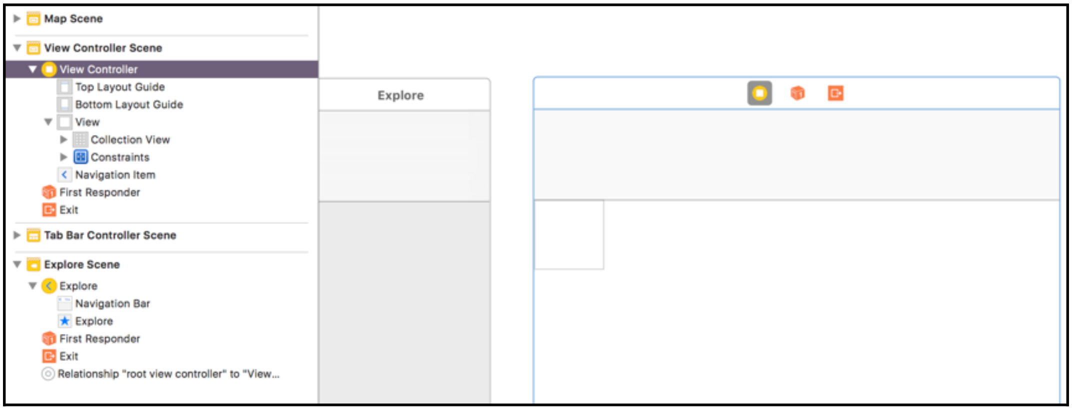
□ Width 240

□ Height 128

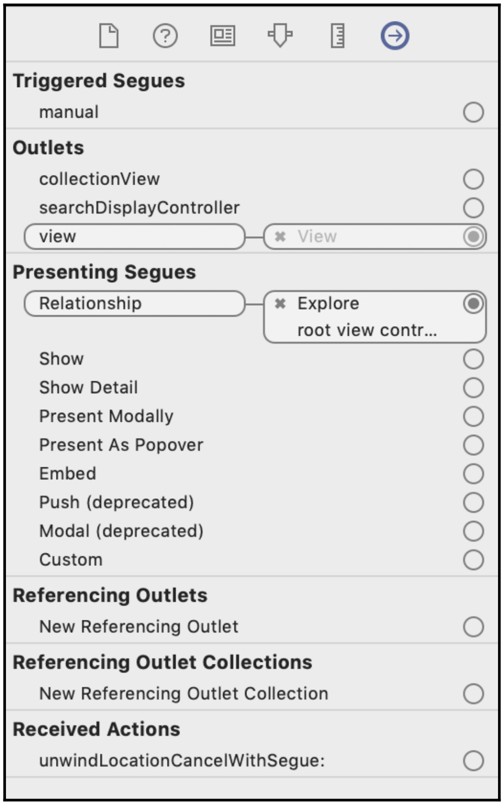
□ Equal Widths

□ Equal Heights

□ Aspect Ratio



- 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...



Triggered Segues

manual ○

Outlets

collectionView ○

searchDisplayController ○

view — ✖ View ●

Presenting Segues

Relationship — ✖ Explore root view contr... ●

Show ○

Show Detail ○

Present Modally ○

Present As Popover ○

Embed ○

Push (deprecated) ○

Modal (deprecated) ○

Custom ○

Referencing Outlets

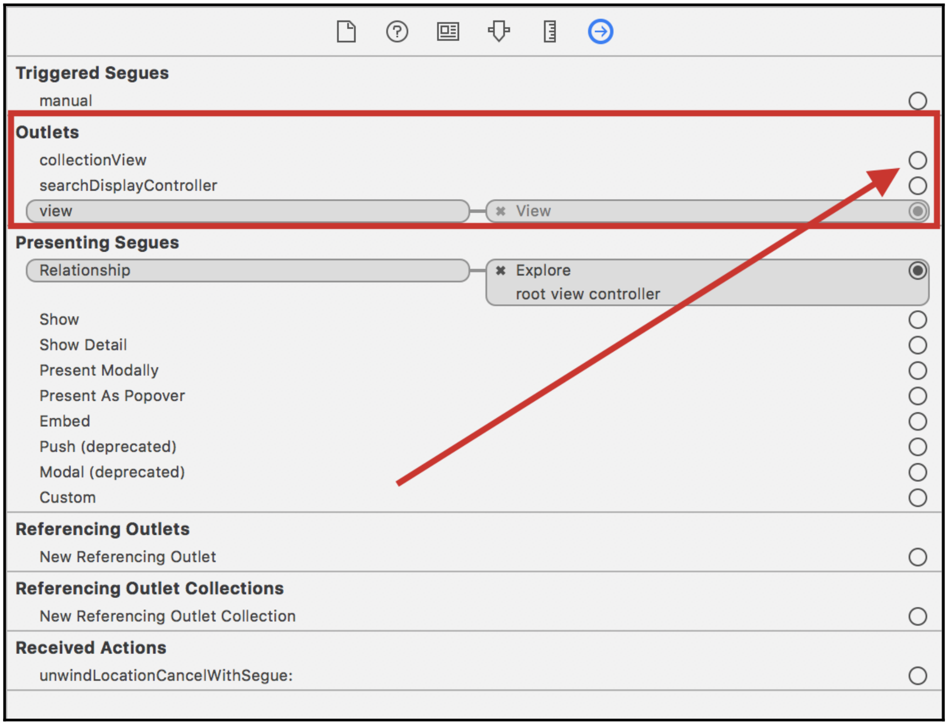
New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○

Received Actions

unwindLocationCancelWithSegue: ○



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

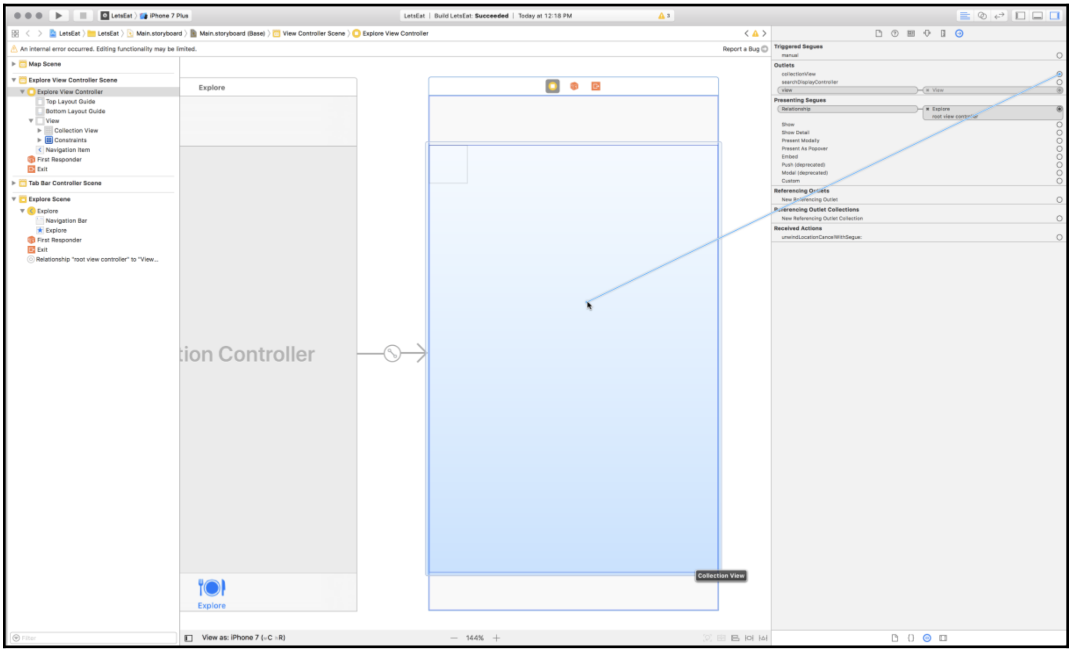
New Referencing Outlet ○

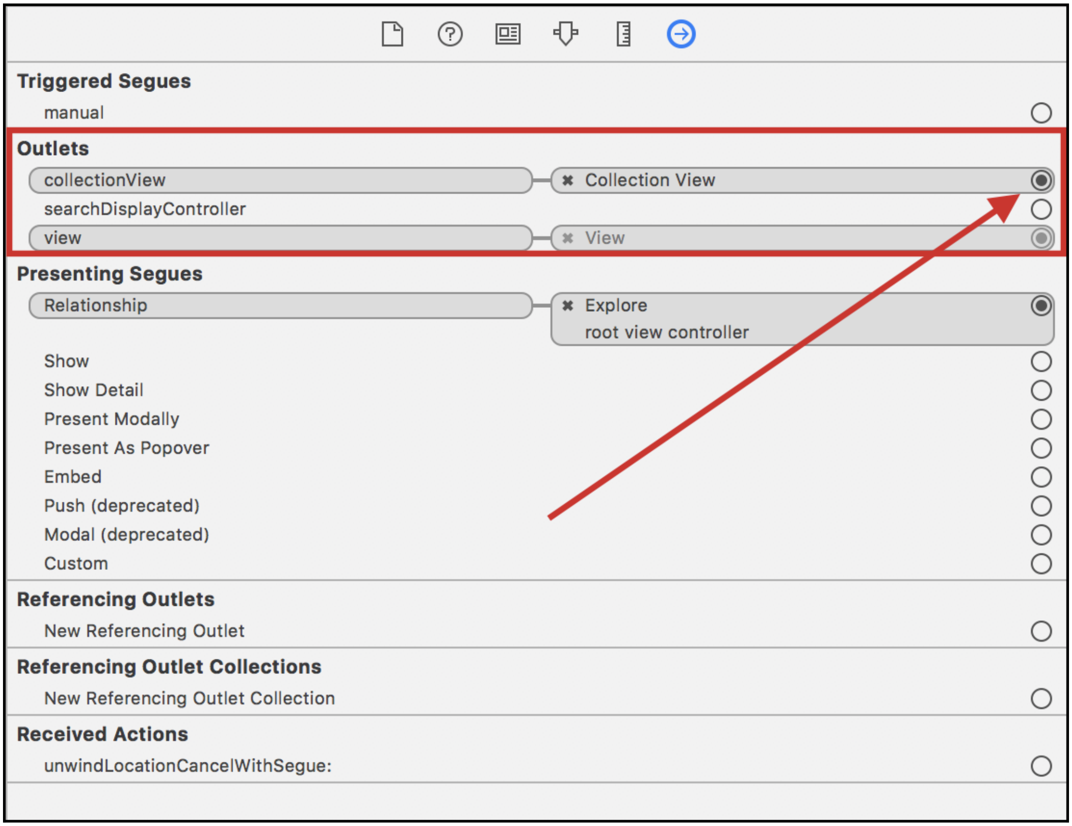
Referencing Outlet Collections

New Referencing Outlet Collection ○

Received Actions

unwindLocationCancelWithSegue: ○





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

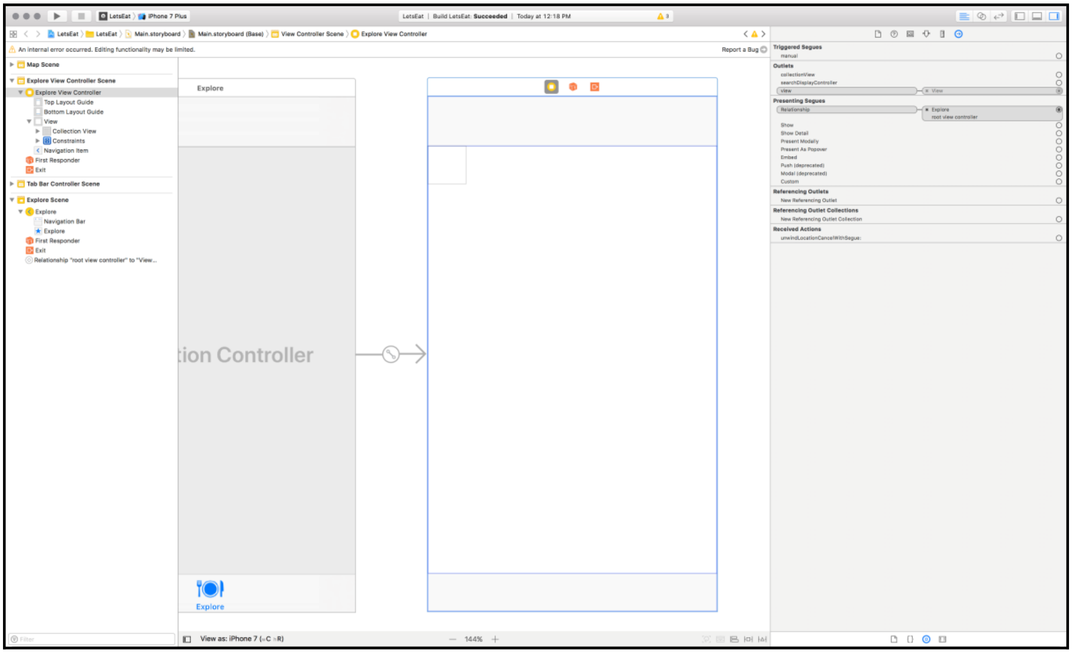
New Referencing Outlet ○

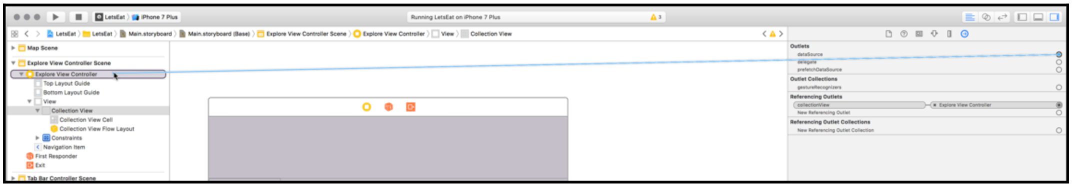
Referencing Outlet Collections

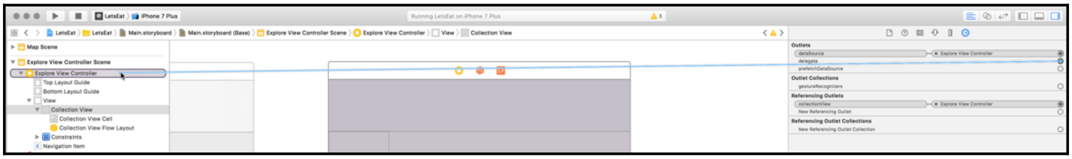
New Referencing Outlet Collection ○

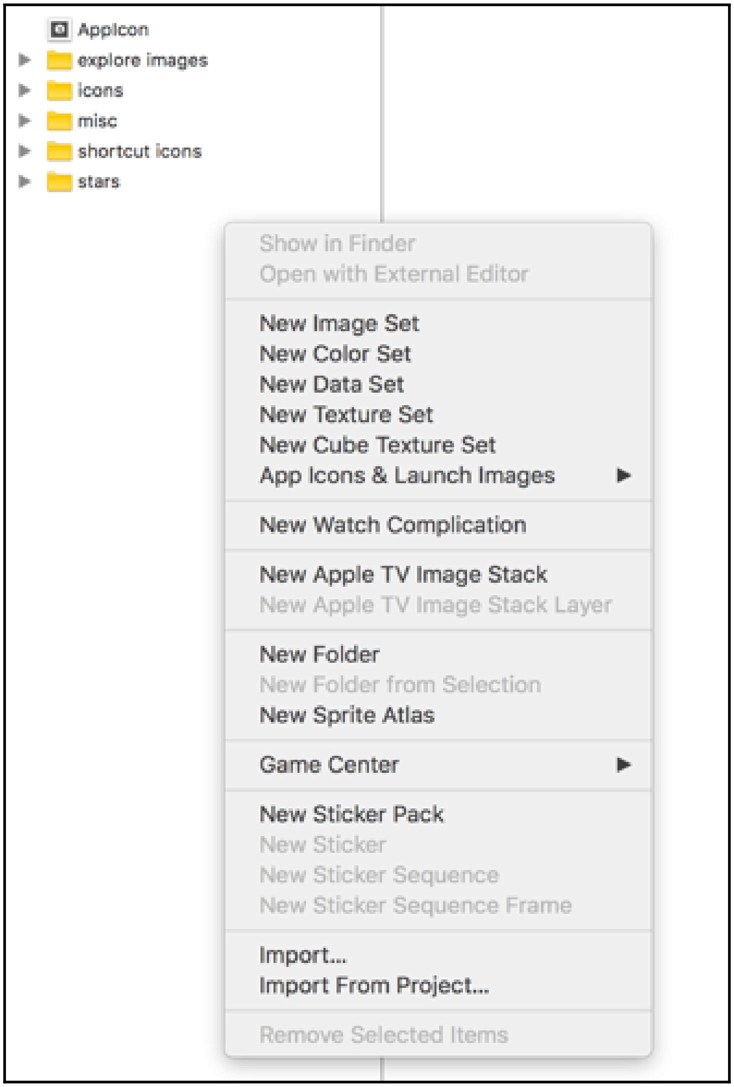
Received Actions

unwindLocationCancelWithSegue: ○

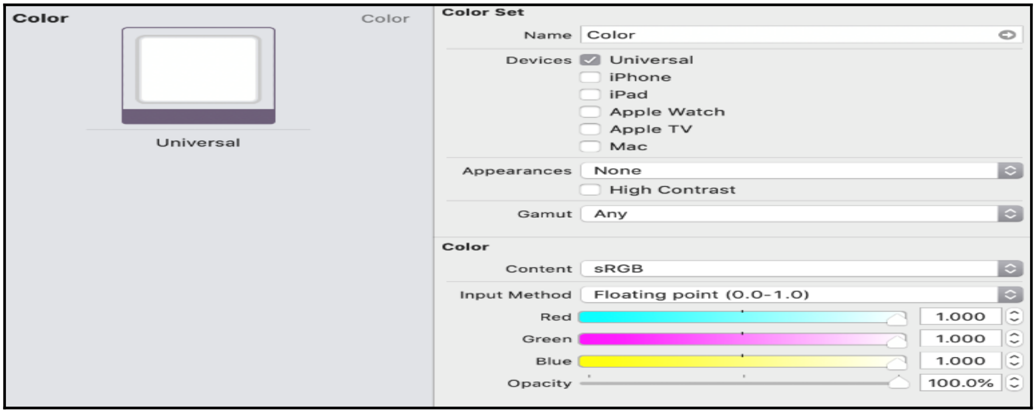

LetsEat 〉 iPhone 7 Plus
LetsEat | Build LetsEat: Succeeded | Today at 12:18 PM
LetsEat 〉 LetsEat 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 View Controller Scene 〉 Explore View Controller
An internal error occurred. Editing functionality may be limited.
Report a Bug
Map Scene
Explore View Controller Scene
Explore 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
tion Controller
Explore
View as: iPhone 7 (wC hR)
144%
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:


LetsEat 〉 iPhone 7 Plus
Running LetsEat on iPhone 7 Plus
LetsEat 〉 LetsEat 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 Explore View Controller Scene 〉 Explore View Controller 〉 View 〉 Collection View
Map Scene
Explore View Controller Scene
Explore View Controller
Top Layout Guide
Bottom Layout Guide
View
Collection View
Collection View Cell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Tab Bar Controller Scene
Outlets
dataSource
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
collectionView
Explore View Controller
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


LetsEat 〉 iPhone 7 Plus
Running LetsEat on iPhone 7 Plus
LetsEat 〉 LetsEat 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 Explore View Controller Scene 〉 Explore View Controller 〉 View 〉 Collection View
Map Scene
Explore View Controller Scene
Explore View Controller
Top Layout Guide
Bottom Layout Guide
View
Collection View
Collection View Cell
Collection View Flow Layout
Constraints
Navigation Item
Outlets
dataSource
Explore View Controller
delegate
prefetchDataSource
Outlet Collections
gestureRecognizers
Referencing Outlets
collectionView
Explore View Controller
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection



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



Color Color

Universal

Color Set

Name: Color

Devices: ☑ Universal
□ iPhone
□ iPad
□ Apple Watch
□ Apple TV
□ Mac

Appearances: None
□ High Contrast

Gamut: Any

Color

Content: sRGB

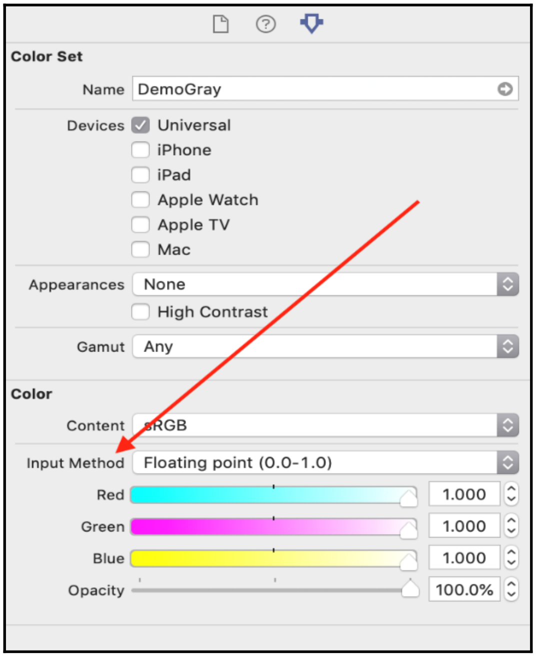
Input Method: Floating point (0.0-1.0)

Red: 1.000

Green: 1.000

Blue: 1.000

Opacity: 100.0%



Color Set

Name: DemoGray

Devices: ☑ Universal
□ iPhone
□ iPad
□ Apple Watch
□ Apple TV
□ Mac

Appearances: None
□ High Contrast

Gamut: Any

Color

Content: sRGB

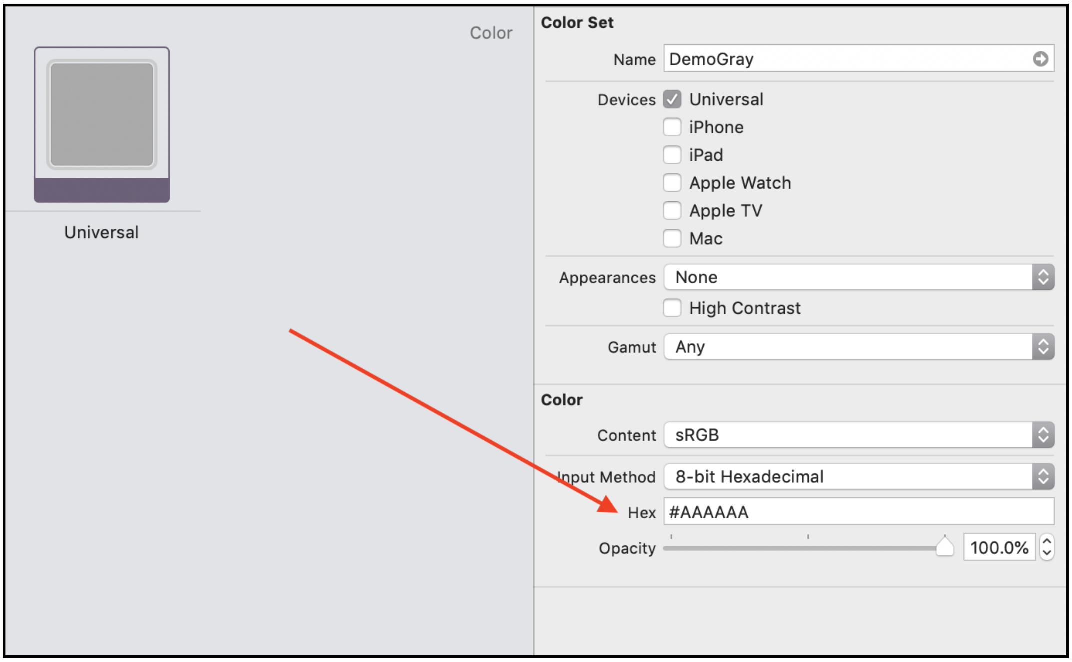
Input Method: Floating point (0.0-1.0)

Red: 1.000

Green: 1.000

Blue: 1.000

Opacity: 100.0%



Color Set

Name DemoGray

Devices ☑ Universal
□ iPhone
□ iPad
□ Apple Watch
□ Apple TV
□ Mac

Appearances None
□ High Contrast

Gamut Any

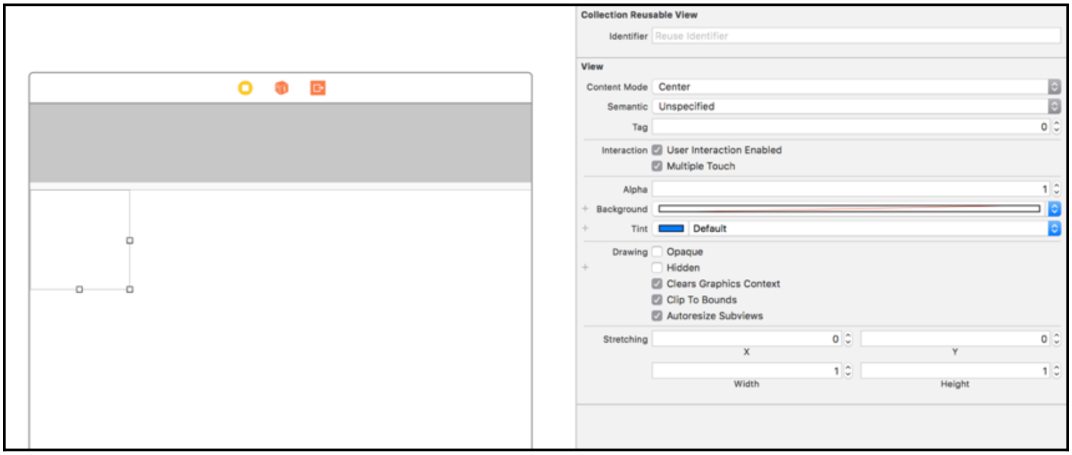
Color

Content sRGB

Input Method 8-bit Hexadecimal

Hex #AAAAAA

Opacity 100.0%



Collection Reusable View

Identifier Reuse Identifier

View

Content Mode Center

Semantic Unspecified

Tag 0

Interaction ☑ User Interaction Enabled
☑ Multiple Touch

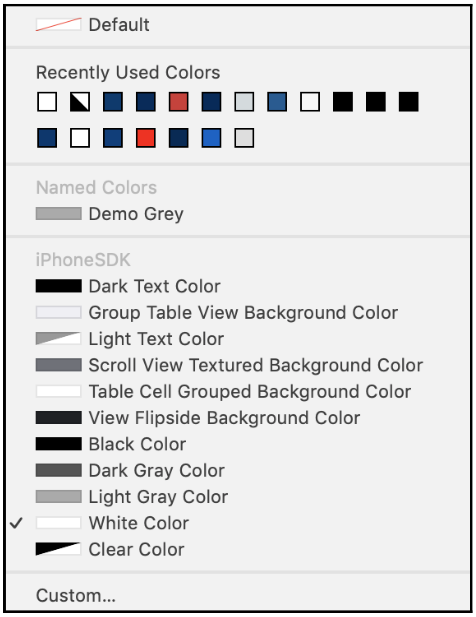
Alpha 1

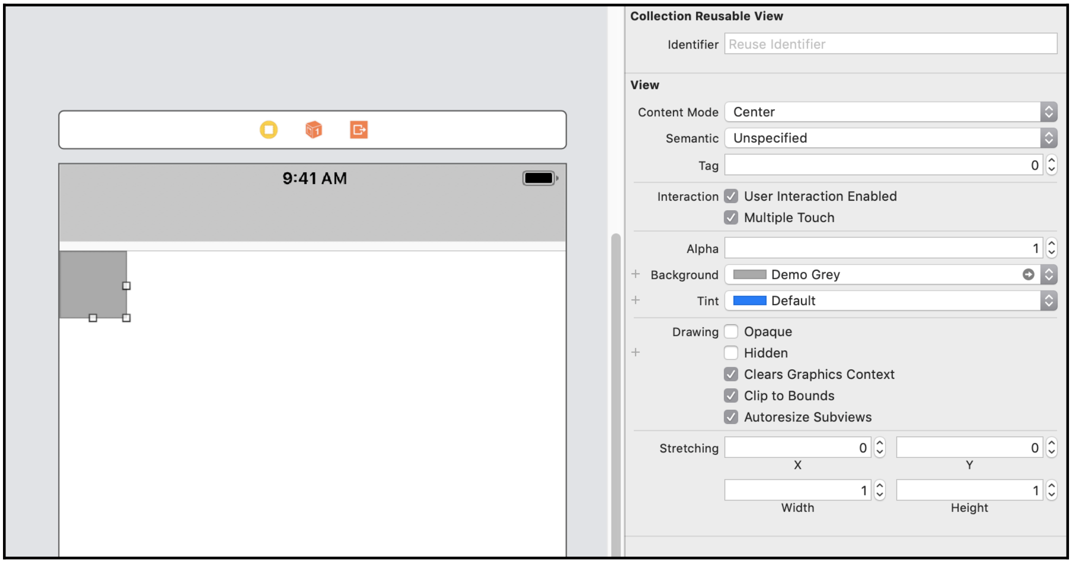
Background

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

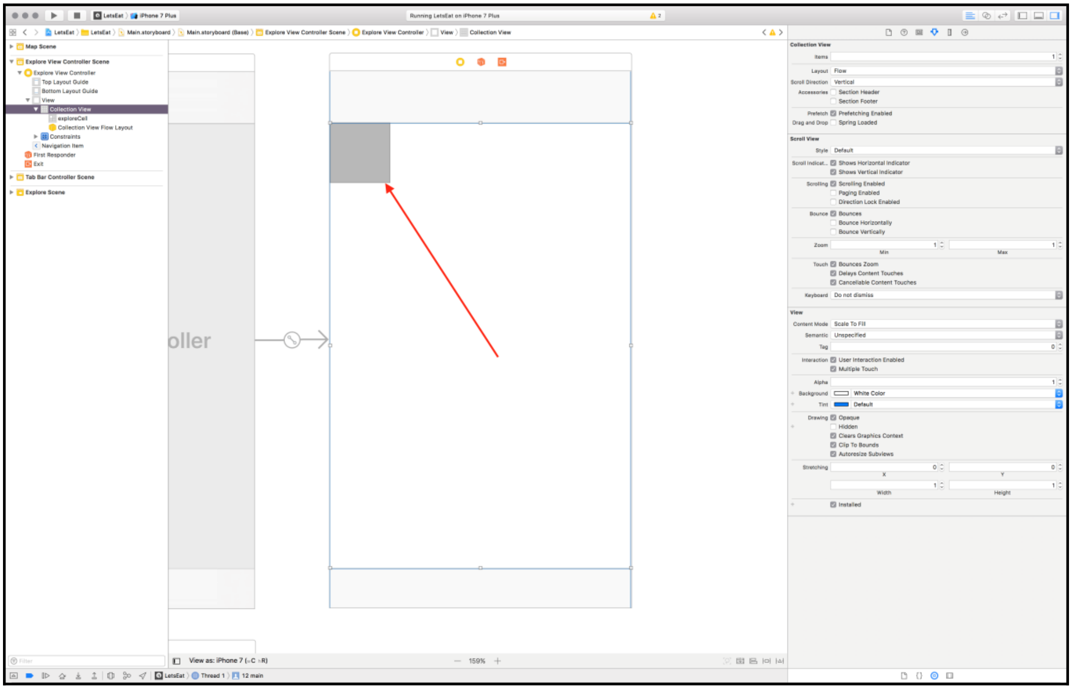
Alpha: 1

Background: Demo Grey

Tint: Default

Drawing: □ Opaque
□ Hidden
☑ Clears Graphics Context
☑ Clip to Bounds
☑ Autoresize Subviews

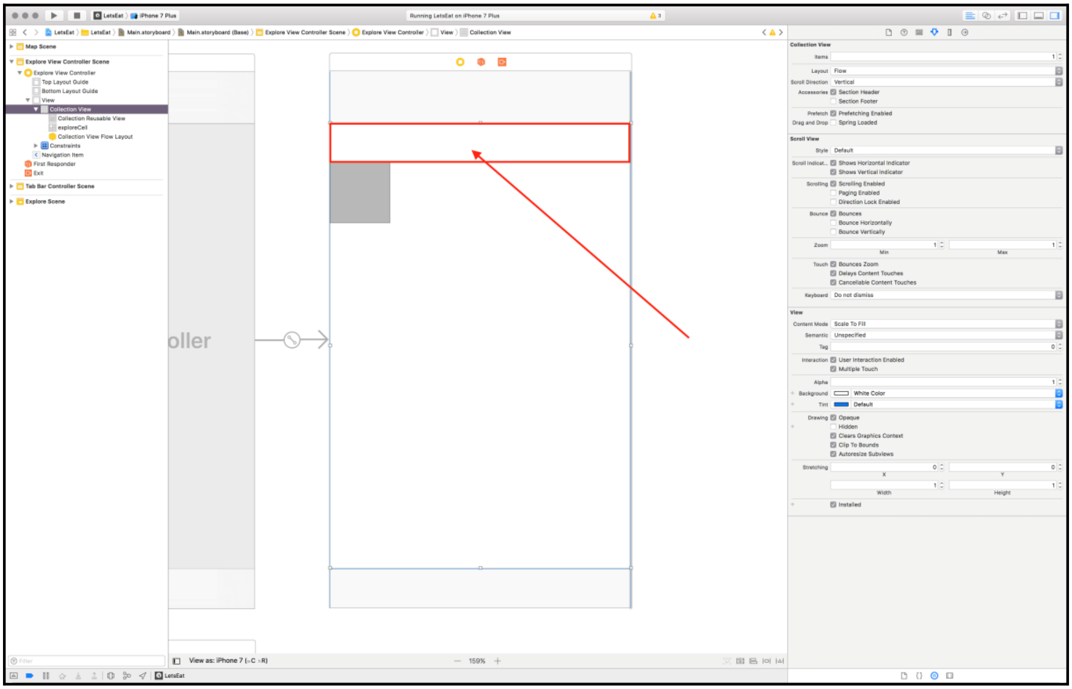
Stretching: X 0, Y 0, Width 1, Height 1


LetsEat
iPhone 7 Plus
Running LetsEat on iPhone 7 Plus
LetsEat 〉 LetsEat 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 Explore View Controller Scene 〉 Explore View Controller 〉 View 〉 Collection View
Map Scene
Explore View Controller Scene
Explore View Controller
Top Layout Guide
Bottom Layout Guide
View
Collection View
exploreCell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Tab Bar Controller Scene
Explore Scene
oller
View as: iPhone 7 (wC hR)
159%
LetsEat 〉 Thread 1 〉 12 main

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

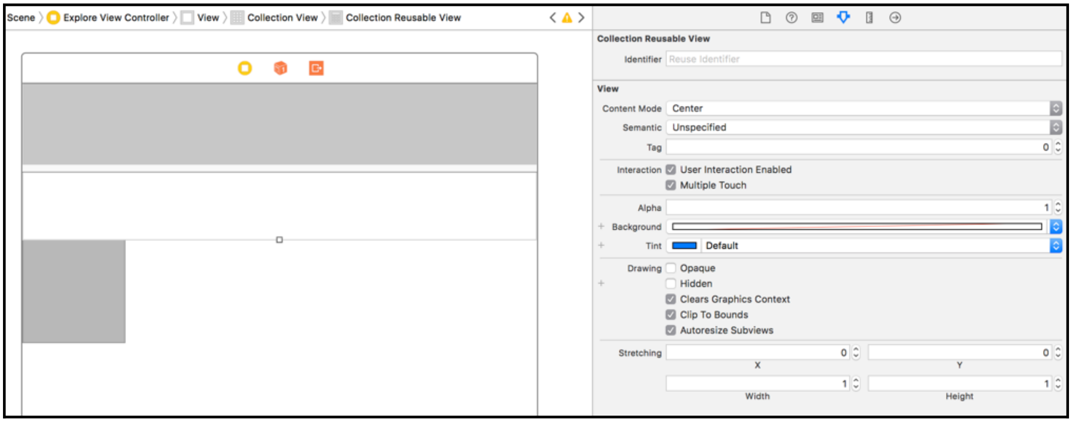
View
Content Mode Scale To Fill
Semantic Unspecified
Tag 0
Interaction ☑ User Interaction Enabled
☑ Multiple Touch
Alpha 1
Background White Color
Tint Default
Drawing ☑ Opaque
□ Hidden
☑ Clears Graphics Context
☑ Clip To Bounds
☑ Autoresize Subviews
Stretching 0 (X) 0 (Y)
1 (Width) 1 (Height)
☑ Installed


LetsEat
iPhone 7 Plus
Running LetsEat on iPhone 7 Plus
LetsEat 〉 LetsEat 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 Explore View Controller Scene 〉 Explore View Controller 〉 View 〉 Collection View
Map Scene
Explore View Controller Scene
Explore View Controller
Top Layout Guide
Bottom Layout Guide
View
Collection View
Collection Reusable View
exploreCell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Tab Bar Controller Scene
Explore Scene
oller
View as: iPhone 7 (wC hR)
159%
LetsEat

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
☑ Multiple Touch
Alpha 1
Background White Color
Tint Default
Drawing ☑ Opaque
□ Hidden
☑ Clears Graphics Context
☑ Clip To Bounds
☑ Autoresize Subviews
Stretching 0 (X) 0 (Y)
1 (Width) 1 (Height)
☑ Installed



Scene 〉 Explore View Controller 〉 View 〉 Collection View 〉 Collection Reusable View

Collection Reusable View

Identifier: Reuse Identifier

View

Content Mode: Center

Semantic: Unspecified

Tag: 0

Interaction: ☑ User Interaction Enabled
☑ Multiple Touch

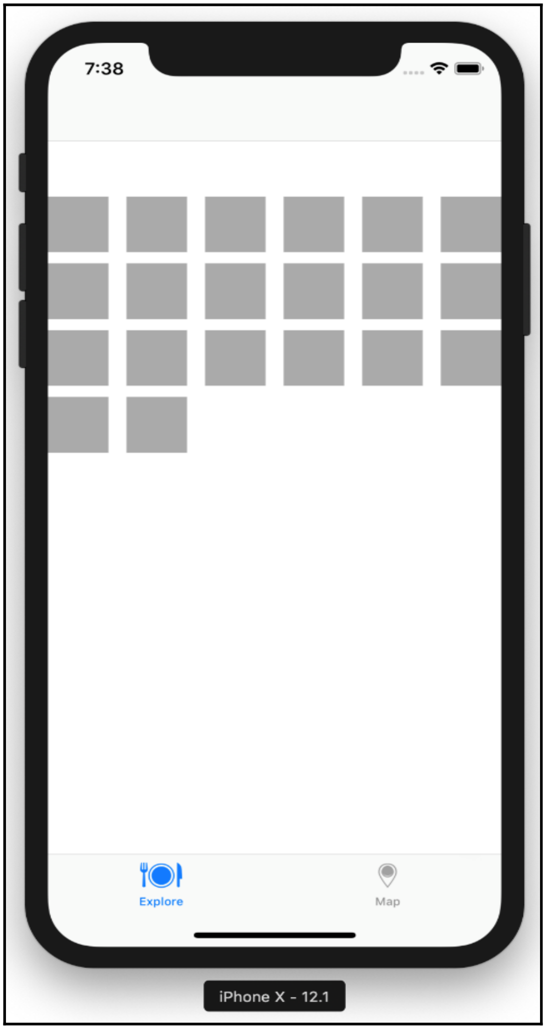
Alpha: 1

Background

Tint: Default

Drawing: □ Opaque
□ Hidden
☑ Clears Graphics Context
☑ Clip To Bounds
☑ Autoresize Subviews

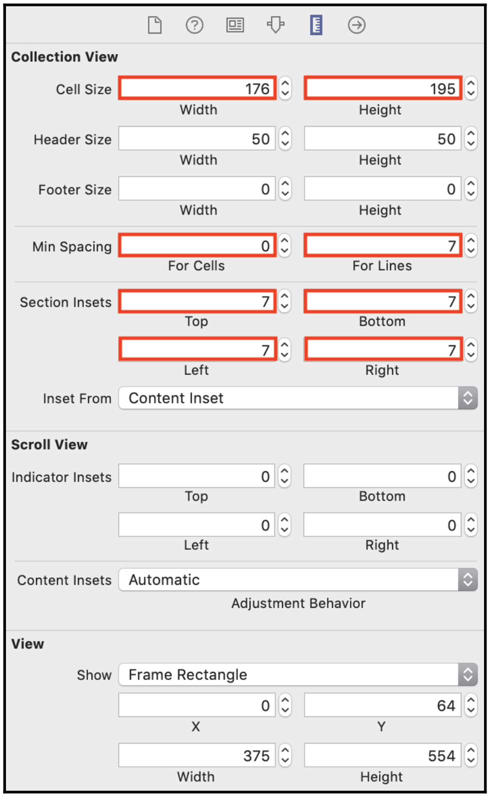
Stretching: X 0, Y 0
Width 1, Height 1



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



Collection View

Cell Size: Width 176, Height 195

Header Size: Width 50, Height 50

Footer Size: Width 0, Height 0

Min Spacing: For Cells 0, For Lines 7

Section Insets: Top 7, Bottom 7, Left 7, Right 7

Inset From: Content Inset

Scroll View

Indicator Insets: Top 0, Bottom 0, Left 0, Right 0

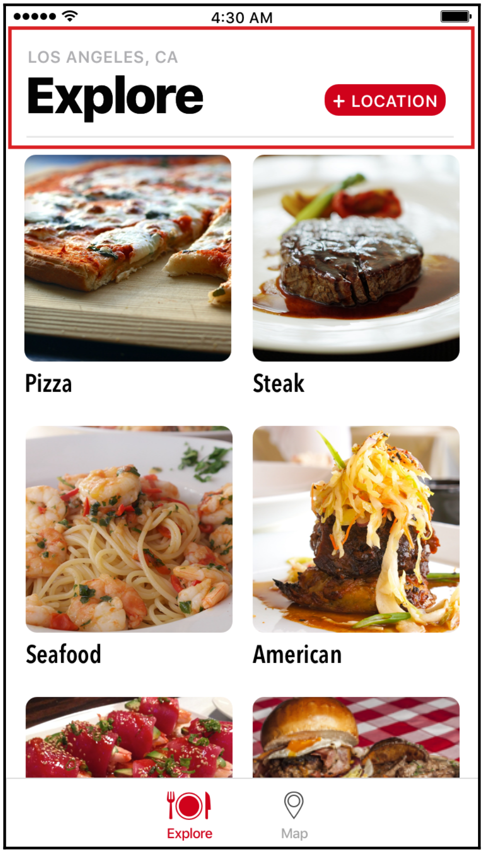
Content Insets: Automatic (Adjustment Behavior)

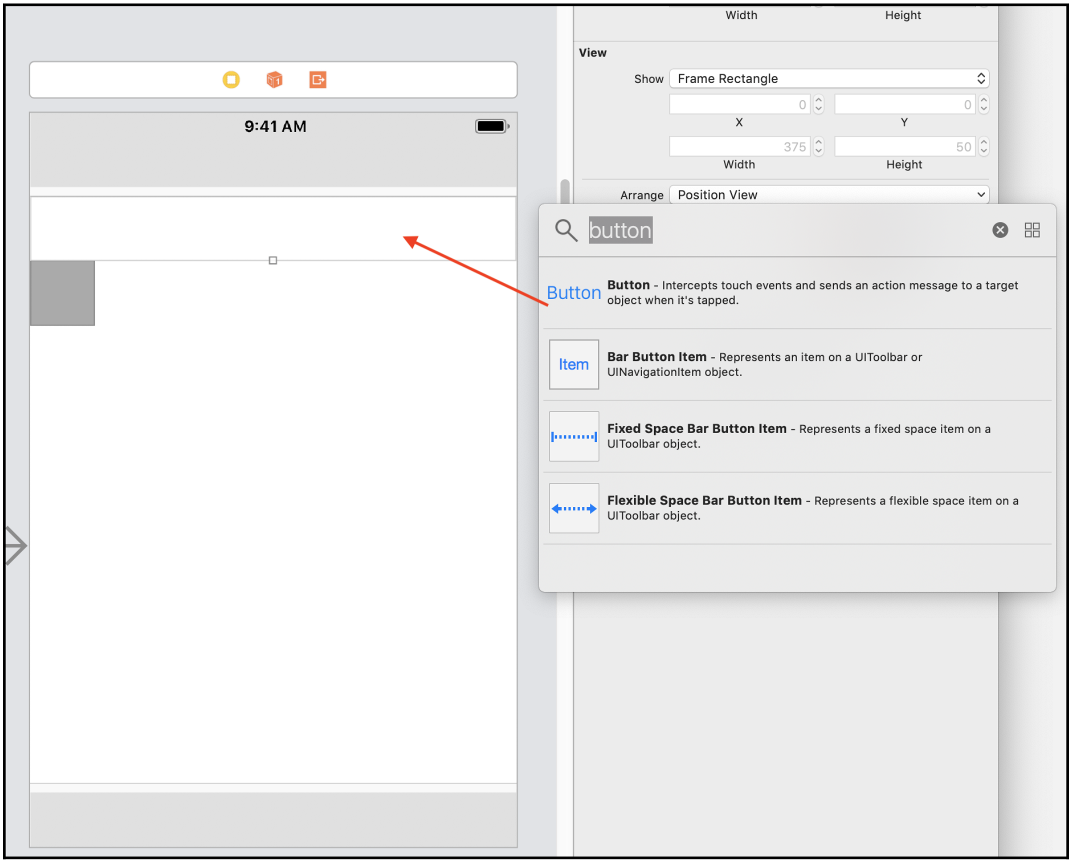
View

Show: Frame Rectangle

X 0, Y 64

Width 375, Height 554


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

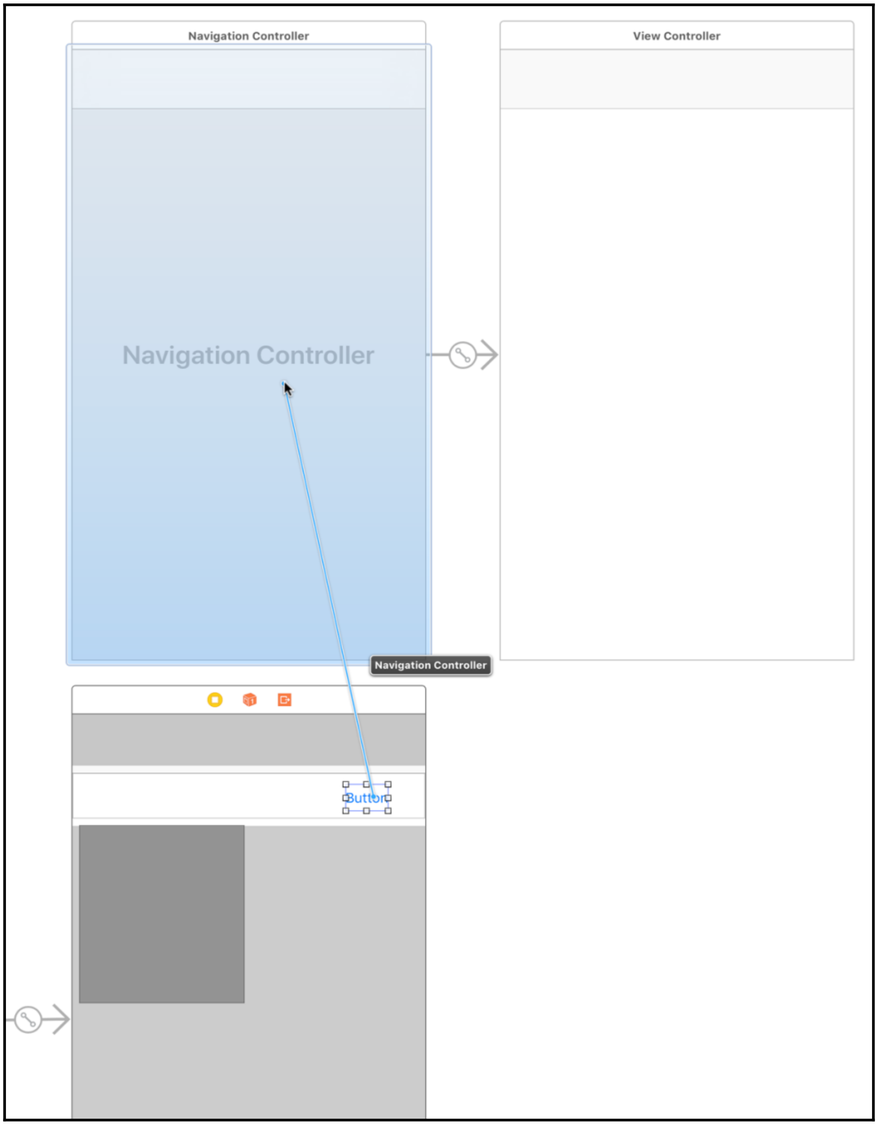


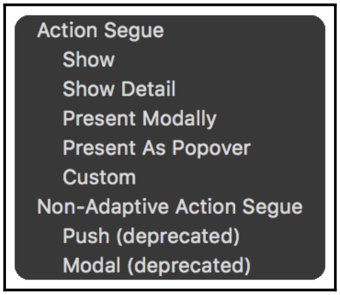
Button - Intercepts touch events and sends an action message to a target object when it's tapped.

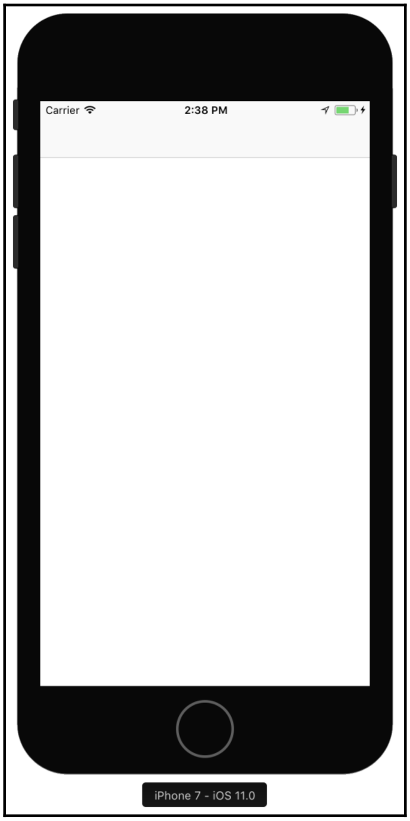
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.

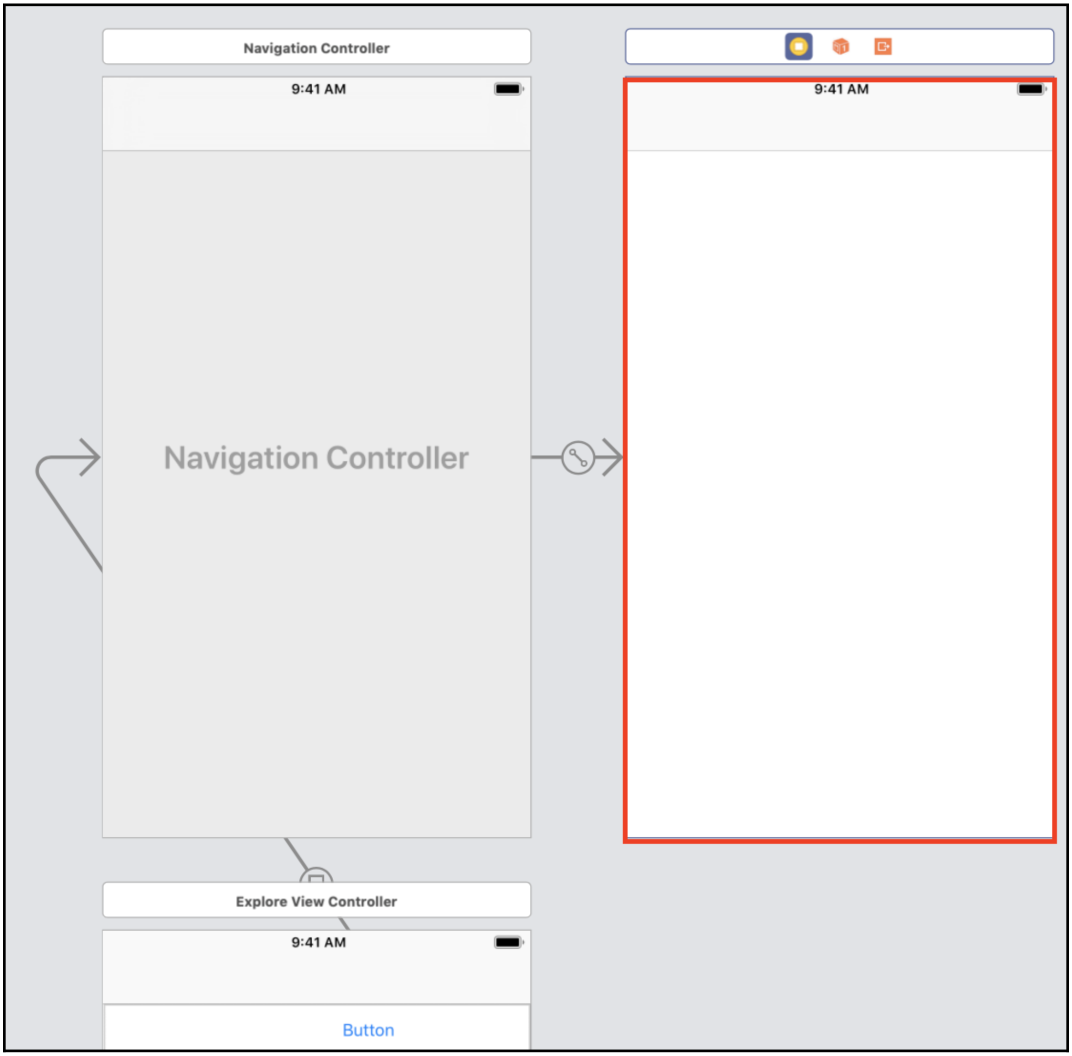
Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.

Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.

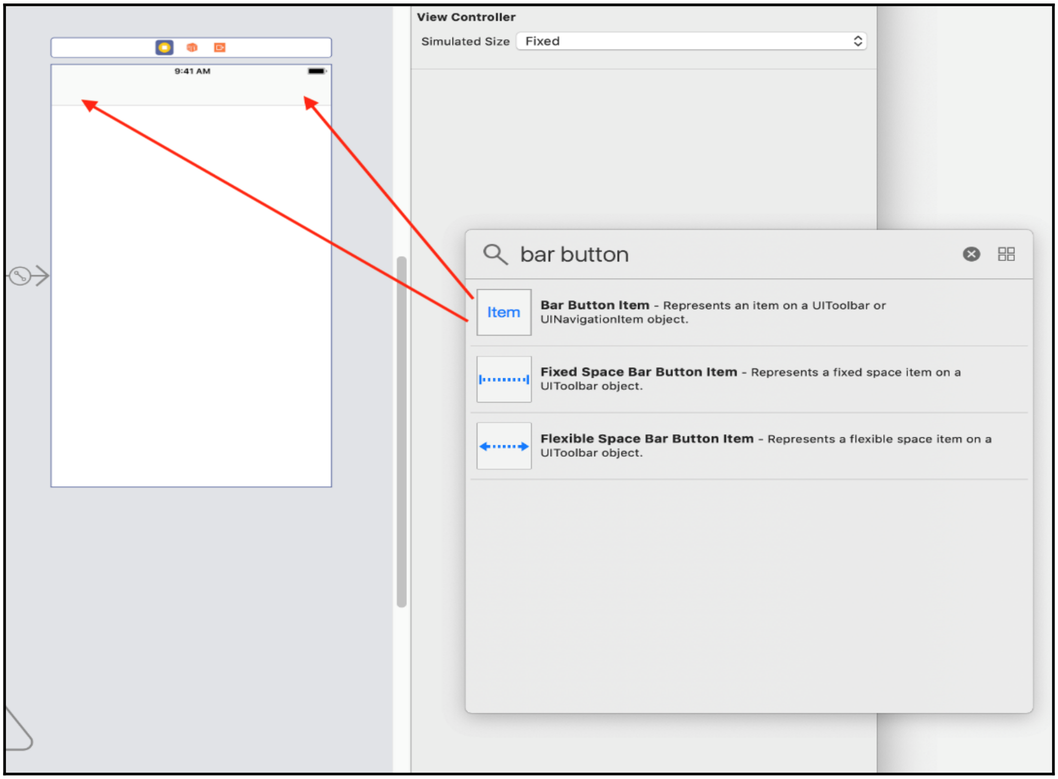

Navigation Controller
View Controller
Navigation Controller
Navigation Controller
Button


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
9:41 AM
Navigation Controller
9:41 AM
Explore View Controller
9:41 AM
Button



View Controller

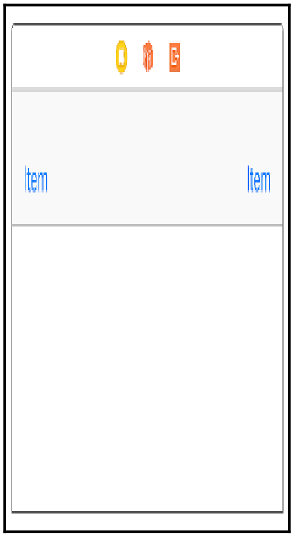
Simulated Size Fixed

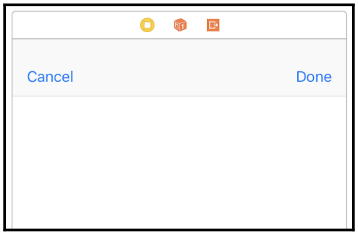
bar button

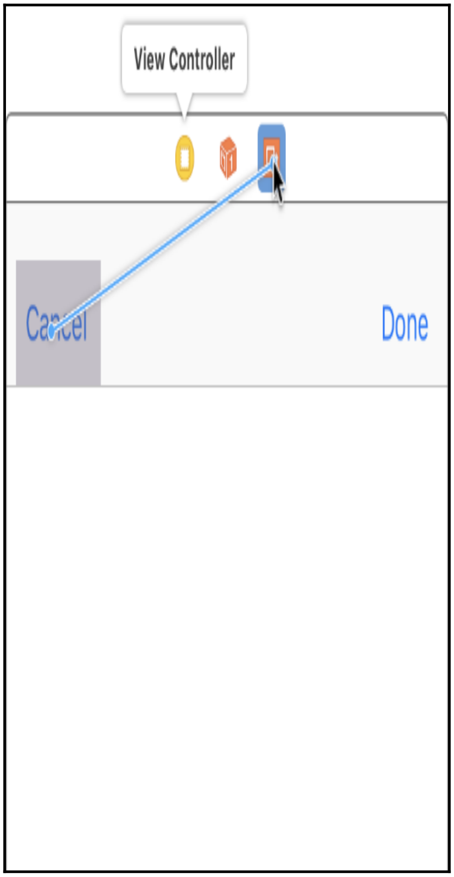
Bar Button Item - Represents an item on a UIToolbar or UINavigationItem object.

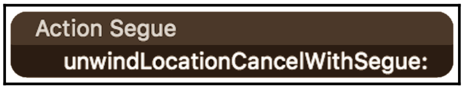
Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.

Flexible Space Bar Button Item - Represents a flexible space item on a UIToolbar object.




Cancel
Done


View Controller
Cancel
Done


Action Segue
unwindLocationCancelWithSegue:

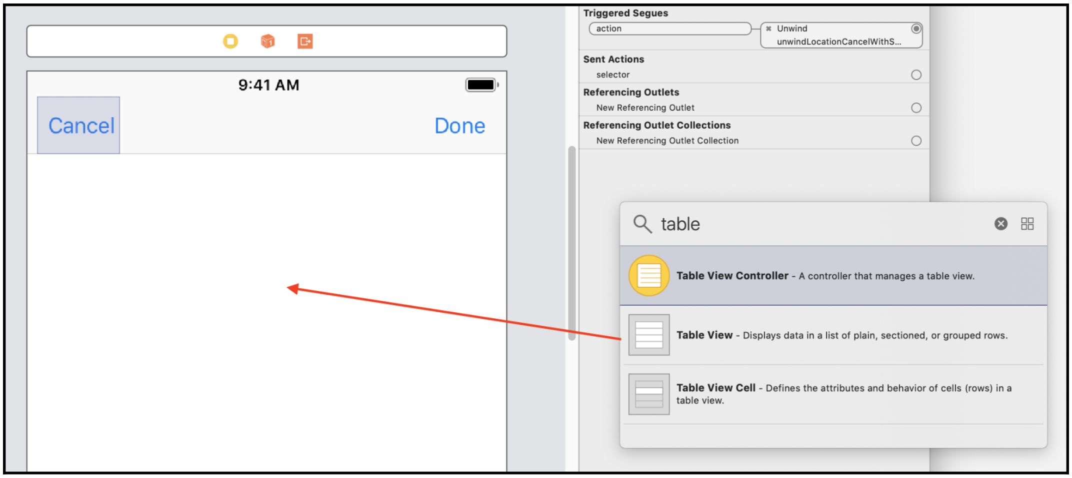
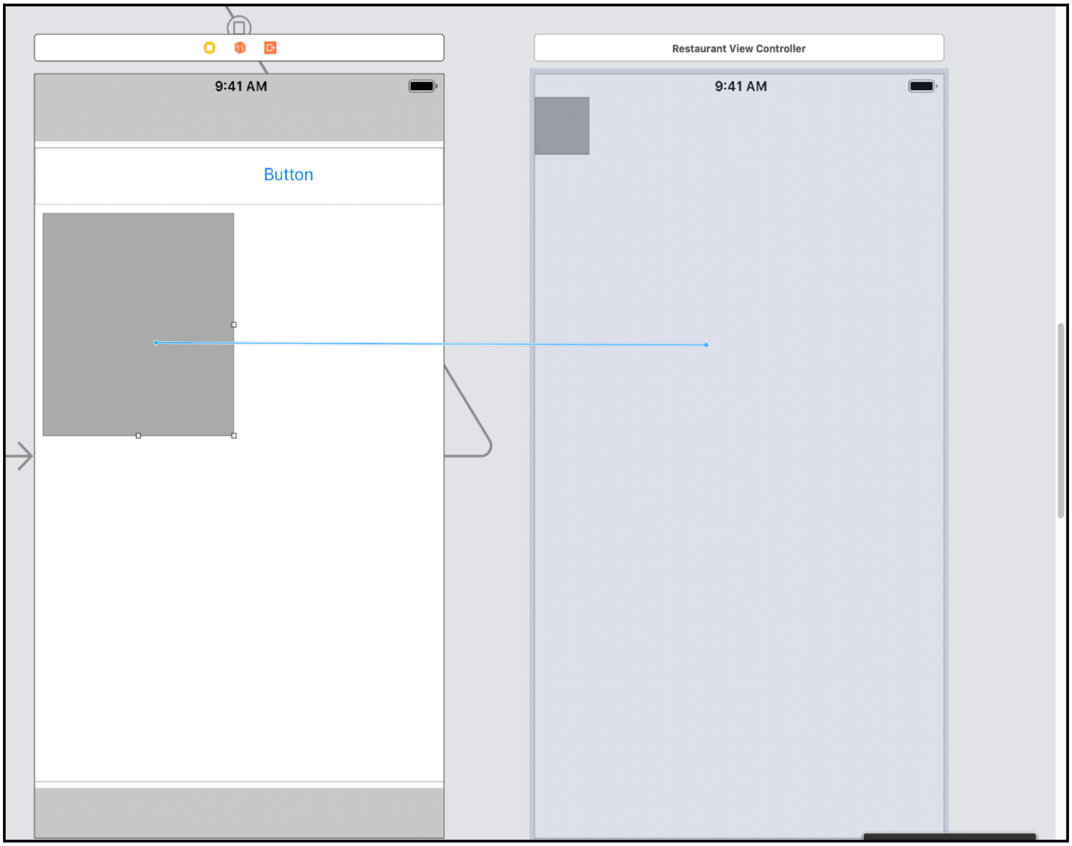


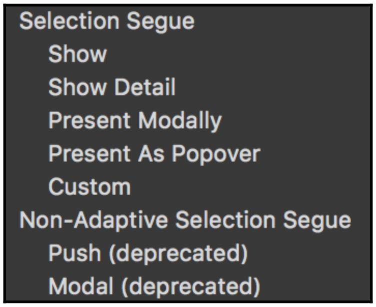
Table View Controller - A controller that manages a table view.

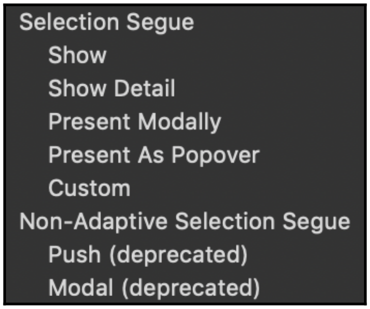
Table View - Displays data in a list of plain, sectioned, or grouped rows.

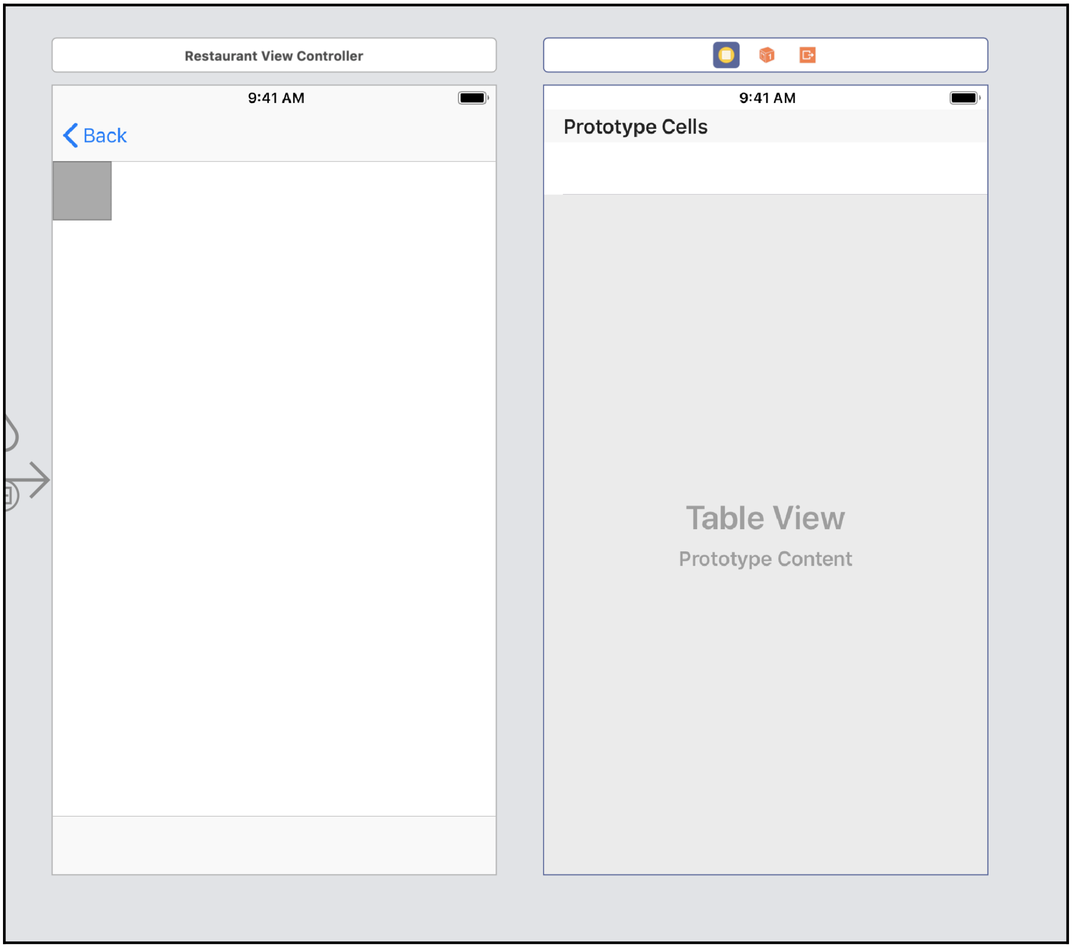
Table View Cell - Defines the attributes and behavior of cells (rows) in a table view.

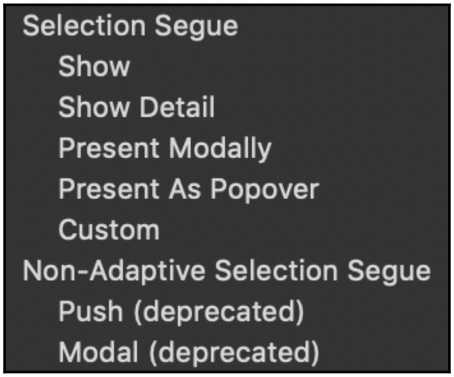
Chapter 9:
Finishing Up Our App Structure in Storyboard

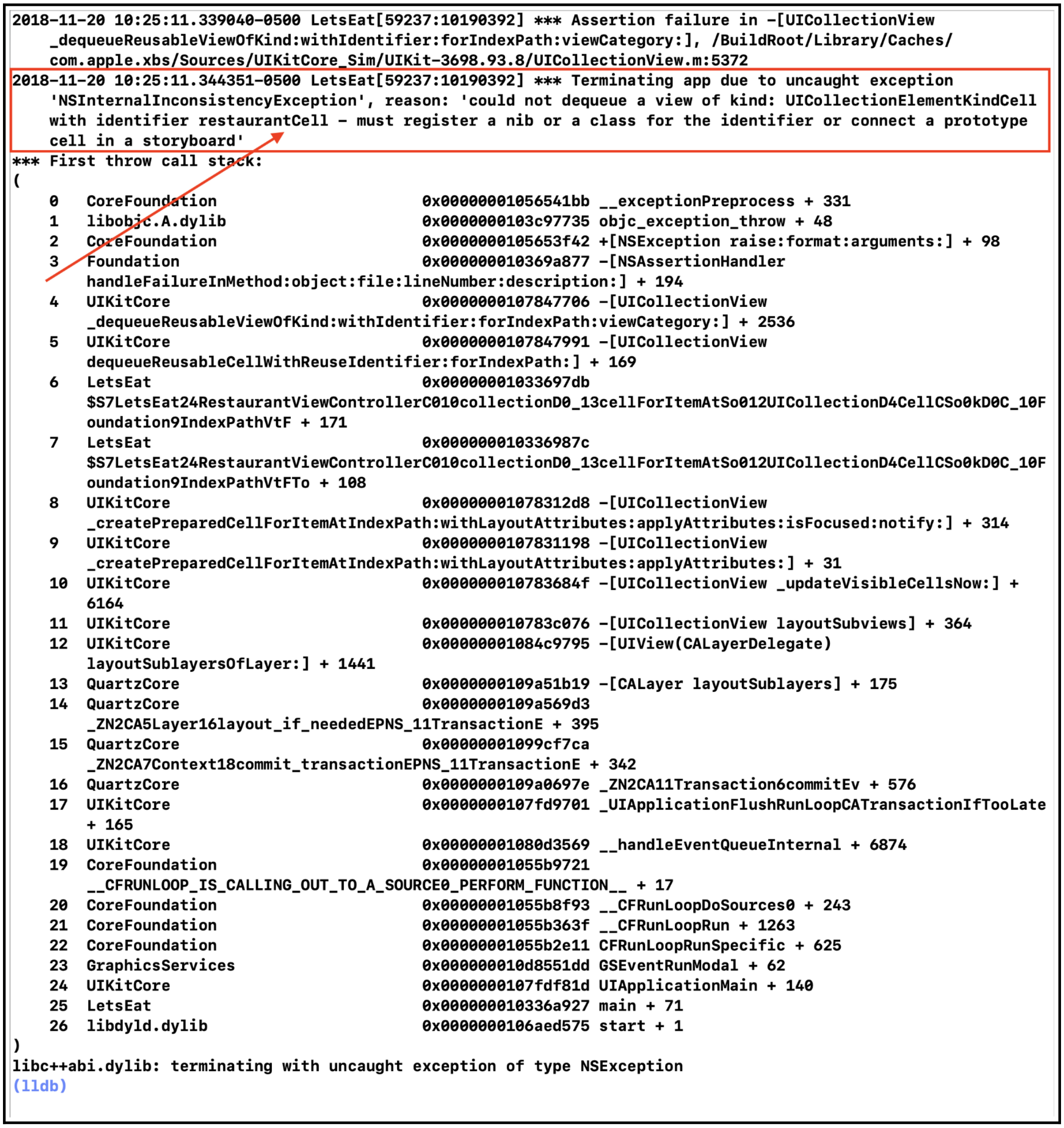





Selection Segue
Show
Show Detail
Present Modally
Present As Popover
Custom
Non-Adaptive Selection Segue
Push (deprecated)
Modal (deprecated)


Restaurant View Controller
9:41 AM
Back
9:41 AM
Prototype Cells
Table View
Prototype Content


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



```
2018-11-20 10:25:11.339040-0500 LetsEat[59237:10190392] *** Assertion failure in -[UICollectionView
    _dequeueReusableViewOfKind:withIdentifier:forIndexPath:viewCategory:], /BuildRoot/Library/Caches/
    com.apple.xbs/Sources/UIKitCore_Sim/UIKit-3698.93.8/UICollectionView.m:5372
2018-11-20 10:25:11.344351-0500 LetsEat[59237:10190392] *** Terminating app due to uncaught exception
    'NSInternalInconsistencyException', reason: 'could not dequeue a view of kind: UICollectionElementKindCell
    with identifier restaurantCell - must register a nib or a class for the identifier or connect a prototype
    cell in a storyboard'
*** First throw call stack:
(
    0   CoreFoundation                      0x00000001056541bb __exceptionPreprocess + 331
    1   libobjc.A.dylib                     0x0000000103c97735 objc_exception_throw + 48
    2   CoreFoundation                      0x0000000105653f42 +[NSException raise:format:arguments:] + 98
    3   Foundation                          0x000000010369a877 -[NSAssertionHandler
    handleFailureInMethod:object:file:lineNumber:description:] + 194
    4   UIKitCore                           0x0000000107847706 -[UICollectionView
    _dequeueReusableViewOfKind:withIdentifier:forIndexPath:viewCategory:] + 2536
    5   UIKitCore                           0x0000000107847991 -[UICollectionView
    dequeueReusableCellWithReuseIdentifier:forIndexPath:] + 169
    6   LetsEat                             0x00000001033697db
    $S7LetsEat24RestaurantViewControllerC010collectionD0_13cellForItemAtSo012UICollectionD4CellCSo0kD0C_10F
    oundation9IndexPathVtF + 171
    7   LetsEat                             0x000000010336987c
    $S7LetsEat24RestaurantViewControllerC010collectionD0_13cellForItemAtSo012UICollectionD4CellCSo0kD0C_10F
    oundation9IndexPathVtFTo + 108
    8   UIKitCore                           0x00000001078312d8 -[UICollectionView
    _createPreparedCellForItemAtIndexPath:withLayoutAttributes:applyAttributes:isFocused:notify:] + 314
    9   UIKitCore                           0x0000000107831198 -[UICollectionView
    _createPreparedCellForItemAtIndexPath:withLayoutAttributes:applyAttributes:] + 31
    10  UIKitCore                           0x000000010783684f -[UICollectionView _updateVisibleCellsNow:] +
    6164
    11  UIKitCore                           0x000000010783c076 -[UICollectionView layoutSubviews] + 364
    12  UIKitCore                           0x00000001084c9795 -[UIView(CALayerDelegate)
    layoutSublayersOfLayer:] + 1441
    13  QuartzCore                          0x0000000109a51b19 -[CALayer layoutSublayers] + 175
    14  QuartzCore                          0x0000000109a569d3
    _ZN2CA5Layer16layout_if_neededEPNS_11TransactionE + 395
    15  QuartzCore                          0x00000001099cf7ca
    _ZN2CA7Context18commit_transactionEPNS_11TransactionE + 342
    16  QuartzCore                          0x0000000109a0697e _ZN2CA11Transaction6commitEv + 576
    17  UIKitCore                           0x0000000107fd9701 _UIApplicationFlushRunLoopCATransactionIfTooLate
    + 165
    18  UIKitCore                           0x00000001080d3569 __handleEventQueueInternal + 6874
    19  CoreFoundation                      0x00000001055b9721
    __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__ + 17
    20  CoreFoundation                      0x00000001055b8f93 __CFRunLoopDoSources0 + 243
    21  CoreFoundation                      0x00000001055b363f __CFRunLoopRun + 1263
    22  CoreFoundation                      0x00000001055b2e11 CFRunLoopRunSpecific + 625
    23  GraphicsServices                    0x000000010d8551dd GSEventRunModal + 62
    24  UIKitCore                           0x0000000107fdf81d UIApplicationMain + 140
    25  LetsEat                             0x0000000103336a927 main + 71
    26  libdyld.dylib                       0x0000000106aed575 start + 1
)
libc++abi.dylib: terminating with uncaught exception of type NSException
(lldb)
```

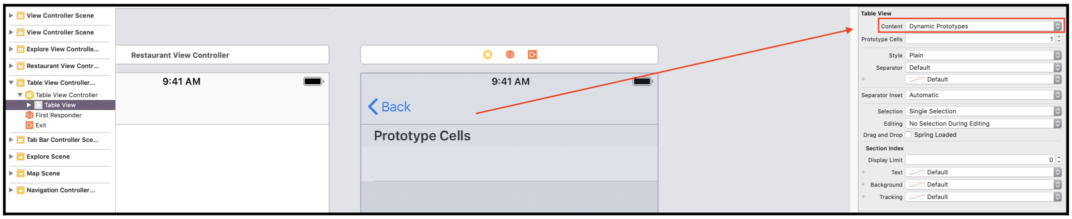
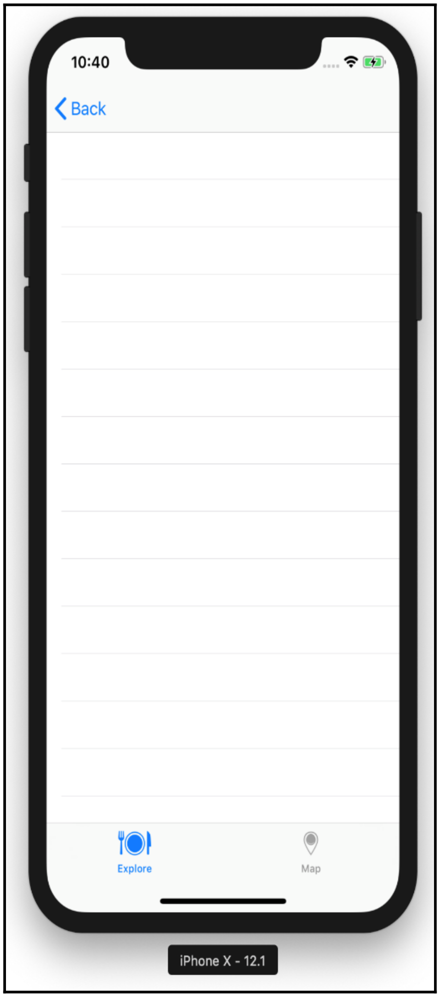
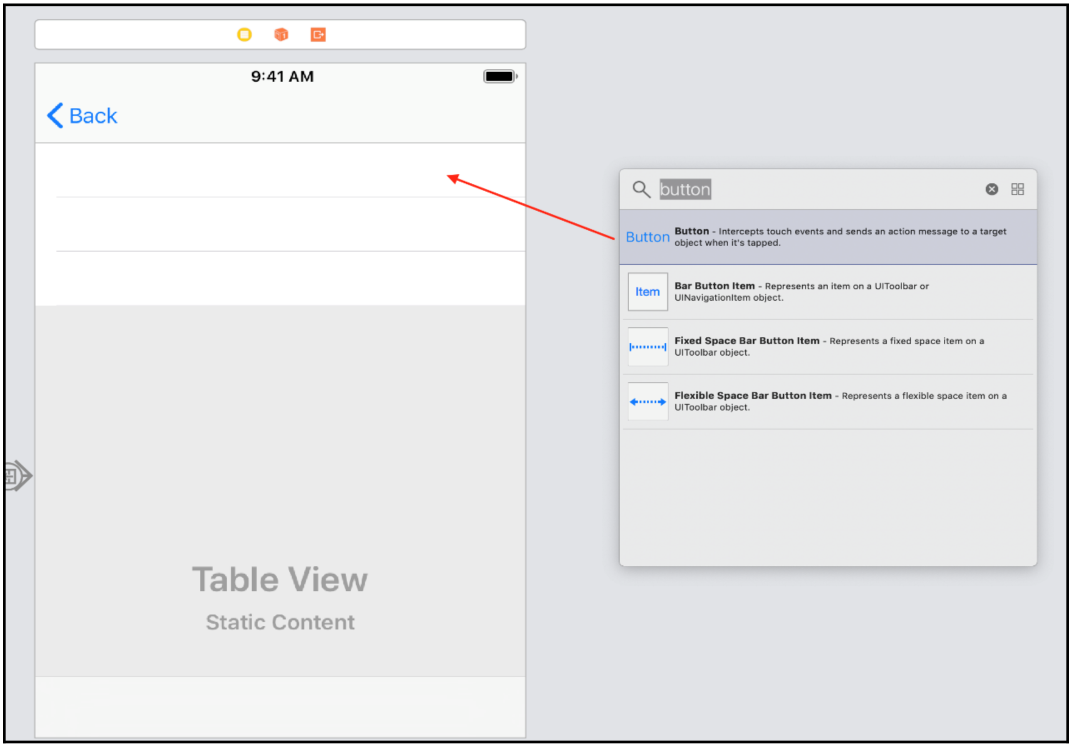


Table View
Content Dynamic Prototypes
Prototype Cells 1
Style Plain
Separator Default
Default
Separator Inset Automatic
Selection Single Selection
Editing No Selection During Editing
Drag and Drop □ Spring Loaded
Section Index
Display Limit 0
Text Default
Background Default
Tracking Default



10:40
Back
Explore
Map
iPhone X - 12.1



Button - Intercepts touch events and sends an action message to a target object when it's tapped.

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.

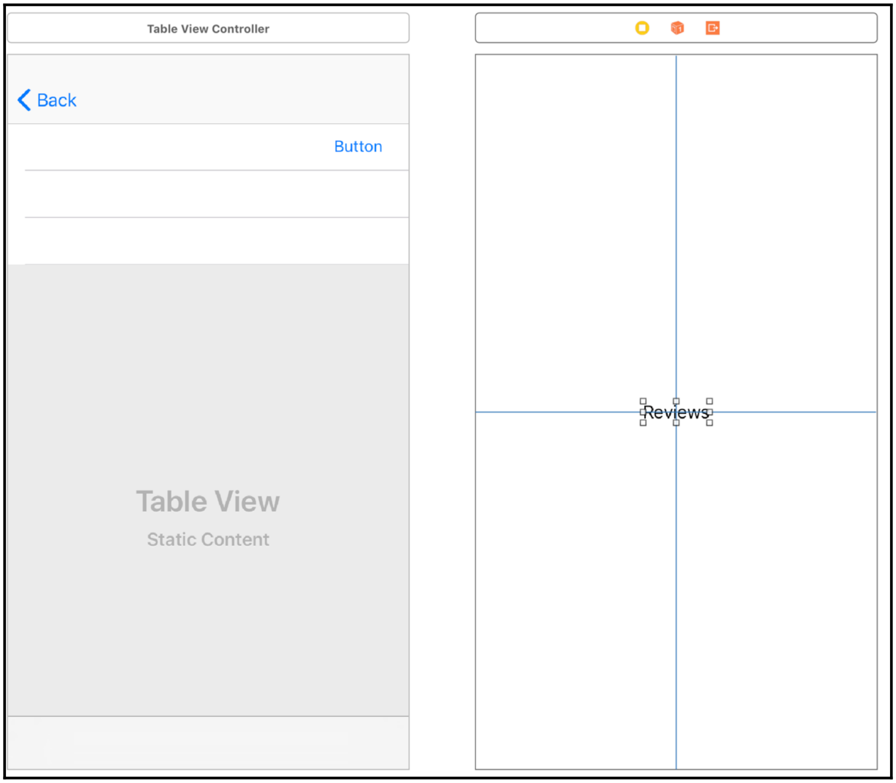
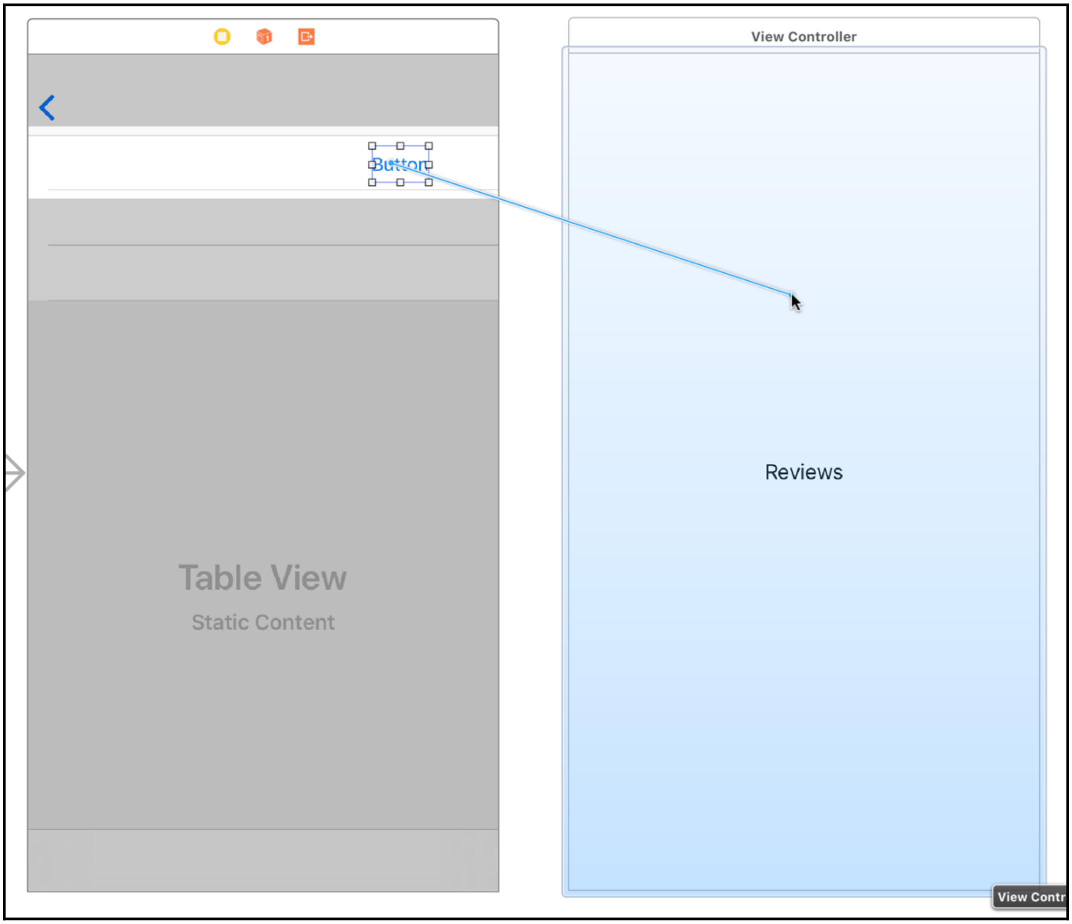
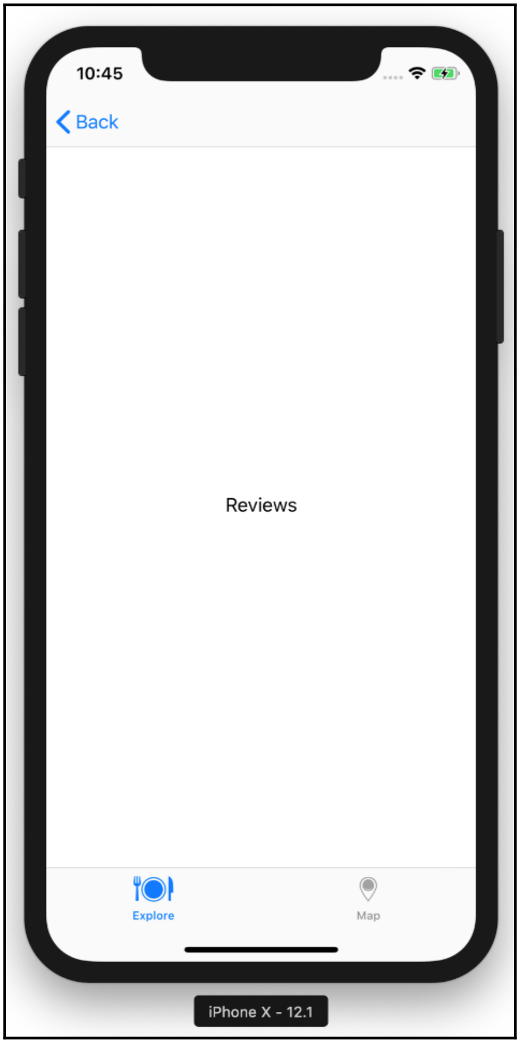
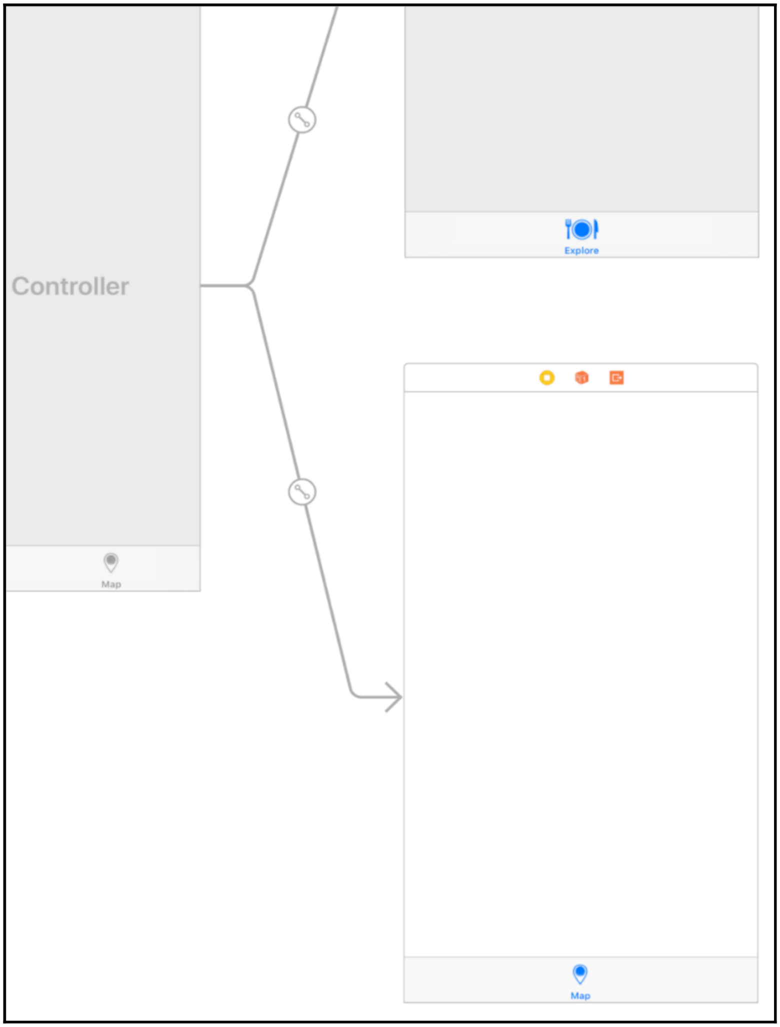
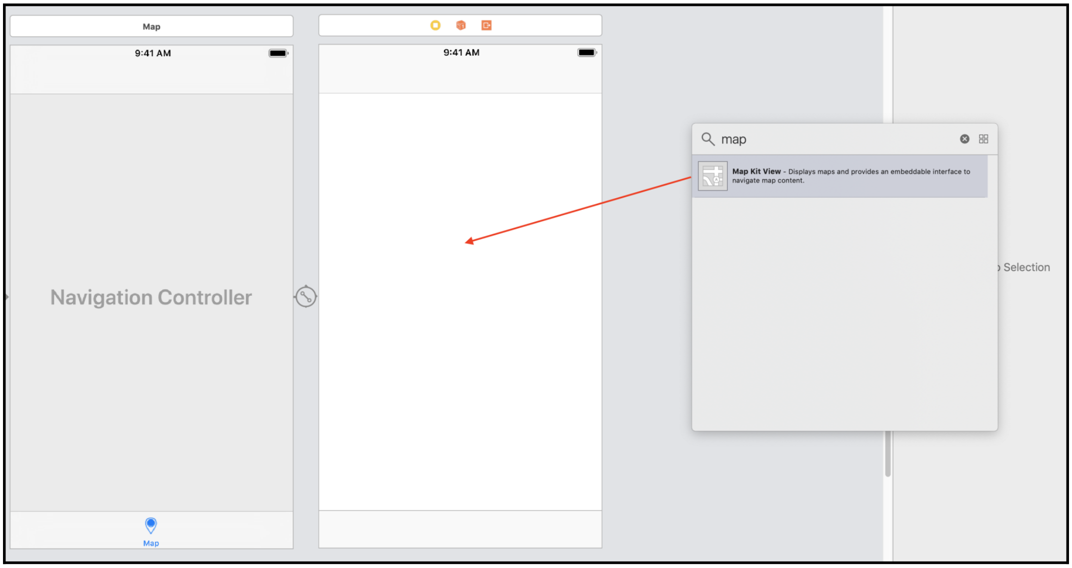


Table View Controller
Back
Button
Table View
Static Content
Reviews

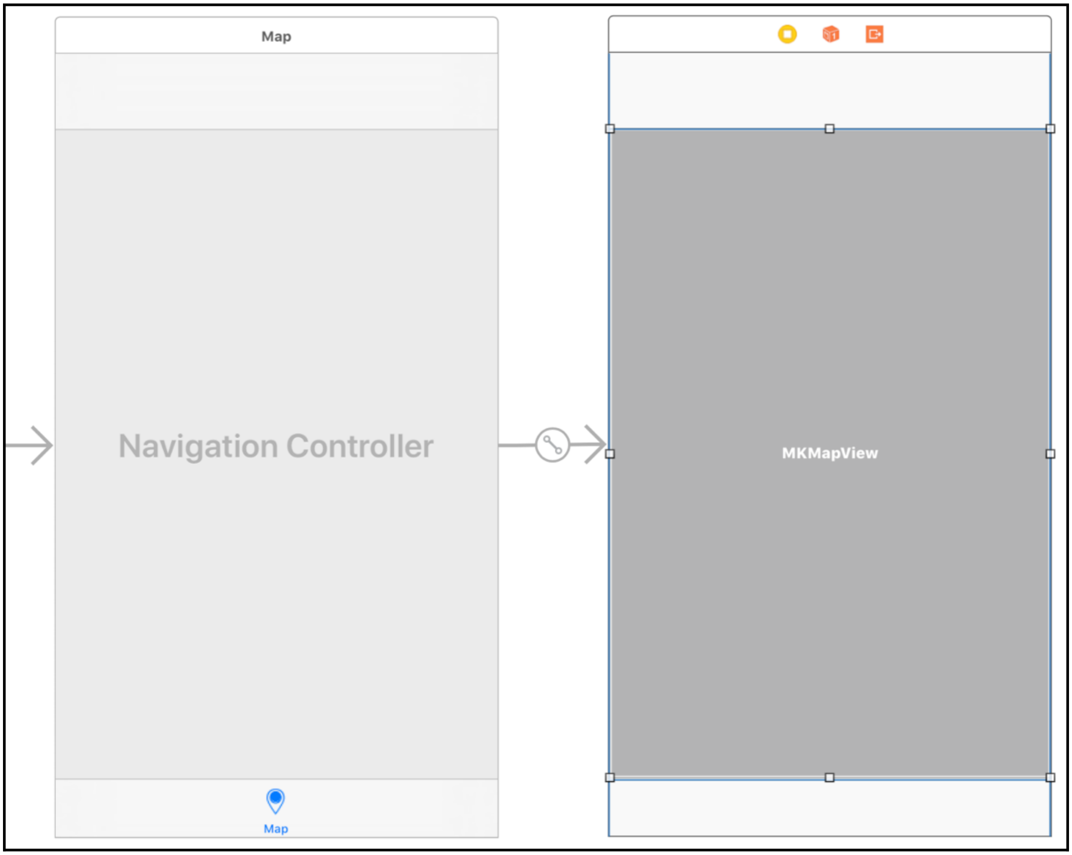

View Controller
Button
Table View
Static Content
Reviews
View Contr

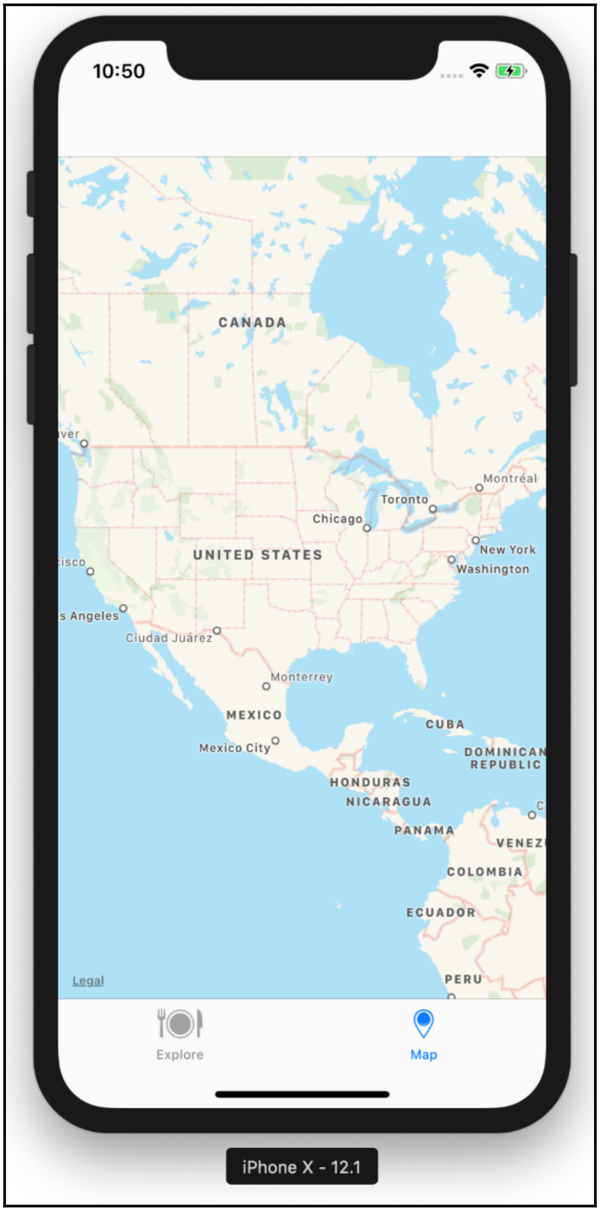

10:45
Back
Reviews
Explore
Map
iPhone X - 12.1

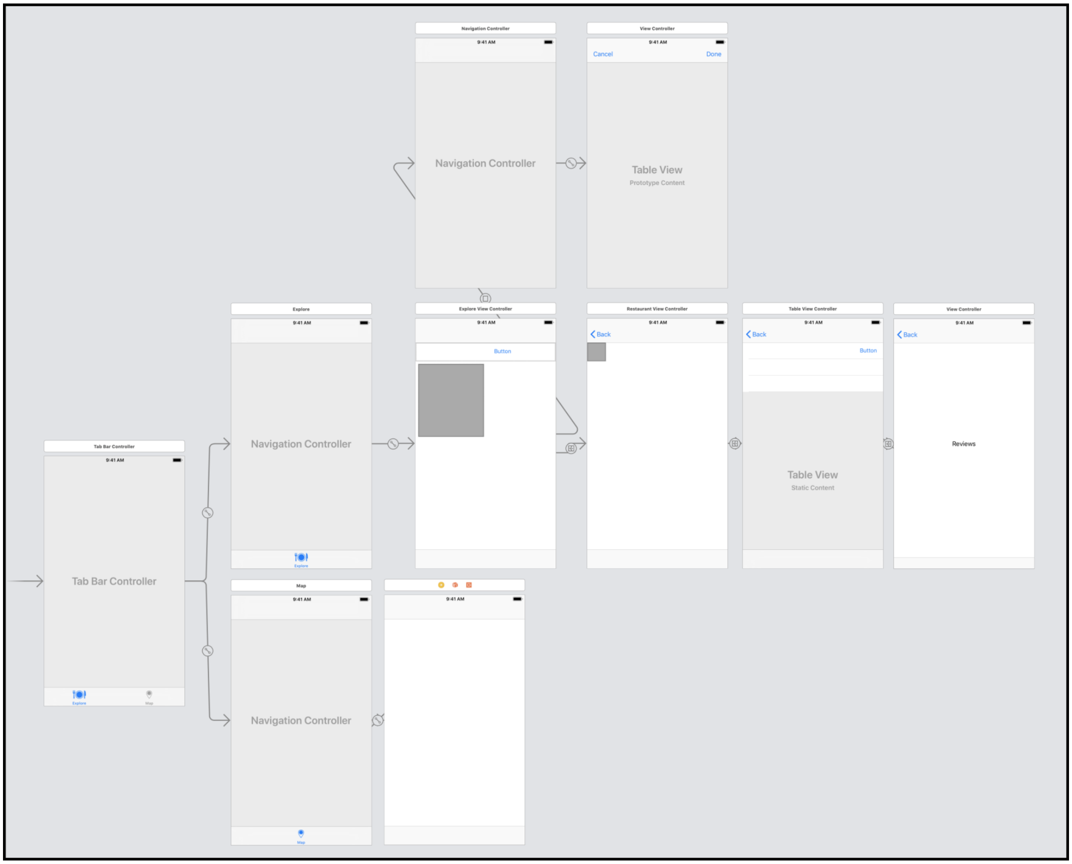

Controller
Map
Explore
Map



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

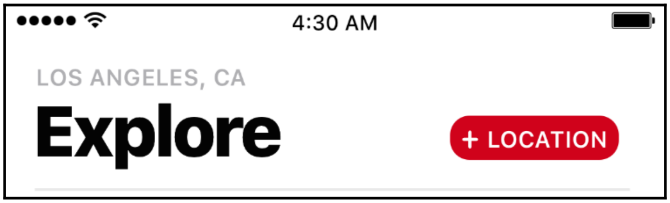

Map
Navigation Controller
Map
MKMapView

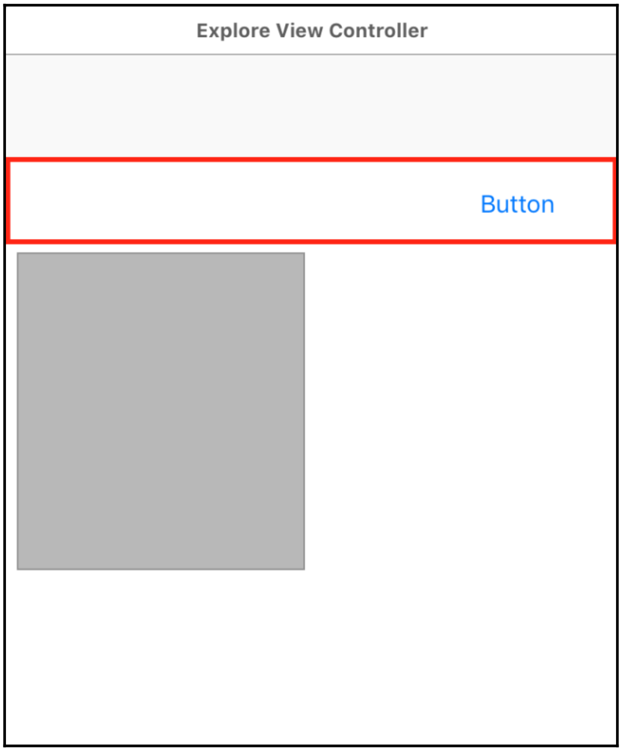

10:50
CANADA
UNITED STATES
Montréal
Toronto
Chicago
New York
Washington
ver
cisco
s Angeles
Ciudad Juárez
Monterrey
MEXICO
Mexico City
CUBA
DOMINICAN REPUBLIC
HONDURAS
NICARAGUA
PANAMA
VENEZ
COLOMBIA
ECUADOR
PERU
Legal
Explore
Map
iPhone X - 12.1

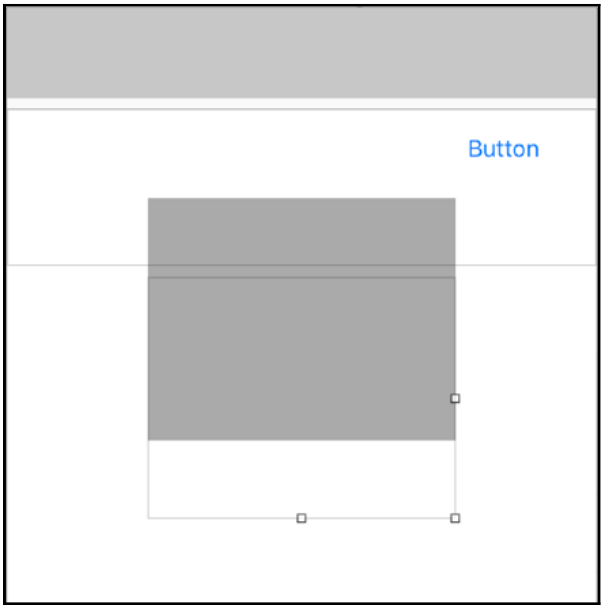


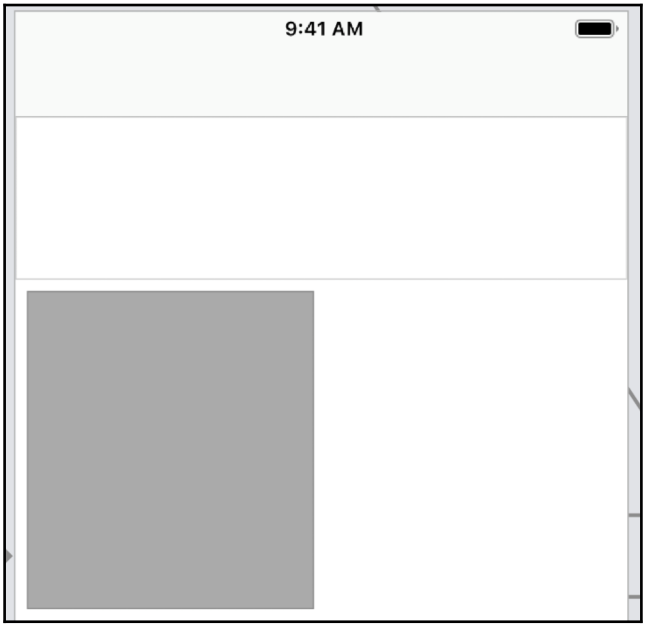
Navigation Controller
9:41 AM
Navigation Controller
View Controller
9:41 AM
Cancel
Done
Table View
Prototype Content
Explore
9:41 AM
Navigation Controller
Explore
Explore View Controller
9:41 AM
Button
Restaurant View Controller
9:41 AM
Back
Table View Controller
9:41 AM
Back
Button
Table View
Static Content
View Controller
9:41 AM
Back
Reviews
Tab Bar Controller
9:41 AM
Tab Bar Controller
Explore
Map
Map
9:41 AM
Navigation Controller
Map
9:41 AM

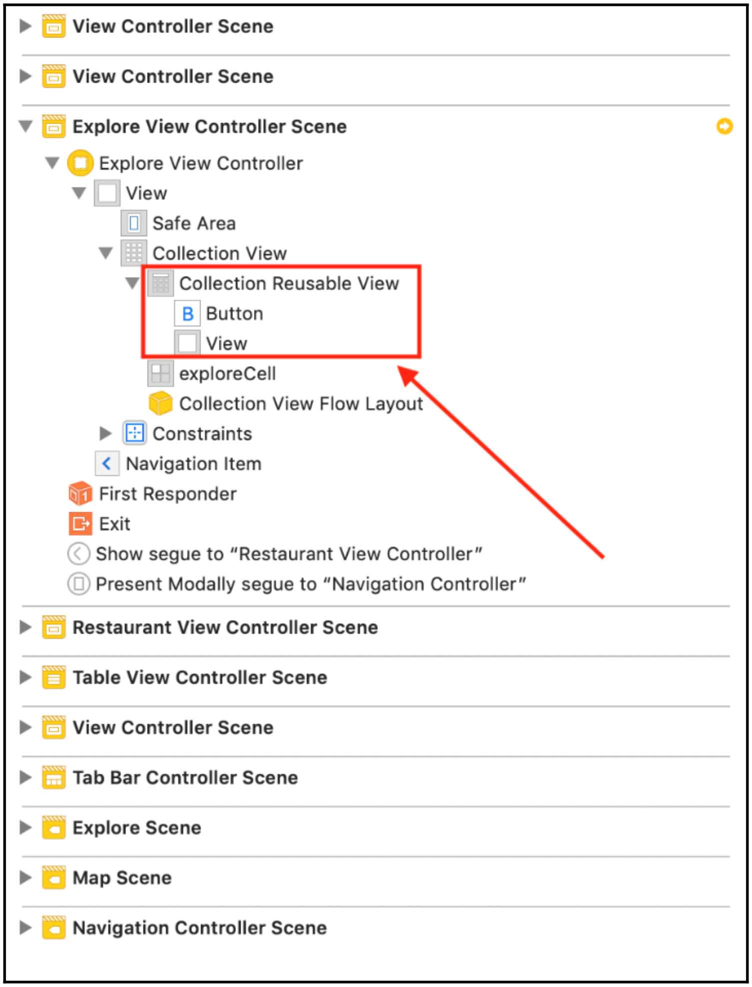
Chapter 10: Designing Cells



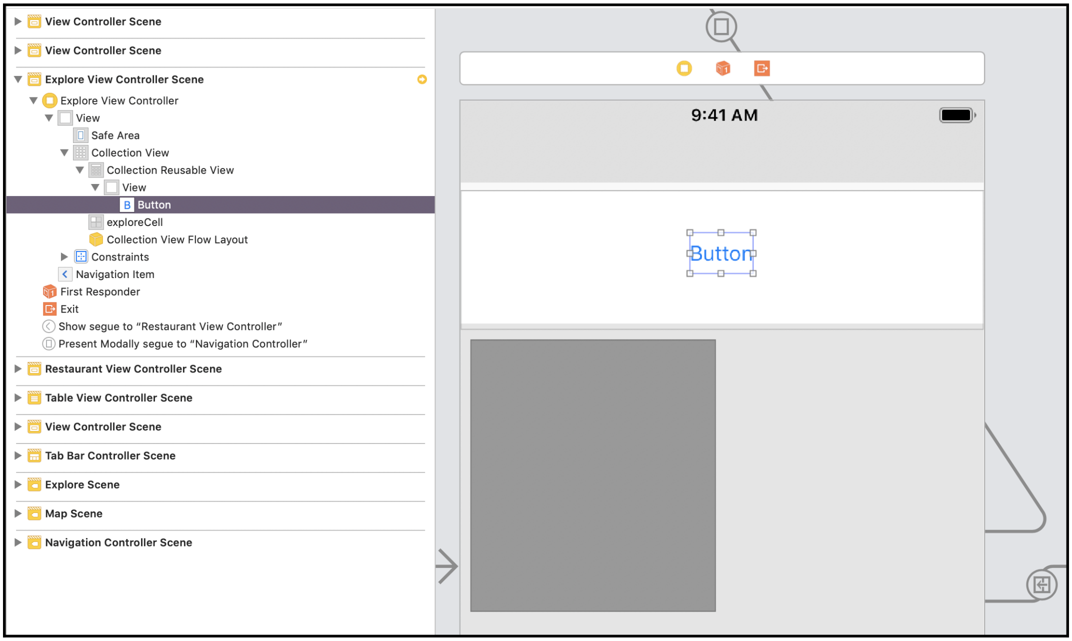


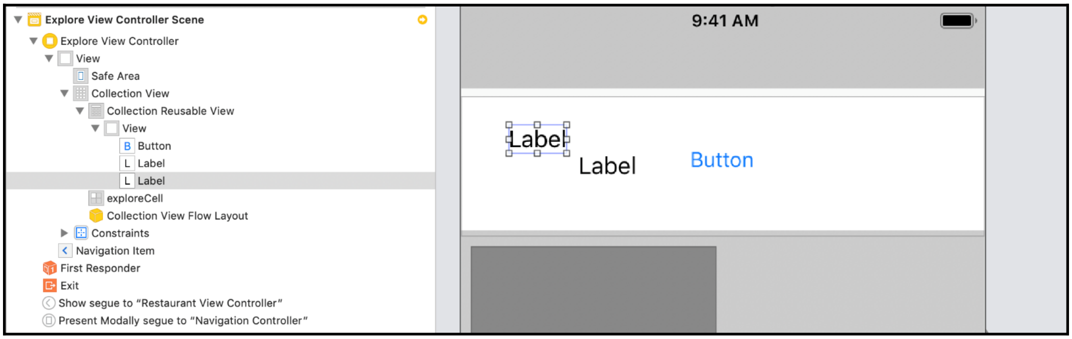

Button

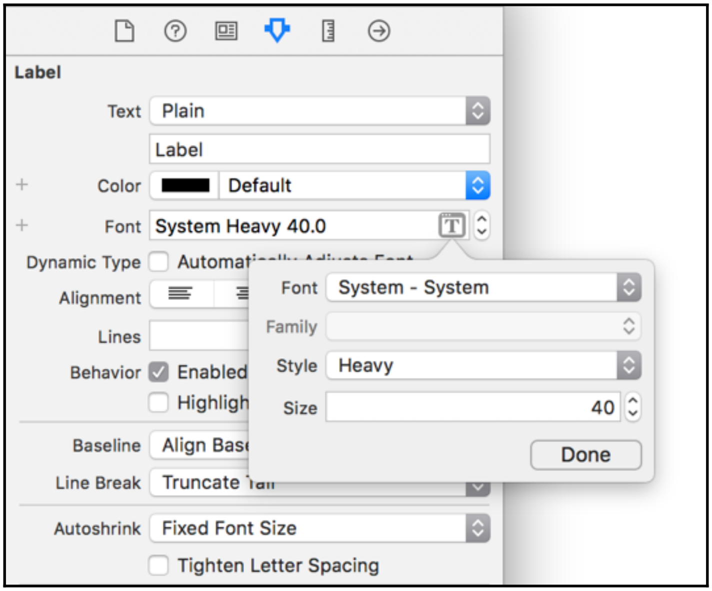

9:41 AM



- View Controller Scene
- View Controller Scene
- Explore View Controller Scene
  - Explore View Controller
    - View
      - Safe Area
      - Collection View
        - Collection Reusable View
          - Button
          - View
        - exploreCell
        - Collection View Flow Layout
      - Constraints
    - Navigation Item
  - First Responder
  - Exit
  - Show segue to "Restaurant View Controller"
  - Present Modally segue to "Navigation Controller"
- Restaurant View Controller Scene
- Table View Controller Scene
- View Controller Scene
- Tab Bar Controller Scene
- Explore Scene
- Map Scene
- Navigation Controller Scene


View Controller Scene
View Controller Scene
Explore View Controller Scene
Explore View Controller
View
Safe Area
Collection View
Collection Reusable View
View
Button
exploreCell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Show segue to "Restaurant View Controller"
Present Modally segue to "Navigation Controller"
Restaurant View Controller Scene
Table View Controller Scene
View Controller Scene
Tab Bar Controller Scene
Explore Scene
Map Scene
Navigation Controller Scene
9:41 AM
Button


Explore View Controller Scene
Explore View Controller
View
Safe Area
Collection View
Collection Reusable View
View
Button
Label
Label
exploreCell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Show segue to "Restaurant View Controller"
Present Modally segue to "Navigation Controller"
9:41 AM
Label
Label
Button



Label

Text: Plain

Label

Color: Default

Font: System Heavy 40.0

Dynamic Type: □ Automatically Adjusts Font

Alignment

Lines

Behavior: ☑ Enabled

□ Highligh

Baseline: Align Base

Line Break: Truncate Tail

Autoshrink: Fixed Font Size

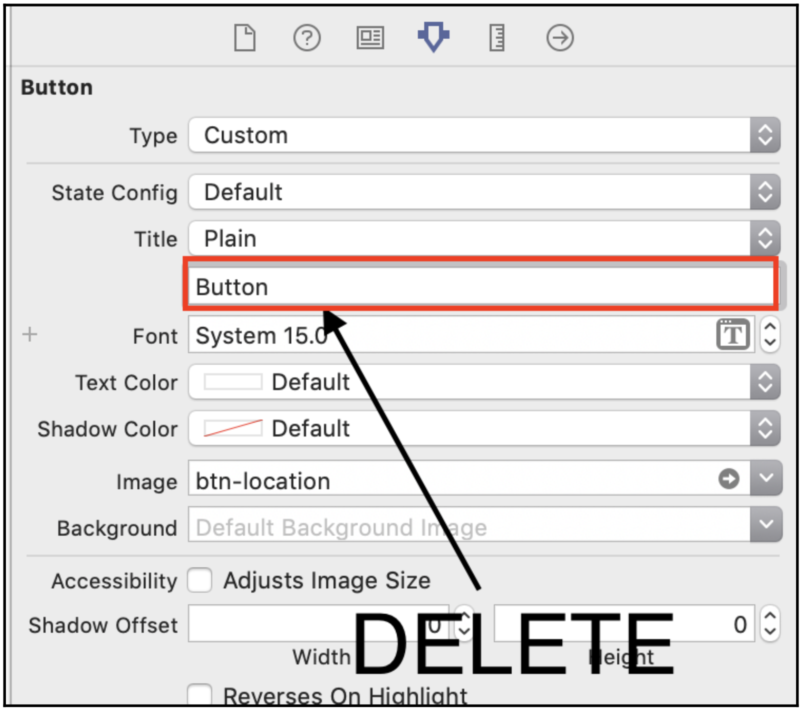
□ Tighten Letter Spacing

Font: System - System

Family

Style: Heavy

Size: 40



Button

Type: Custom

State Config: Default

Title: Plain

Button

Font: System 15.0

Text Color: Default

Shadow Color: Default

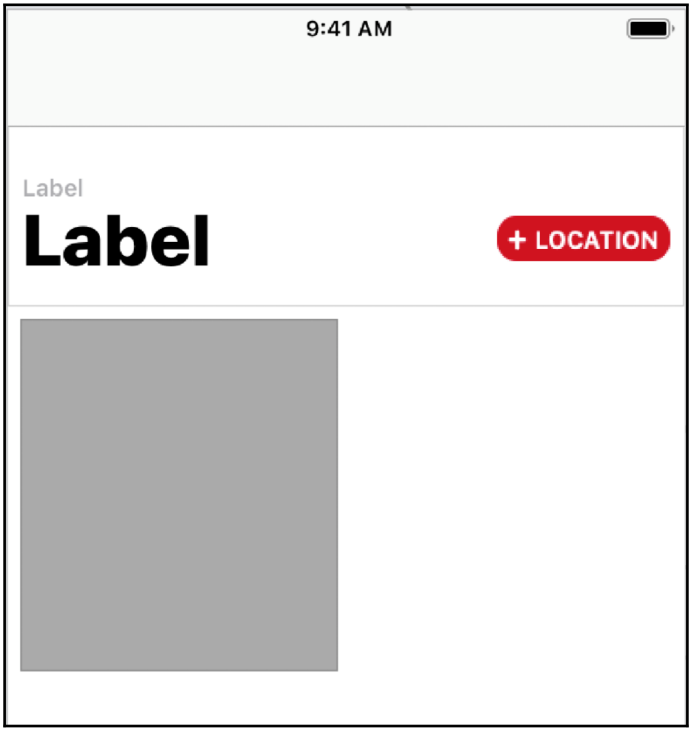
Image: btn-location

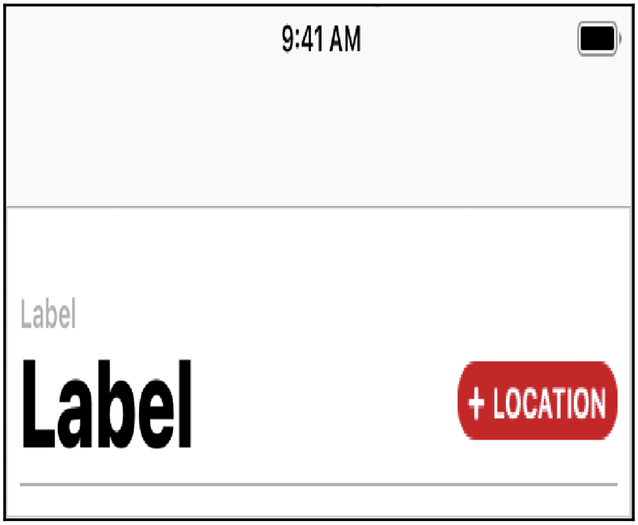
Background: Default Background Image

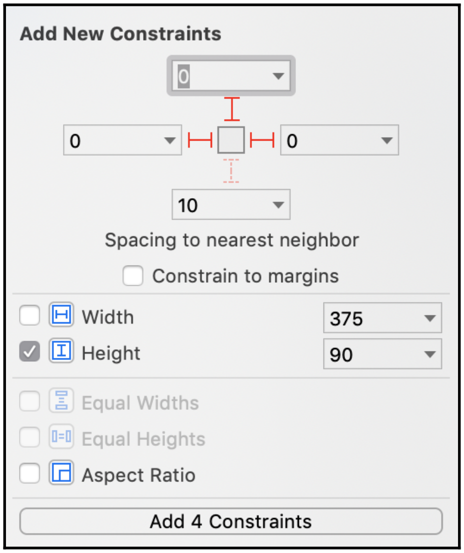
Accessibility: □ Adjusts Image Size

Shadow Offset: 0 Width, 0 Height

□ Reverses On Highlight


9:41 AM
Label
Label
+ LOCATION


9:41 AM
Label
Label
+ LOCATION



Add New Constraints

0

0 0

10

Spacing to nearest neighbor

□ Constrain to margins

□ Width 375

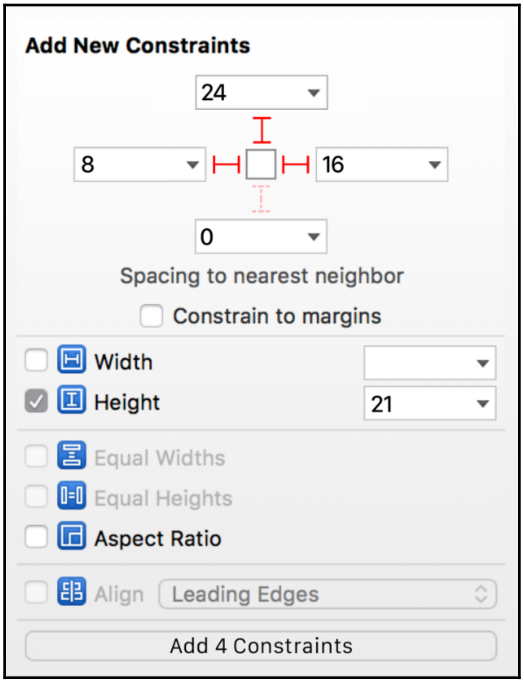
☑ Height 90

□ Equal Widths

□ Equal Heights

□ Aspect Ratio

Add 4 Constraints



Add New Constraints

24

8 16

0

Spacing to nearest neighbor

□ Constrain to margins

□ Width

☑ Height 21

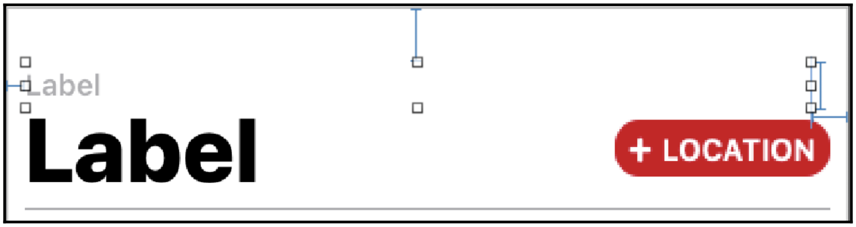
□ Equal Widths

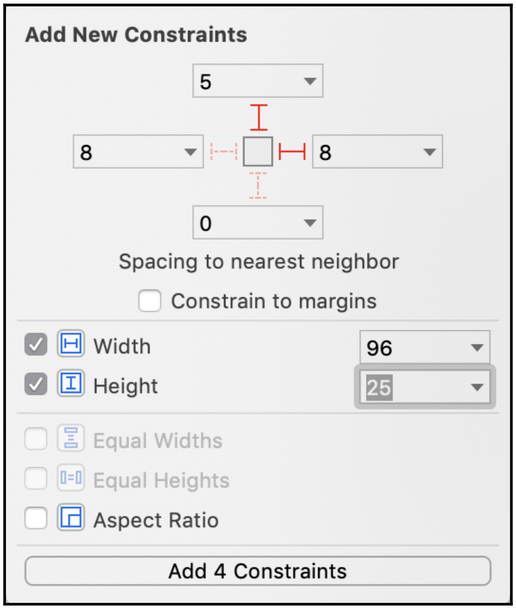
□ Equal Heights

□ Aspect Ratio

□ Align Leading Edges

Add 4 Constraints




Add New Constraints

5

8 | 8

0

Spacing to nearest neighbor

□ Constrain to margins

☑ Width 96

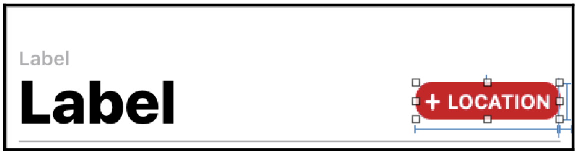
☑ Height 25

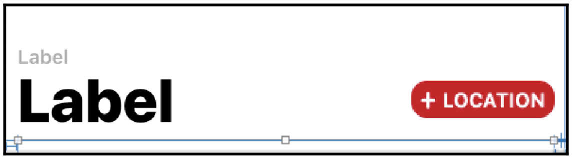
□ Equal Widths

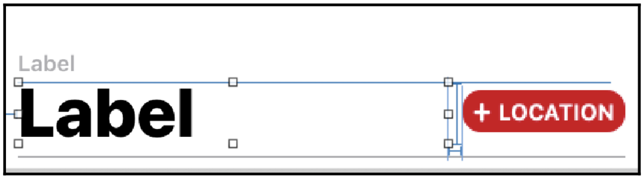
□ Equal Heights

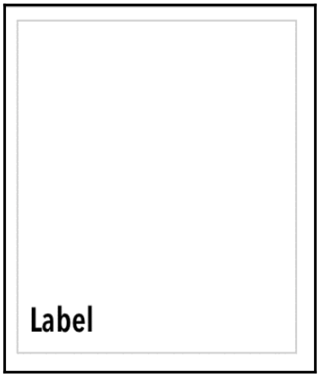
□ Aspect Ratio

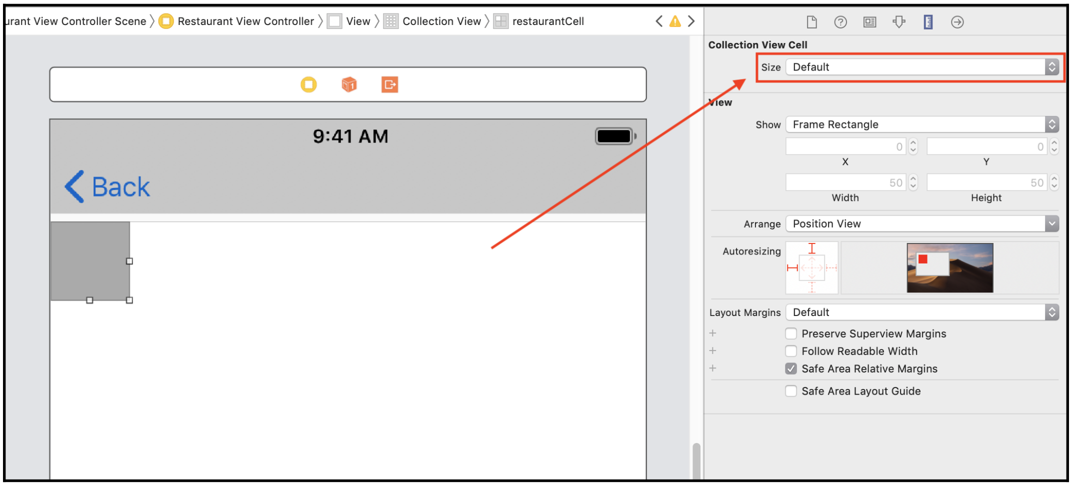
Add 4 Constraints










urant View Controller Scene 〉 Restaurant View Controller 〉 View 〉 Collection View 〉 restaurantCell
9:41 AM
Back

Collection View Cell

Size: Default

View

Show: Frame Rectangle

X: 0 Y: 0

Width: 50 Height: 50

Arrange: Position View

Autoresizing

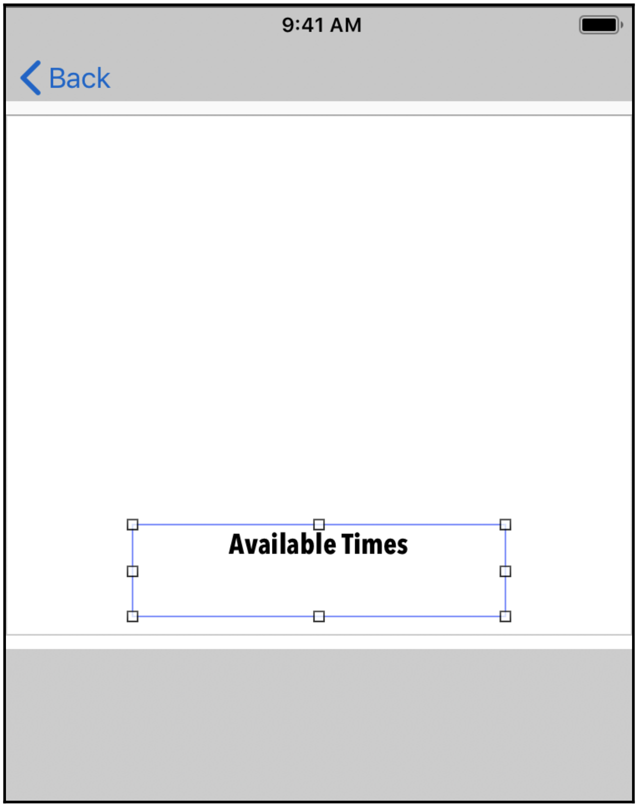
Layout Margins: Default

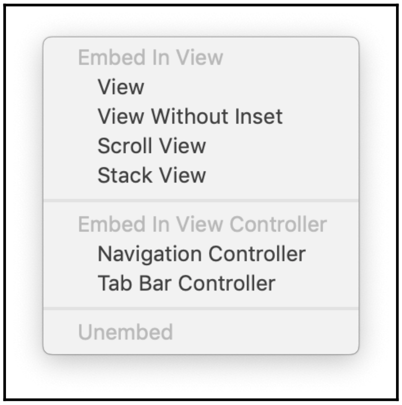
☐ Preserve Superview Margins

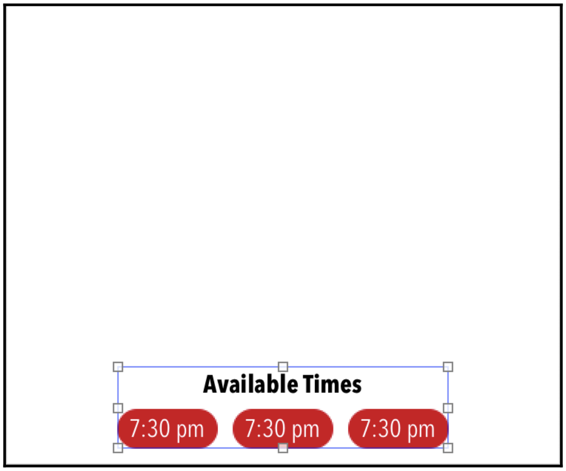
☐ Follow Readable Width

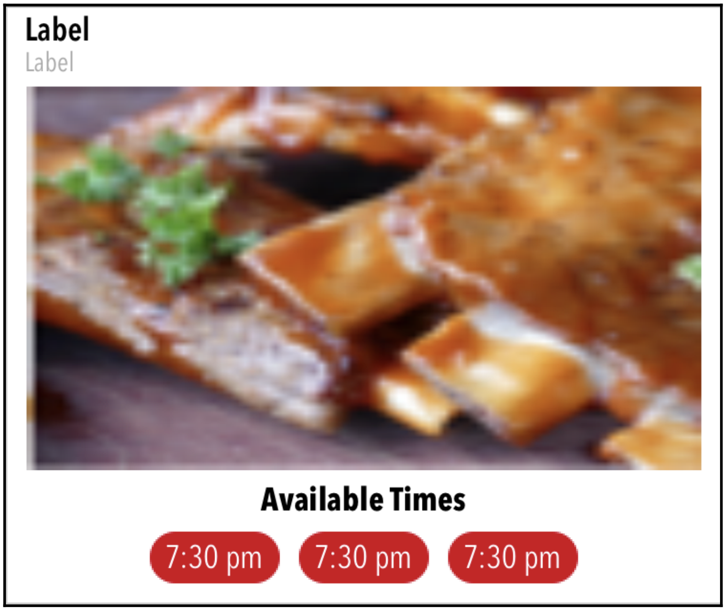
☑ Safe Area Relative Margins

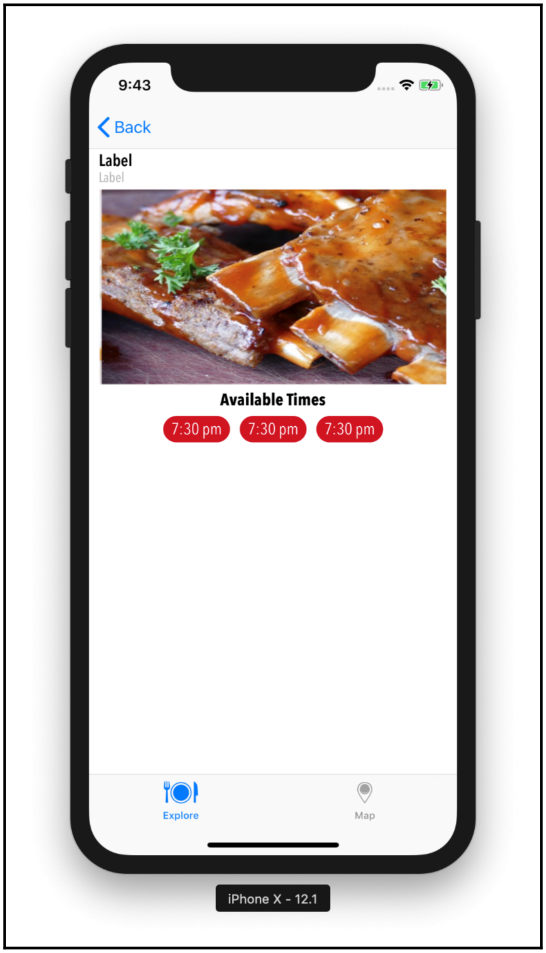
☐ Safe Area Layout Guide

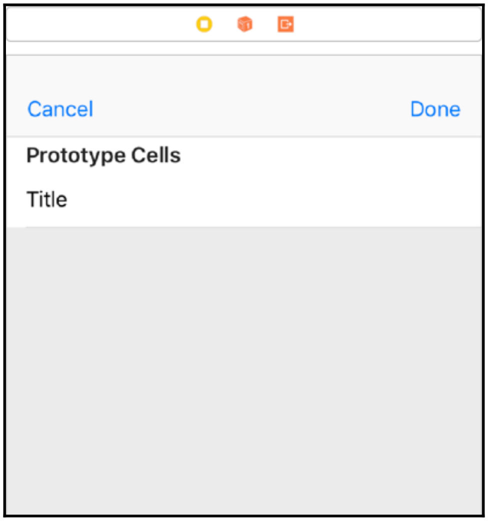

9:41 AM
Back
Available Times


Embed In View
View
View Without Inset
Scroll View
Stack View
Embed In View Controller
Navigation Controller
Tab Bar Controller
Unembed


Available Times
7:30 pm
7:30 pm
7:30 pm

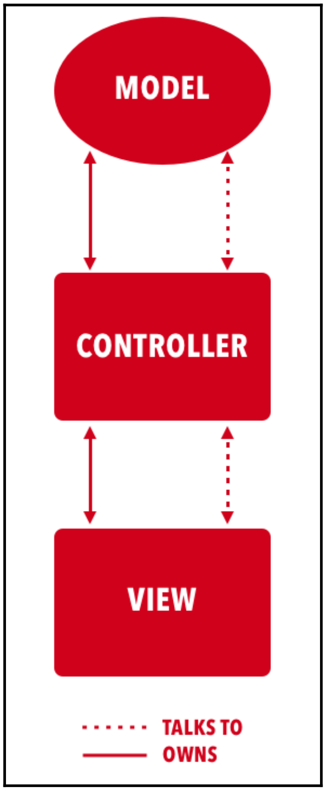

Label
Label
Available Times
7:30 pm
7:30 pm
7:30 pm

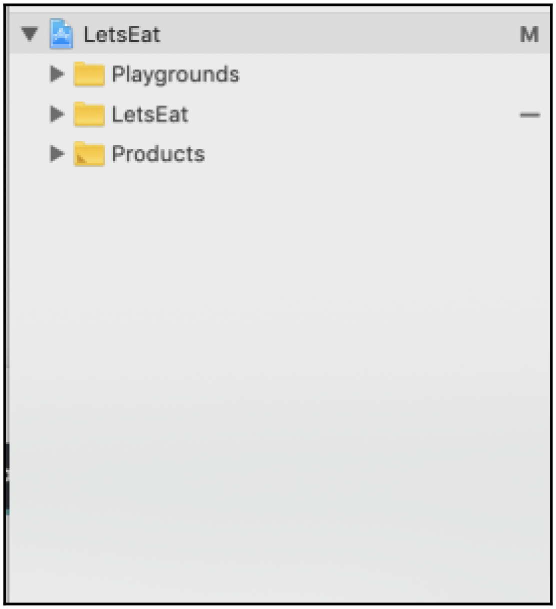

9:43
Back
Label
Label
Available Times
7:30 pm
7:30 pm
7:30 pm
Explore
Map
iPhone X - 12.1

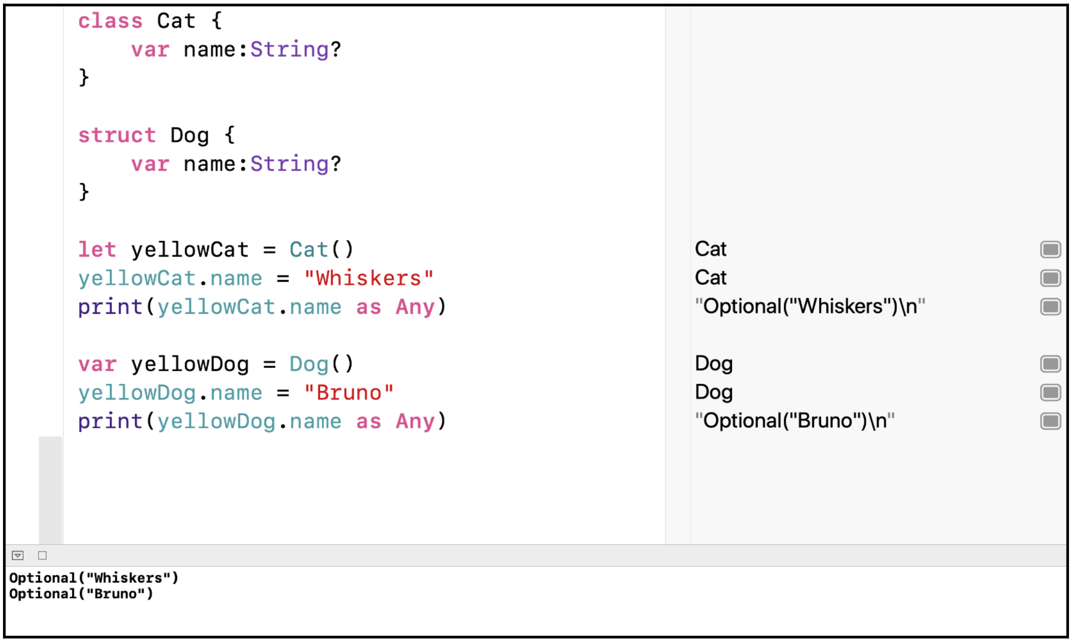


Cancel
Done
Prototype Cells
Title

Chapter 11: Getting Started with the Grid

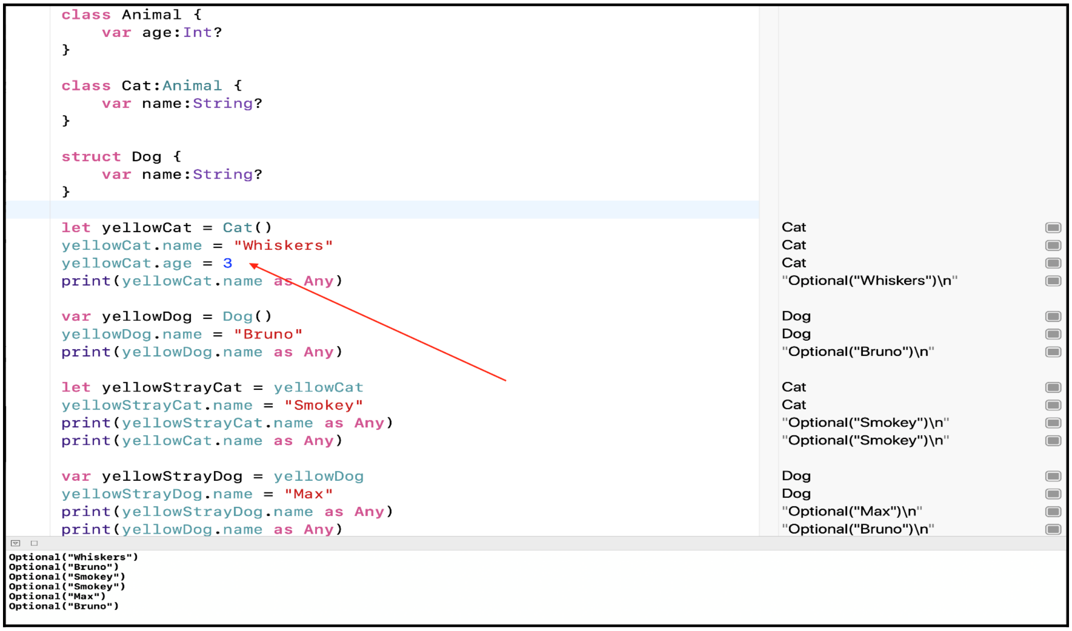






```
class Cat {
    var name:String?
}

struct Dog {
    var name:String?
}

let yellowCat = Cat()                        Cat
yellowCat.name = "Whiskers"                  Cat
print(yellowCat.name as Any)                 "Optional("Whiskers")\n"

var yellowDog = Dog()                        Dog
yellowDog.name = "Bruno"                     Dog
print(yellowDog.name as Any)                 "Optional("Bruno")\n"
```

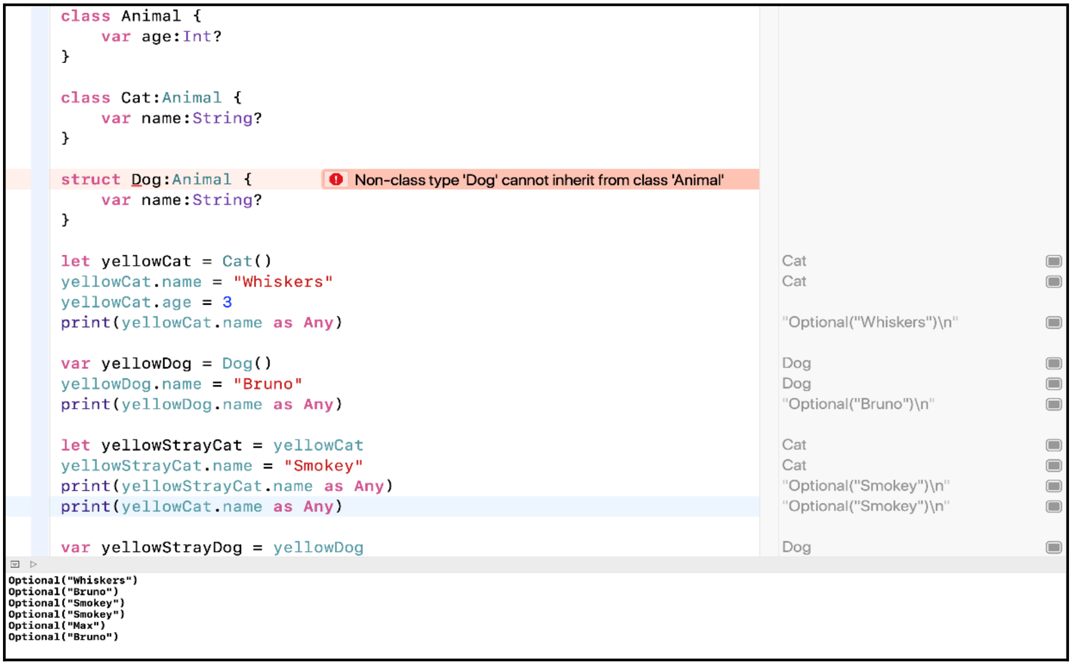
```
Optional("Whiskers")
Optional("Bruno")
```



```
class Animal {
    var age:Int?
}

class Cat:Animal {
    var name:String?
}

struct Dog {
    var name:String?
}

let yellowCat = Cat()
yellowCat.name = "Whiskers"
yellowCat.age = 3
print(yellowCat.name as Any)

var yellowDog = Dog()
yellowDog.name = "Bruno"
print(yellowDog.name as Any)

let yellowStrayCat = yellowCat
yellowStrayCat.name = "Smokey"
print(yellowStrayCat.name as Any)
print(yellowCat.name as Any)

var yellowStrayDog = yellowDog
yellowStrayDog.name = "Max"
print(yellowStrayDog.name as Any)
print(yellowDog.name as Any)
```

```
Cat
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")
```

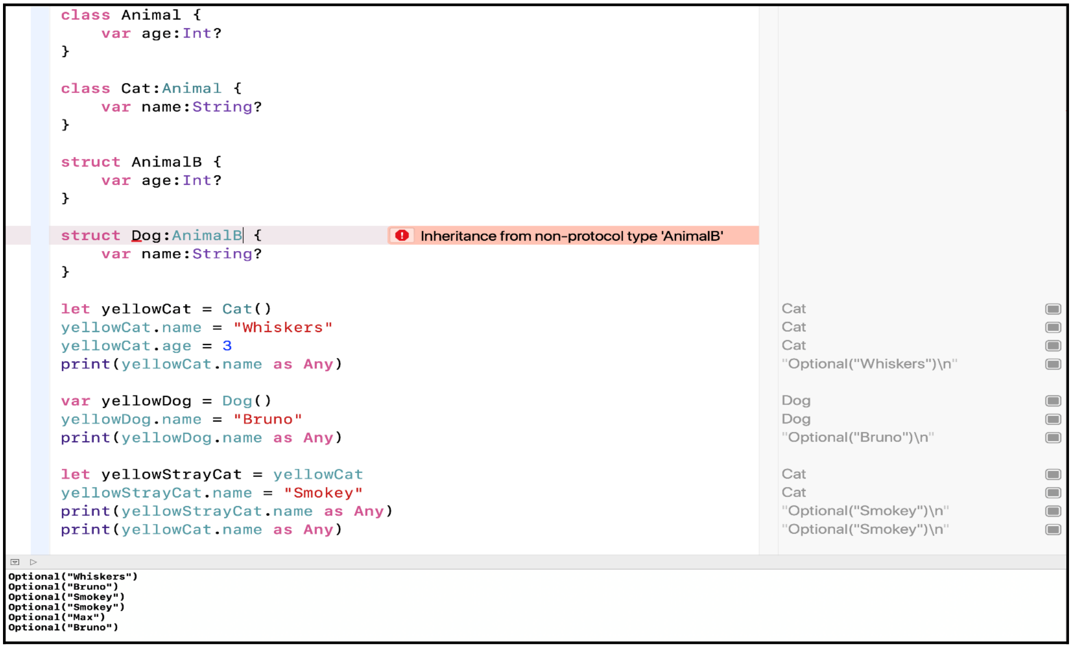


```
class Animal {
    var age:Int?
}

class Cat:Animal {
    var name:String?
}

struct Dog:Animal {
    var name:String?
}

let yellowCat = Cat()
yellowCat.name = "Whiskers"
yellowCat.age = 3
print(yellowCat.name as Any)

var yellowDog = Dog()
yellowDog.name = "Bruno"
print(yellowDog.name as Any)

let yellowStrayCat = yellowCat
yellowStrayCat.name = "Smokey"
print(yellowStrayCat.name as Any)
print(yellowCat.name as Any)

var yellowStrayDog = yellowDog
```

Non-class type 'Dog' cannot inherit from class 'Animal'

```
Cat
Cat

"Optional("Whiskers")\n"

Dog
Dog
"Optional("Bruno")\n"

Cat
Cat
"Optional("Smokey")\n"
"Optional("Smokey")\n"

Dog
```

```
Optional("Whiskers")
Optional("Bruno")
Optional("Smokey")
Optional("Smokey")
Optional("Max")
Optional("Bruno")
```

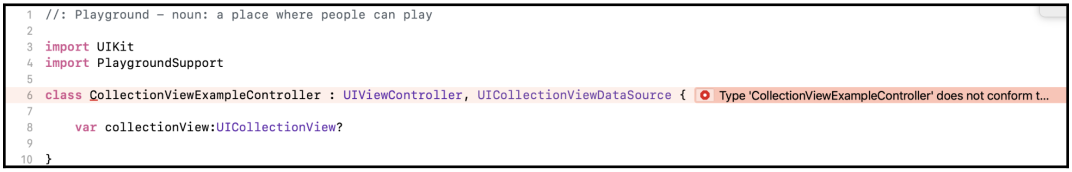


```
class Animal {
    var age:Int?
}

class Cat:Animal {
    var name:String?
}

struct AnimalB {
    var age:Int?
}

struct Dog:AnimalB {
    var name:String?
}

let yellowCat = Cat()
yellowCat.name = "Whiskers"
yellowCat.age = 3
print(yellowCat.name as Any)

var yellowDog = Dog()
yellowDog.name = "Bruno"
print(yellowDog.name as Any)

let yellowStrayCat = yellowCat
yellowStrayCat.name = "Smokey"
print(yellowStrayCat.name as Any)
print(yellowCat.name as Any)
```

Inheritance from non-protocol type 'AnimalB'

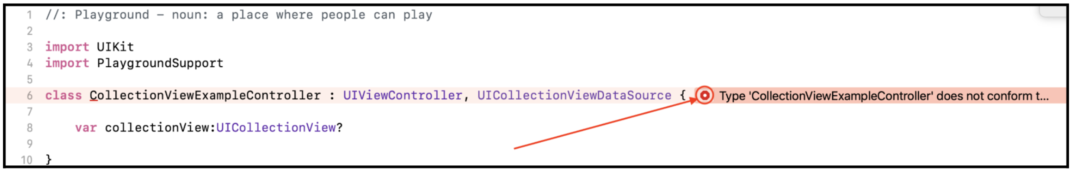
```
Cat
Cat
Cat
"Optional("Whiskers")\n"

Dog
Dog
"Optional("Bruno")\n"

Cat
Cat
"Optional("Smokey")\n"
"Optional("Smokey")\n"
```

```
Optional("Whiskers")
Optional("Bruno")
Optional("Smokey")
Optional("Smokey")
Optional("Max")
Optional("Bruno")
```



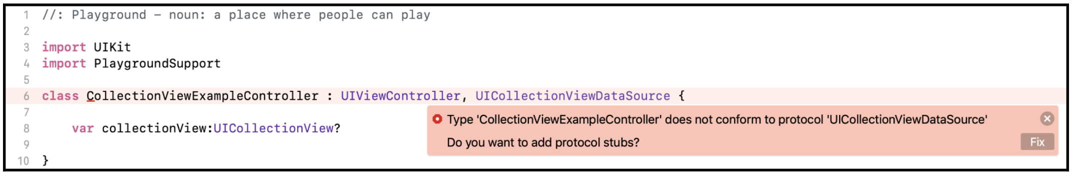
```
//: Playground - noun: a place where people can play

import UIKit
import PlaygroundSupport

class CollectionViewExampleController : UIViewController, UICollectionViewDataSource {

    var collectionView:UICollectionView?

}
```

Type 'CollectionViewExampleController' does not conform t...



```
//: Playground - noun: a place where people can play

import UIKit
import PlaygroundSupport

class CollectionViewExampleController : UIViewController, UICollectionViewDataSource {

    var collectionView:UICollectionView?

}
```

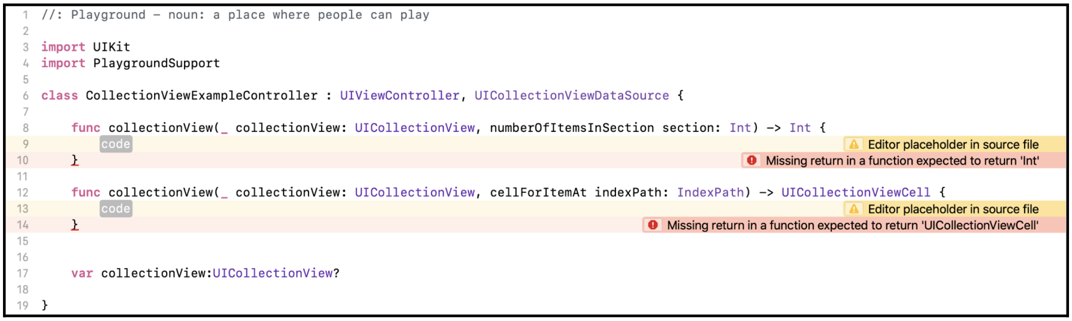
Type 'CollectionViewExampleController' does not conform t...



```
//: Playground - noun: a place where people can play

import UIKit
import PlaygroundSupport

class CollectionViewExampleController : UIViewController, UICollectionViewDataSource {

    var collectionView:UICollectionView?

}
```

Type 'CollectionViewExampleController' does not conform to protocol 'UICollectionViewDataSource'

Do you want to add protocol stubs?



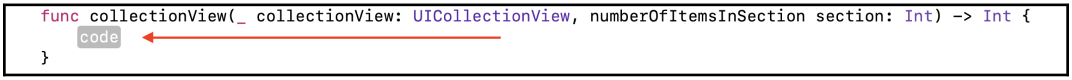
```
//: Playground - noun: a place where people can play

import UIKit
import PlaygroundSupport

class CollectionViewExampleController : UIViewController, UICollectionViewDataSource {

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

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        code
    }


    var collectionView:UICollectionView?

}
```

⚠ Editor placeholder in source file

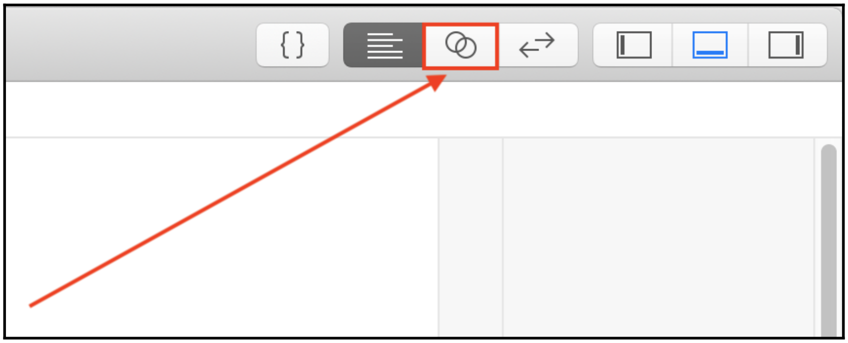
❗ Missing return in a function expected to return 'Int'

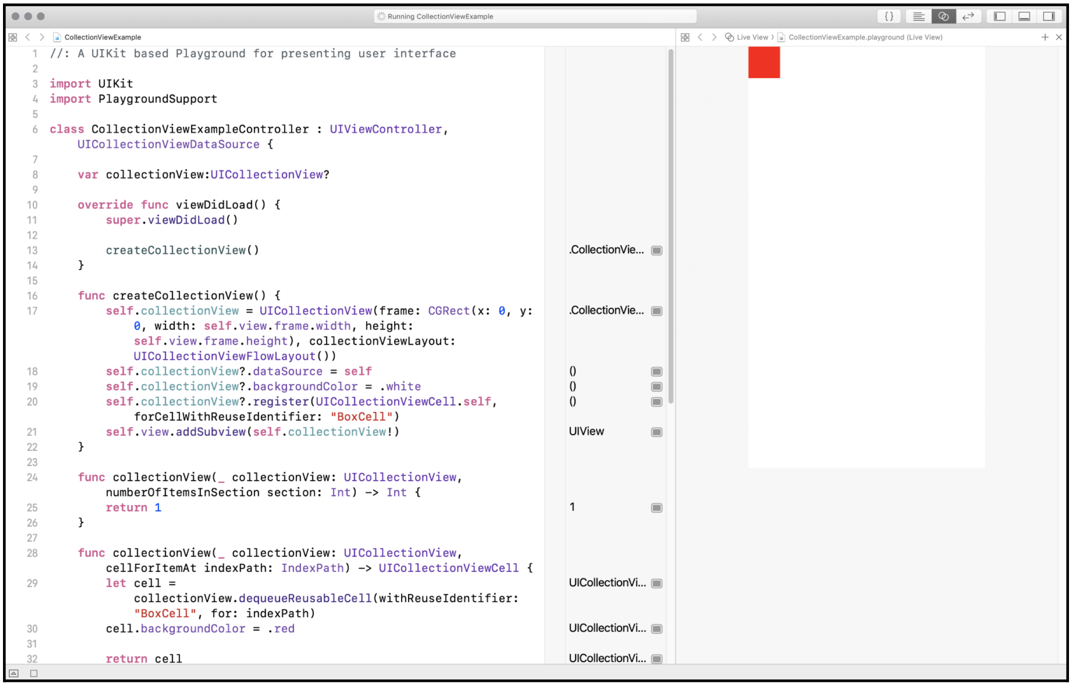
⚠ Editor placeholder in source file

❗ Missing return in a function expected to return 'UICollectionViewCell'

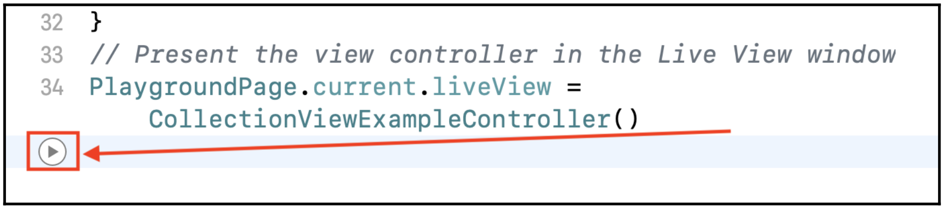


```
func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
    code
}
```

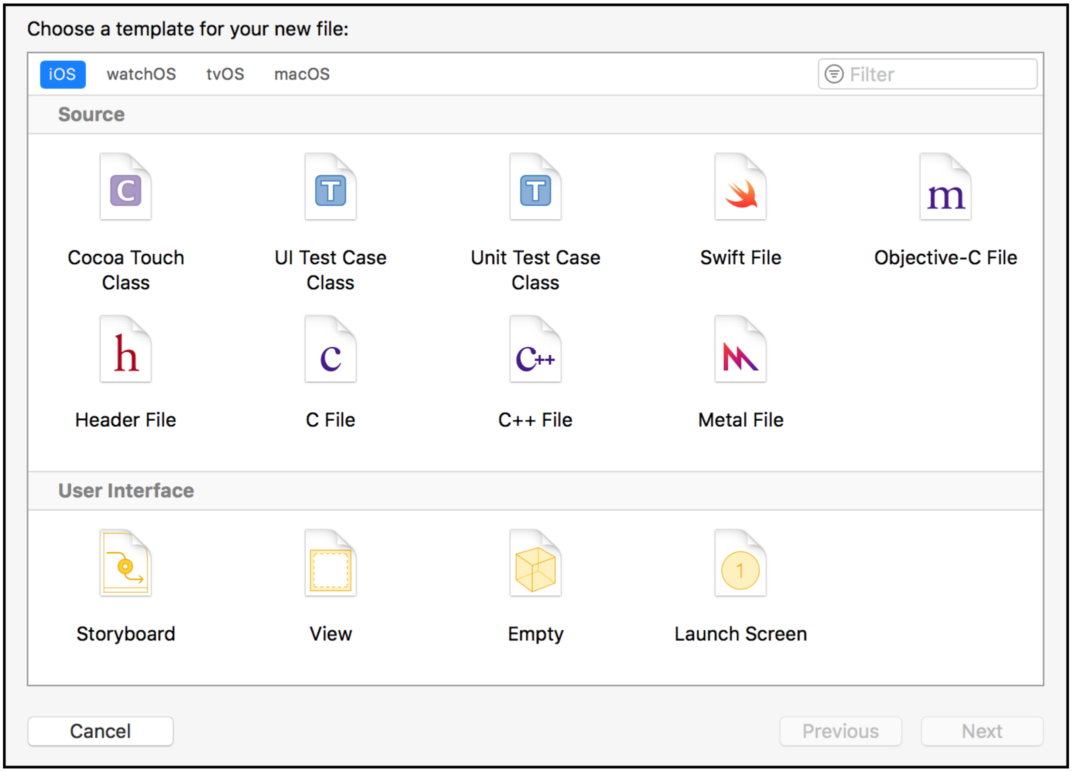




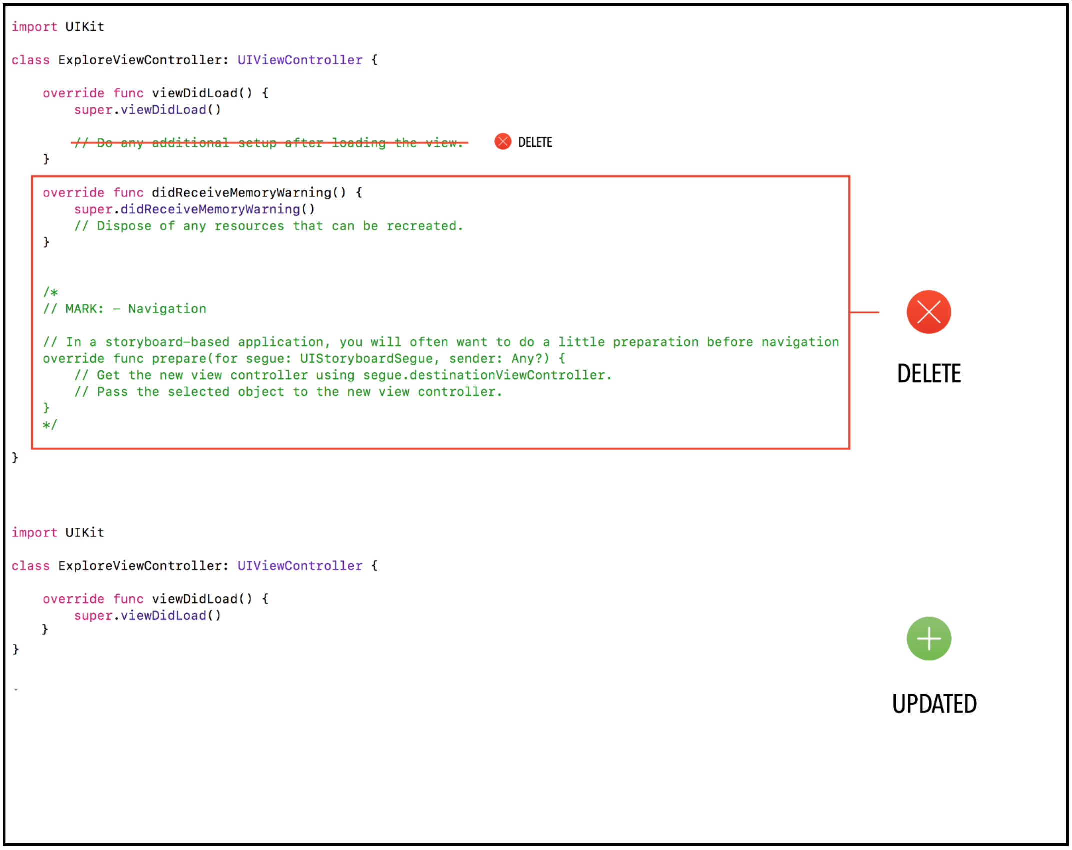
```
//: A UIKit based Playground for presenting user interface

import UIKit
import PlaygroundSupport

class CollectionViewExampleController : UIViewController,
    UICollectionViewDataSource {

    var collectionView:UICollectionView?

    override func viewDidLoad() {
        super.viewDidLoad()

        createCollectionView()
    }

    func createCollectionView() {
        self.collectionView = UICollectionView(frame: CGRect(x: 0, y:
            0, width: self.view.frame.width, height:
            self.view.frame.height), collectionViewLayout:
            UICollectionViewFlowLayout())
        self.collectionView?.dataSource = self
        self.collectionView?.backgroundColor = .white
        self.collectionView?.register(UICollectionViewCell.self,
            forCellWithReuseIdentifier: "BoxCell")
        self.view.addSubview(self.collectionView!)
    }

    func collectionView(_ collectionView: UICollectionView,
        numberOfItemsInSection section: Int) -> Int {
        return 1
    }

    func collectionView(_ collectionView: UICollectionView,
        cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        let cell =
            collectionView.dequeueReusableCell(withReuseIdentifier:
            "BoxCell", for: indexPath)
        cell.backgroundColor = .red

        return cell
```



```
}
// Present the view controller in the Live View window
PlaygroundPage.current.liveView =
     CollectionViewExampleController()
```

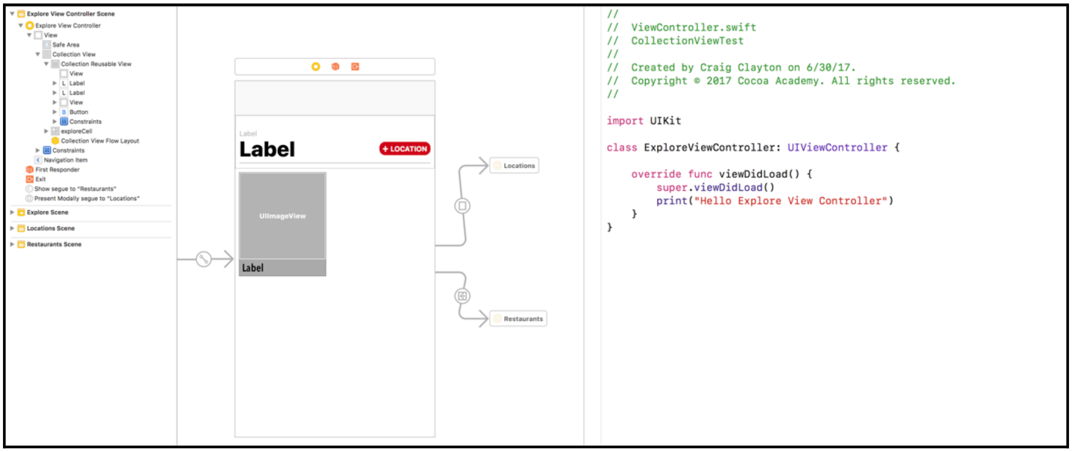


Choose a template for your new file:

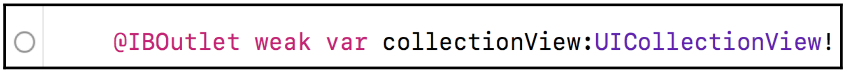


```
import UIKit

class ExploreViewController: 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.
    }
    */
}
```

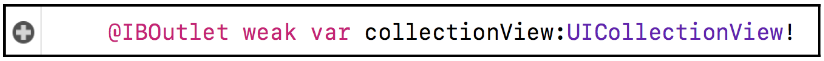
```
import UIKit

class ExploreViewController: UIViewController {

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



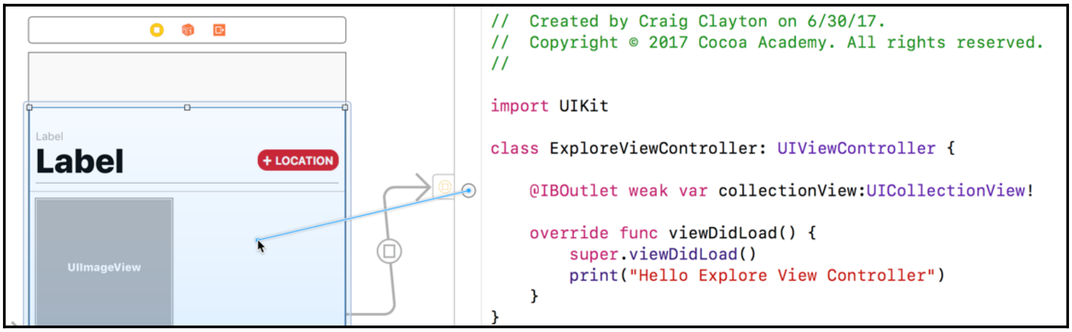
```
//
//  ViewController.swift
//  CollectionViewTest
//
//  Created by Craig Clayton on 6/30/17.
//  Copyright © 2017 Cocoa Academy. All rights reserved.
//

import UIKit

class ExploreViewController: UIViewController {

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



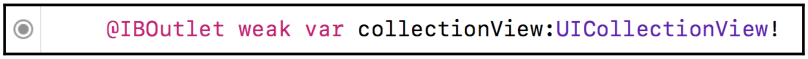
```
@IBOutlet weak var collectionView:UICollectionView!
```



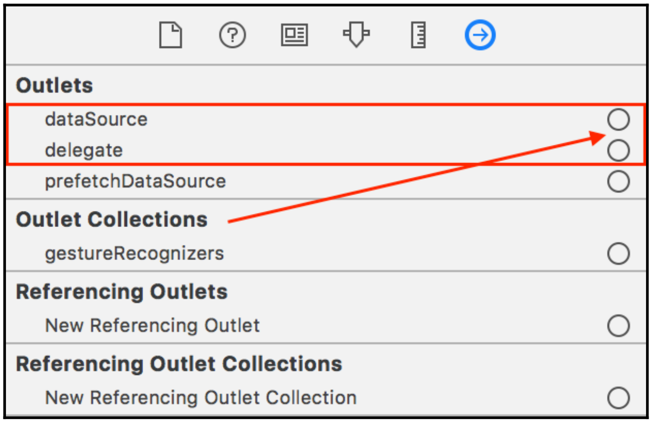
```
@IBOutlet weak var collectionView:UICollectionView!
```

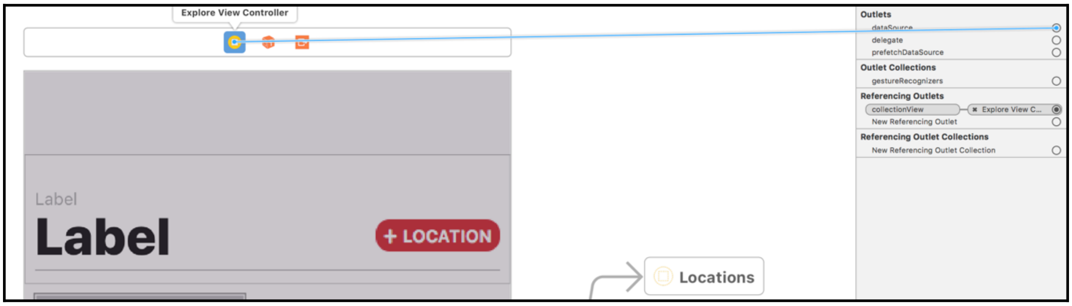


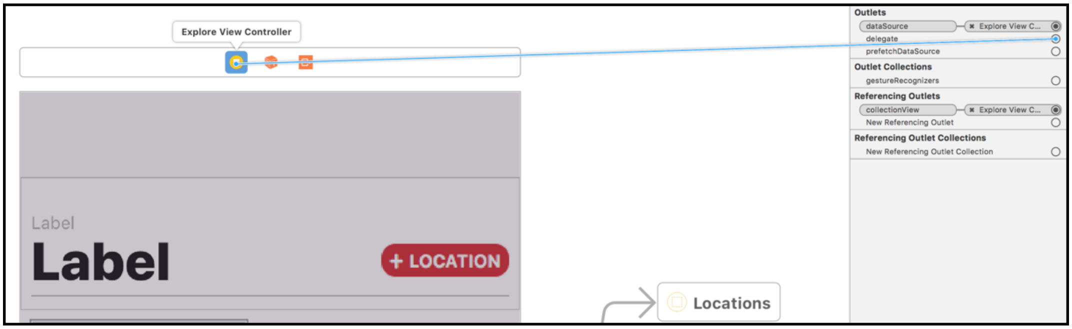
```
//  Created by Craig Clayton on 6/30/17.
//  Copyright © 2017 Cocoa Academy. All rights reserved.
//

import UIKit

class ExploreViewController: UIViewController {

    @IBOutlet weak var collectionView:UICollectionView!

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

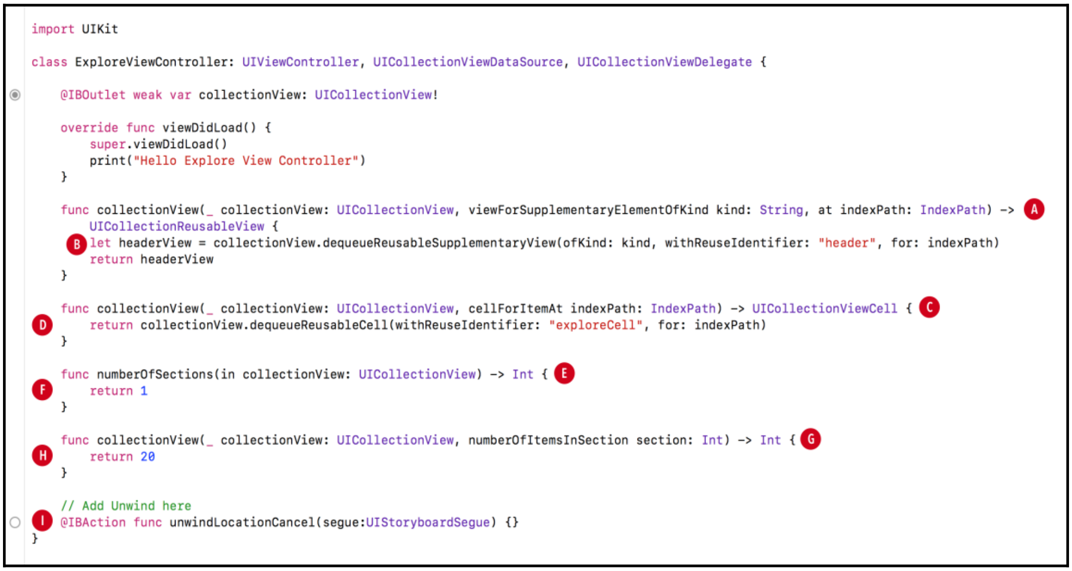


```
@IBOutlet weak var collectionView:UICollectionView!
```



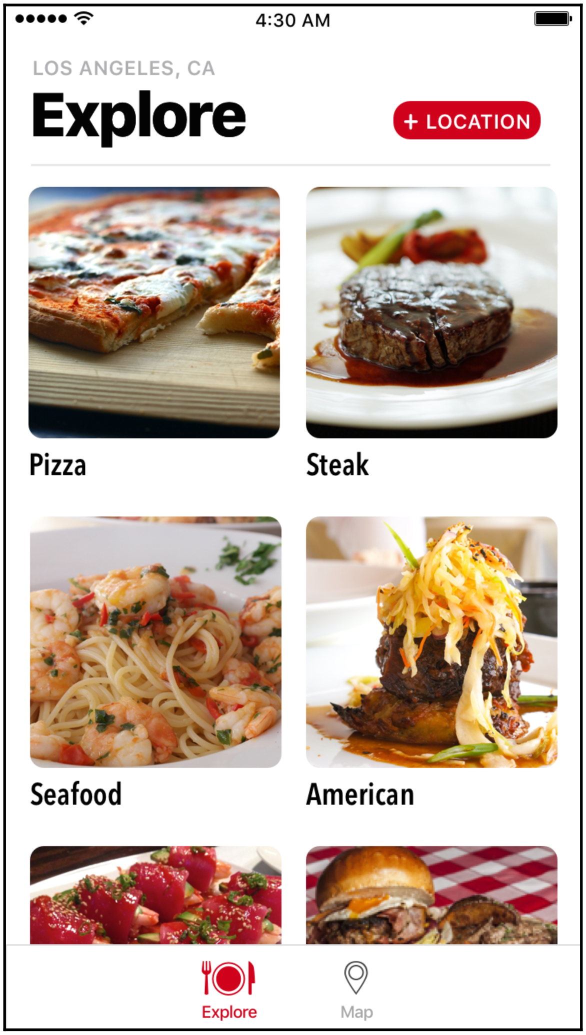


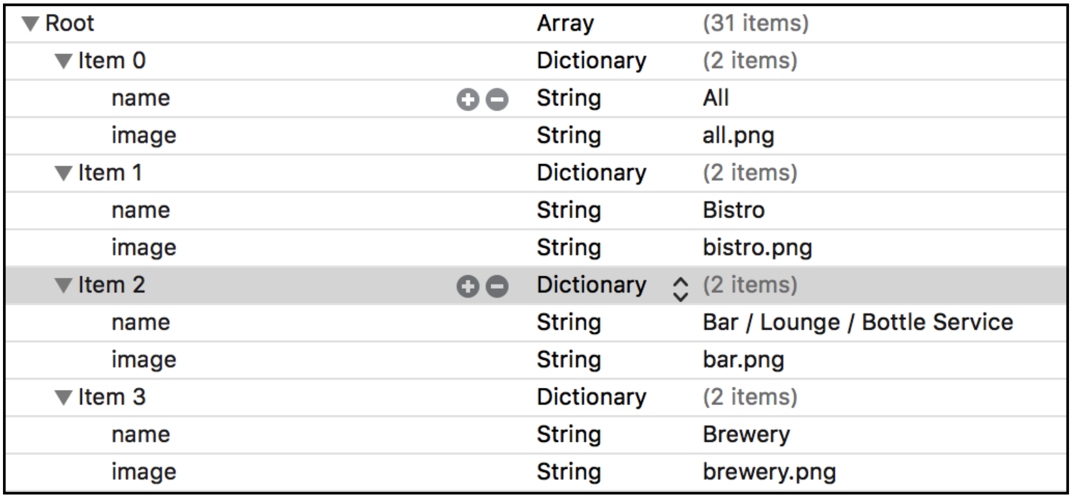




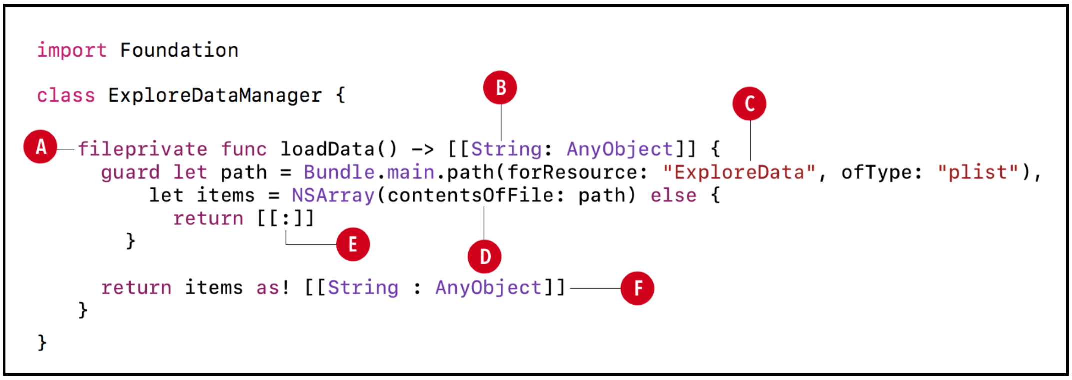
```
import UIKit

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) ->
        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
    @IBAction func unwindLocationCancel(segue:UIStoryboardSegue) {}
}
```

Chapter 12: Getting Data into Our Grid

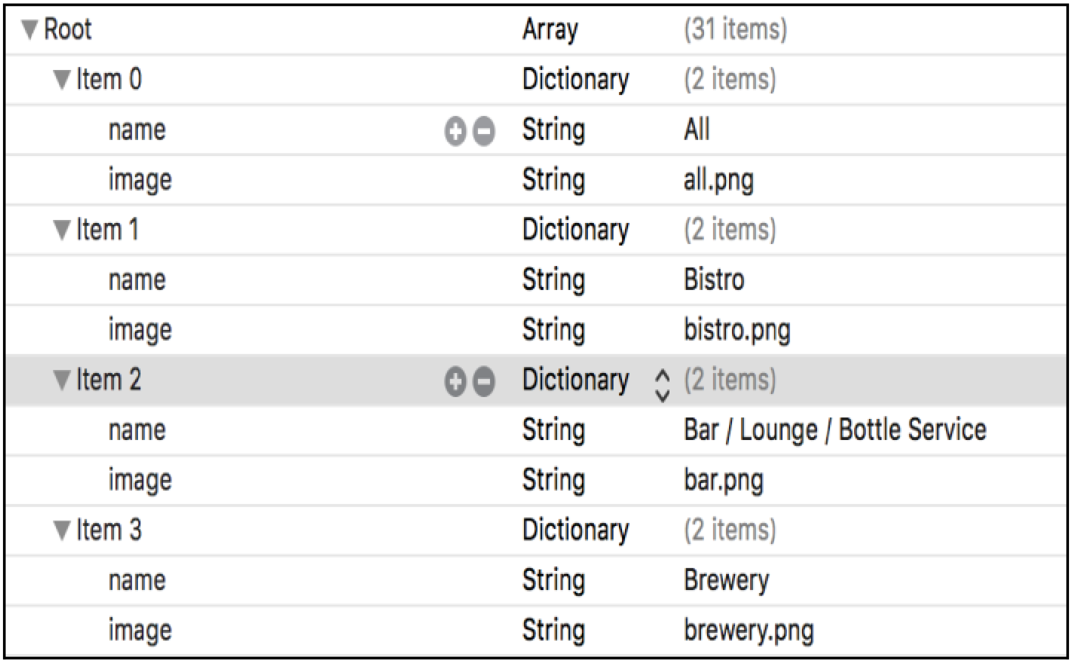




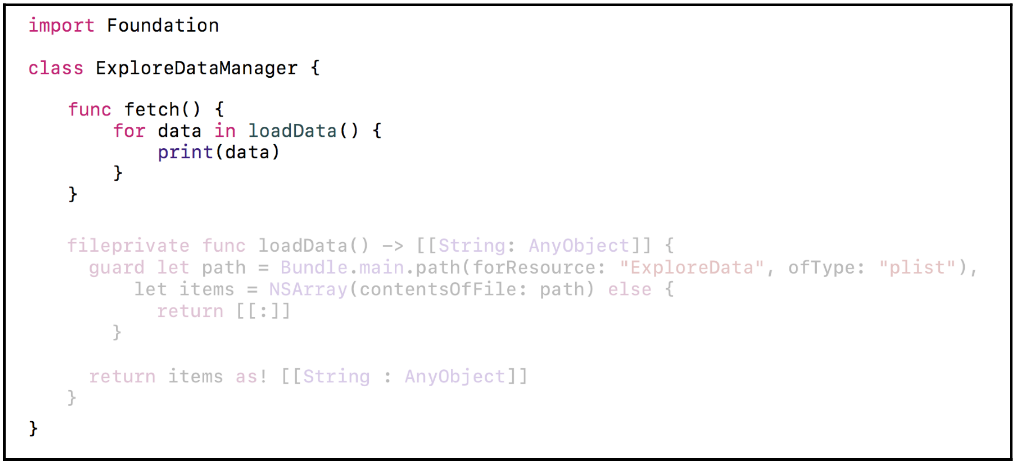
▼ Root	Array	(31 items)
▼ Item 0	Dictionary	(2 items)
name	String	All
image	String	all.png
▼ Item 1	Dictionary	(2 items)
name	String	Bistro
image	String	bistro.png
▼ Item 2	Dictionary	(2 items)
name	String	Bar / Lounge / Bottle Service
image	String	bar.png
▼ Item 3	Dictionary	(2 items)
name	String	Brewery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



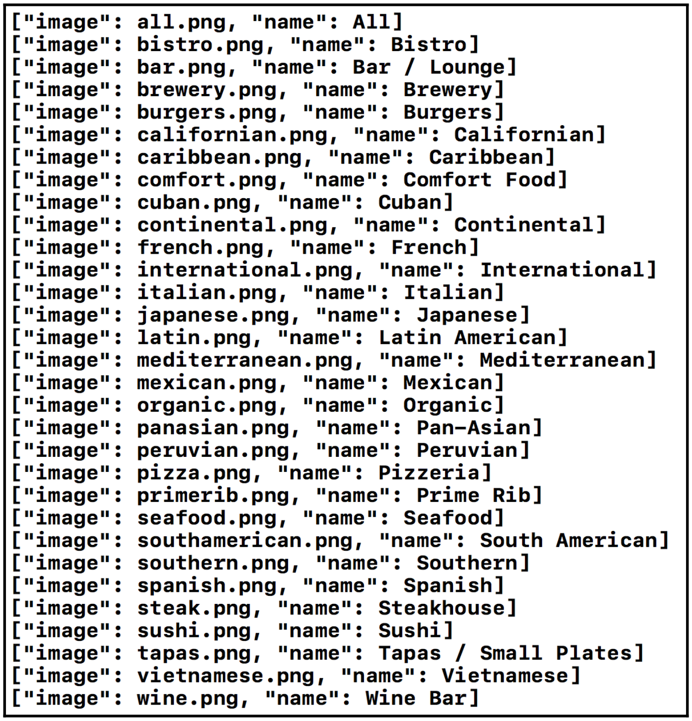
```
import Foundation

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]]
    }
}
```



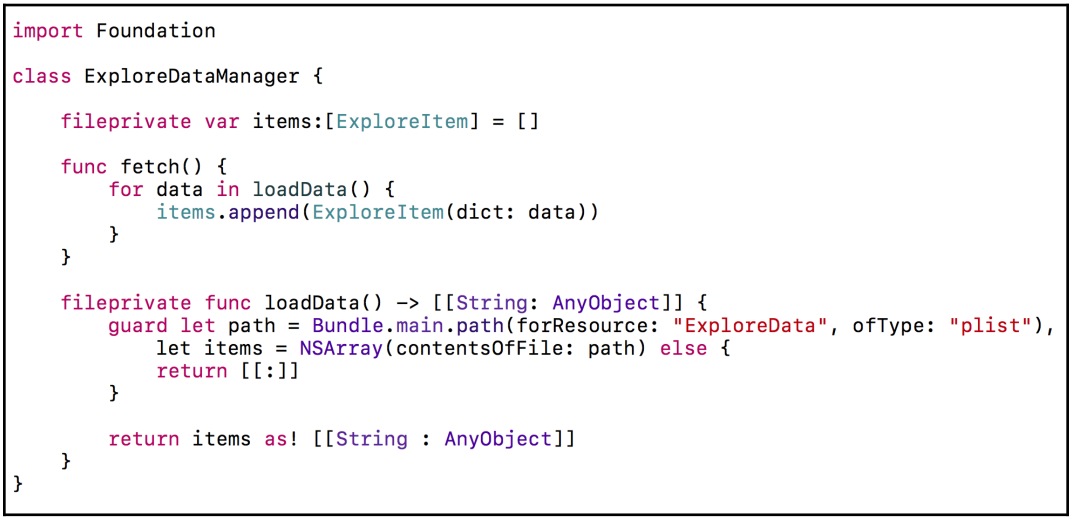
▼ Root	Array	(31 items)
▼ Item 0	Dictionary	(2 items)
name	String	All
image	String	all.png
▼ Item 1	Dictionary	(2 items)
name	String	Bistro
image	String	bistro.png
▼ Item 2	Dictionary	(2 items)
name	String	Bar / Lounge / Bottle Service
image	String	bar.png
▼ Item 3	Dictionary	(2 items)
name	String	Brewery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



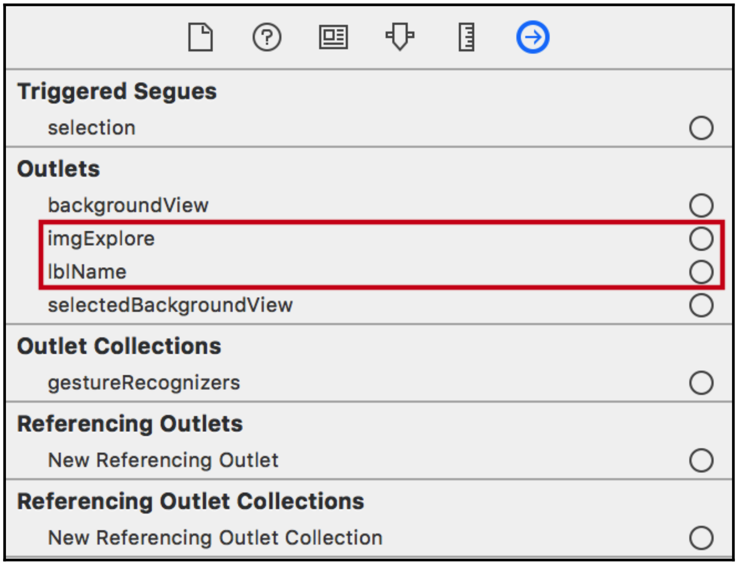
```
import Foundation

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 ○

Outlets

backgroundView ○

imgExplore ○

lblName ○

selectedBackgroundView ○

Outlet Collections

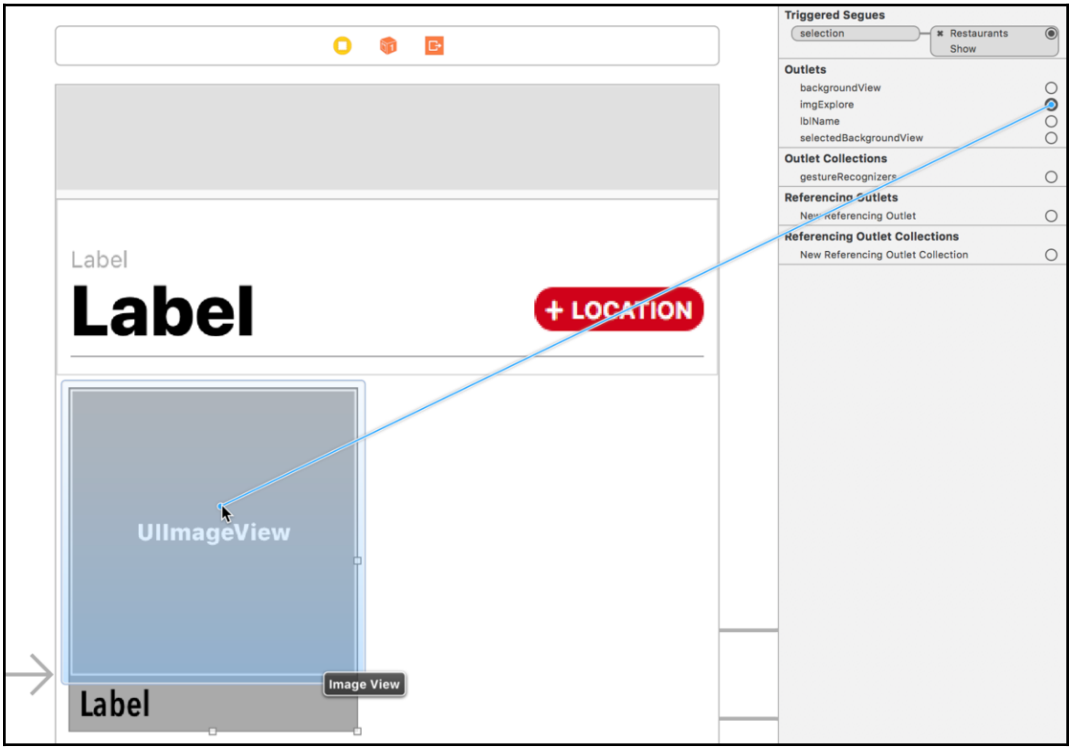
gestureRecognizers ○

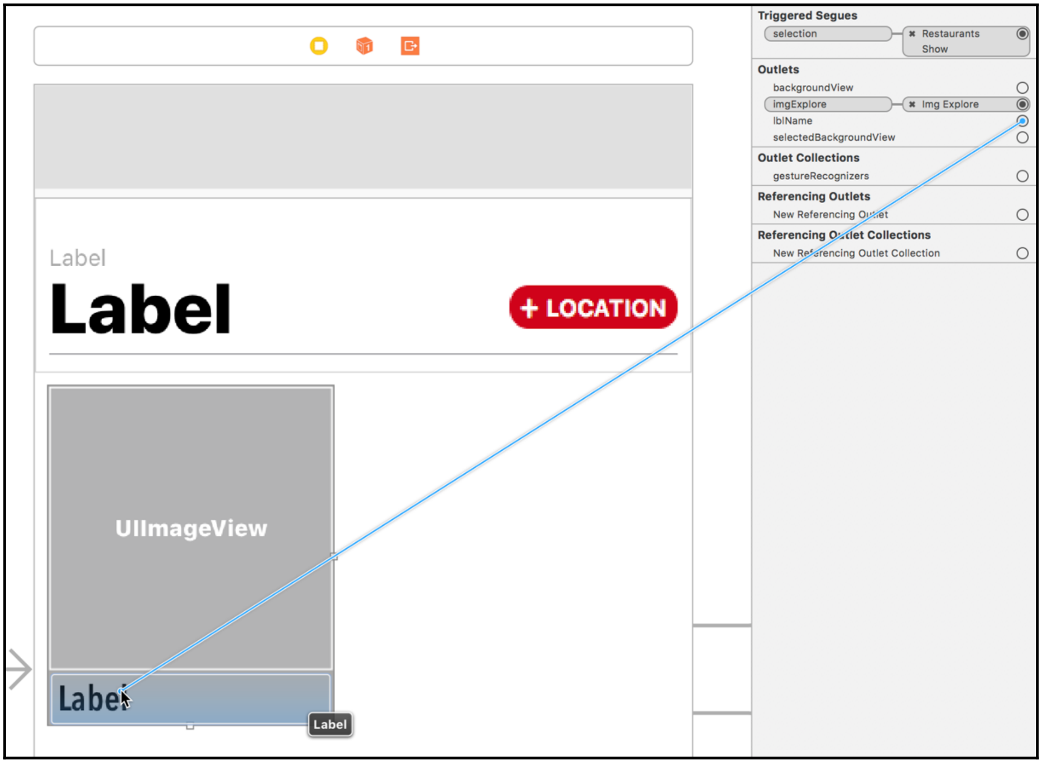
Referencing Outlets

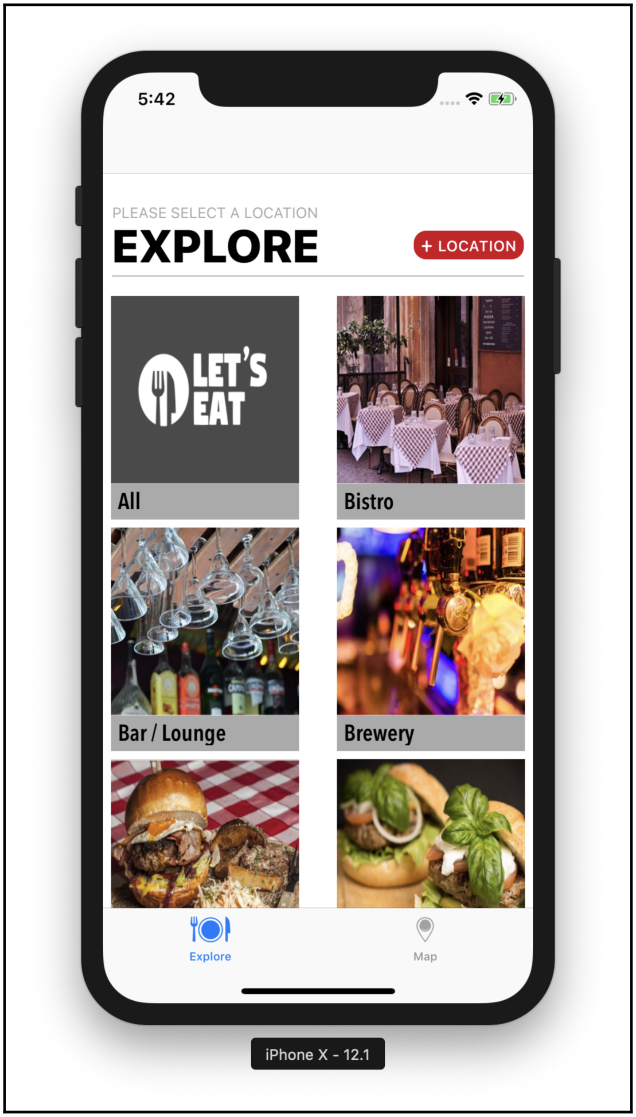
New Referencing Outlet ○

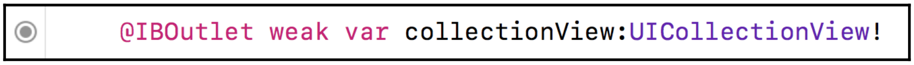
Referencing Outlet Collections

New Referencing Outlet Collection ○

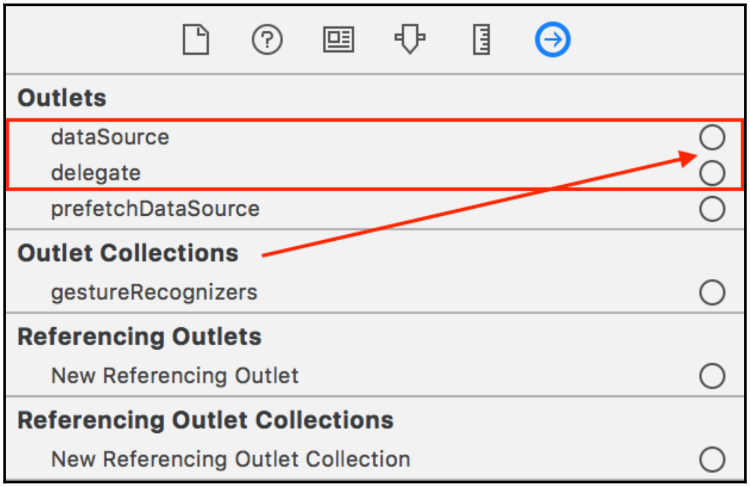



Label
Label
+ LOCATION
UIImageView
Label
Label
Triggered Segues
selection
Restaurants Show
Outlets
backgroundView
imgExplore
Img Explore
lblName
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet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





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



Outlets

dataSource ○

delegate ○

prefetchDataSource ○

Outlet Collections

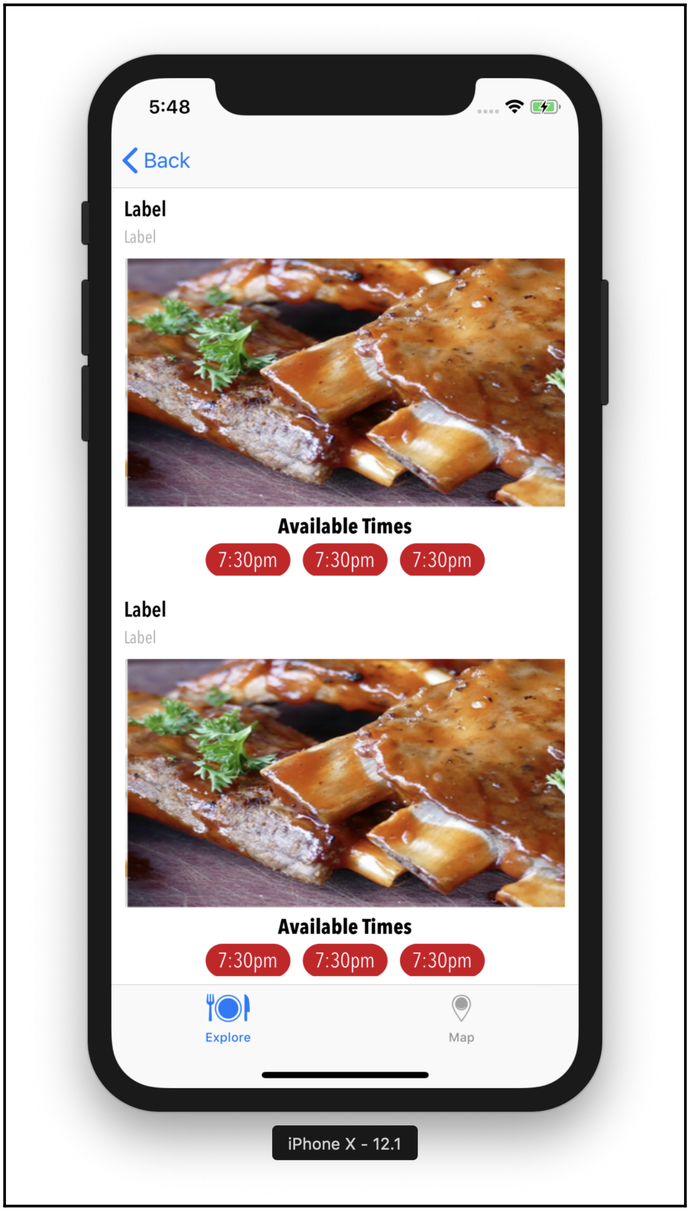
gestureRecognizers ○

Referencing Outlets

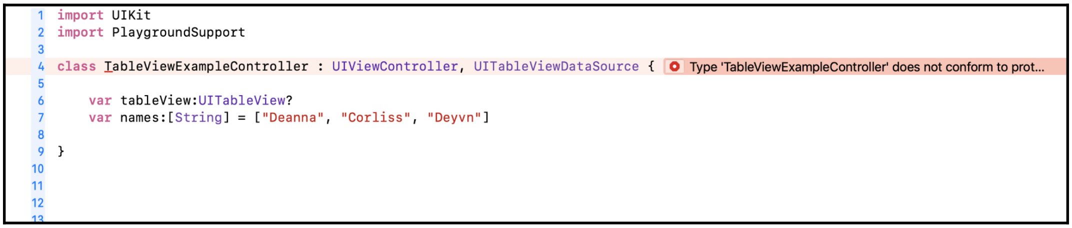
New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○

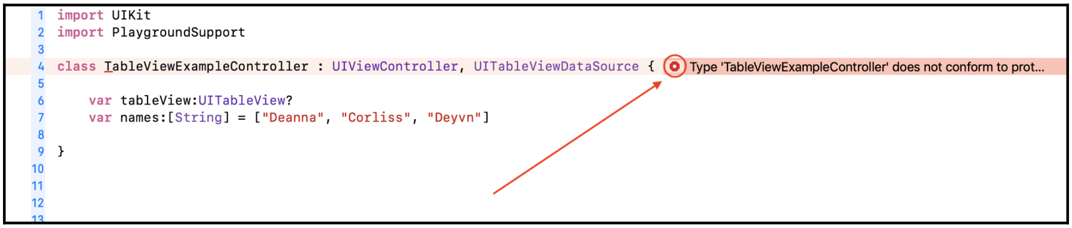

5:48
Back
Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Label
Label
Available Times
7:30pm
7:30pm
7:30pm
Explore
Map
iPhone X - 12.1

Chapter 13: Getting Started with the List



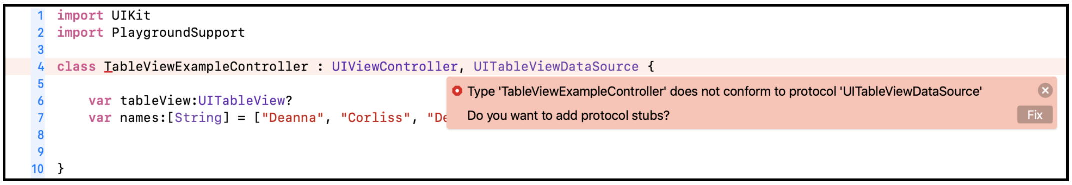
```
import UIKit
import PlaygroundSupport

class TableViewExampleController : UIViewController, UITableViewDataSource {

    var tableView:UITableView?
    var names:[String] = ["Deanna", "Corliss", "Deyvn"]

}
```

Type 'TableViewExampleController' does not conform to prot...



```
import UIKit
import PlaygroundSupport

class TableViewExampleController : UIViewController, UITableViewDataSource {

    var tableView:UITableView?
    var names:[String] = ["Deanna", "Corliss", "Deyvn"]

}
```

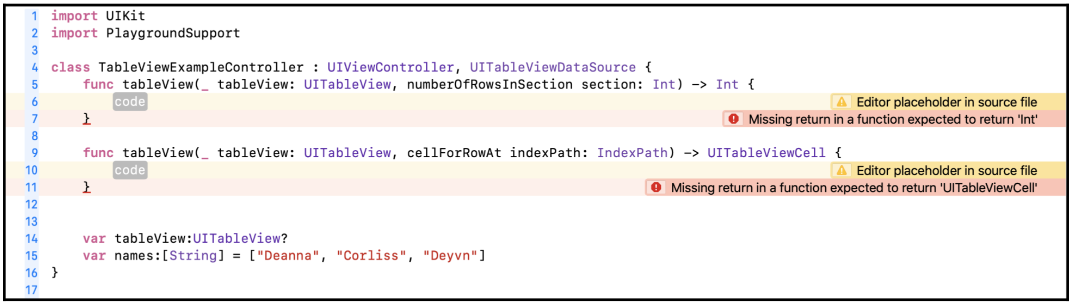
Type 'TableViewExampleController' does not conform to prot...



```
import UIKit
import PlaygroundSupport

class TableViewExampleController : UIViewController, UITableViewDataSource {

    var tableView:UITableView?
    var names:[String] = ["Deanna", "Corliss", "De



}
```

Type 'TableViewExampleController' does not conform to protocol 'UITableViewDataSource'

Do you want to add protocol stubs?



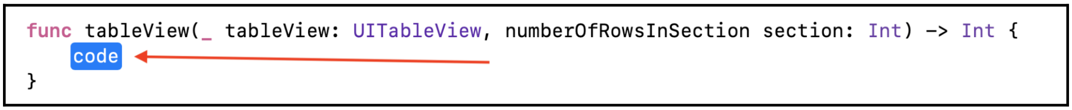
```
import UIKit
import PlaygroundSupport

class TableViewExampleController : UIViewController, UITableViewDataSource {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        code
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        code
    }


    var tableView:UITableView?
    var names:[String] = ["Deanna", "Corliss", "Deyvn"]
}
```

Editor placeholder in source file

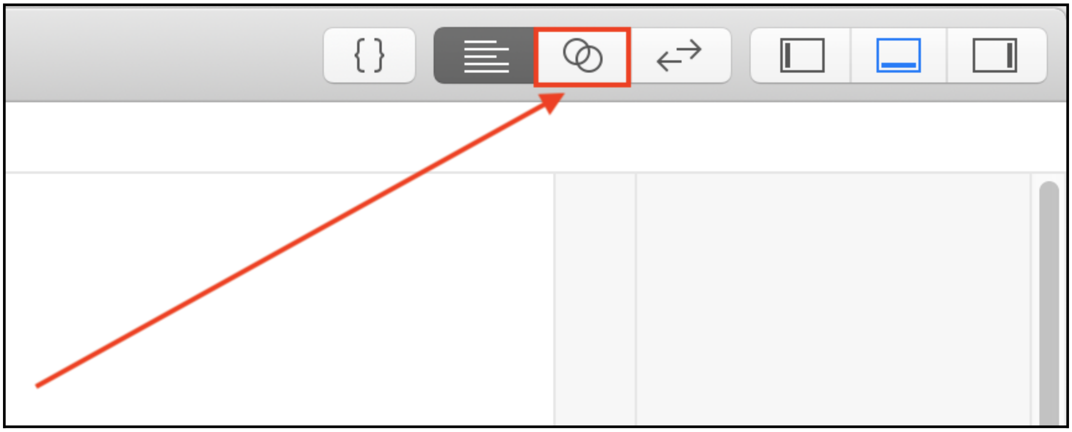
Missing return in a function expected to return 'Int'

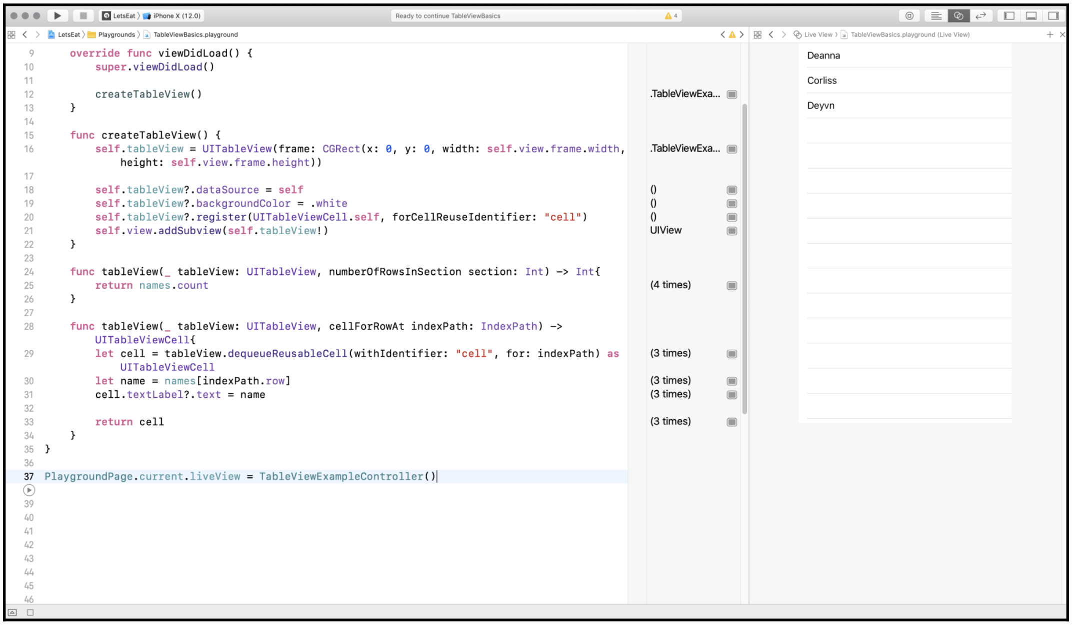
Editor placeholder in source file

Missing return in a function expected to return 'UITableViewCell'



```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    code
}
```

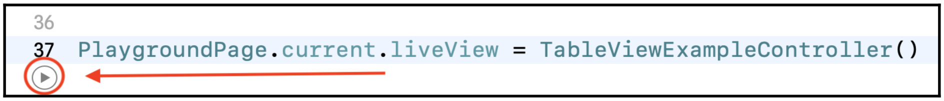



LetsEat 〉 iPhone X (12.0)
Ready to continue TableViewBasics
LetsEat 〉 Playgrounds 〉 TableViewBasics.playground
Live View 〉 TableViewBasics.playground (Live View)

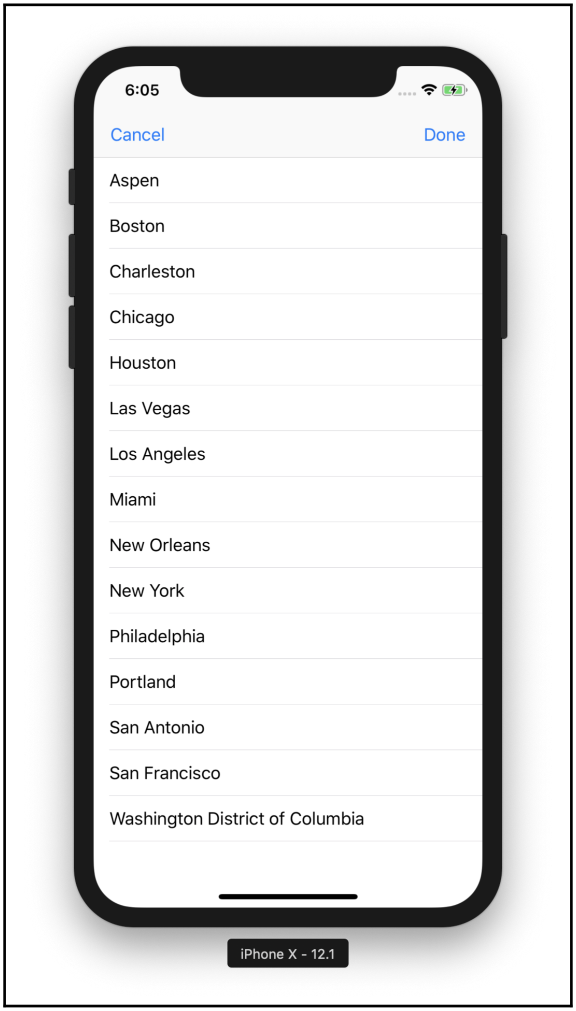
```
    override func viewDidLoad() {
        super.viewDidLoad()

        createTableView()
    }

    func createTableView() {
        self.tableView = UITableView(frame: CGRect(x: 0, y: 0, width: self.view.frame.width,
            height: self.view.frame.height))

        self.tableView?.dataSource = self
        self.tableView?.backgroundColor = .white
        self.tableView?.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
        self.view.addSubview(self.tableView!)
    }

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

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
        UITableViewCell{
        let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath) as
            UITableViewCell
        let name = names[indexPath.row]
        cell.textLabel?.text = name

        return cell
    }
}

PlaygroundPage.current.liveView = TableViewExampleController()
```

```
.TableViewExa...
.TableViewExa...
()
()
()
UIView
(4 times)
(3 times)
(3 times)
(3 times)
(3 times)
```

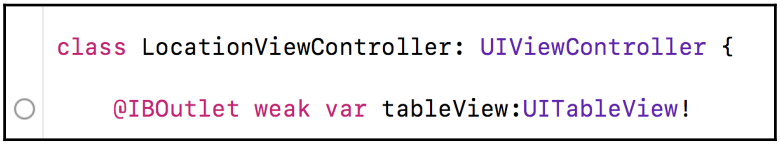
```
Deanna
Corliss
Deyvn
```



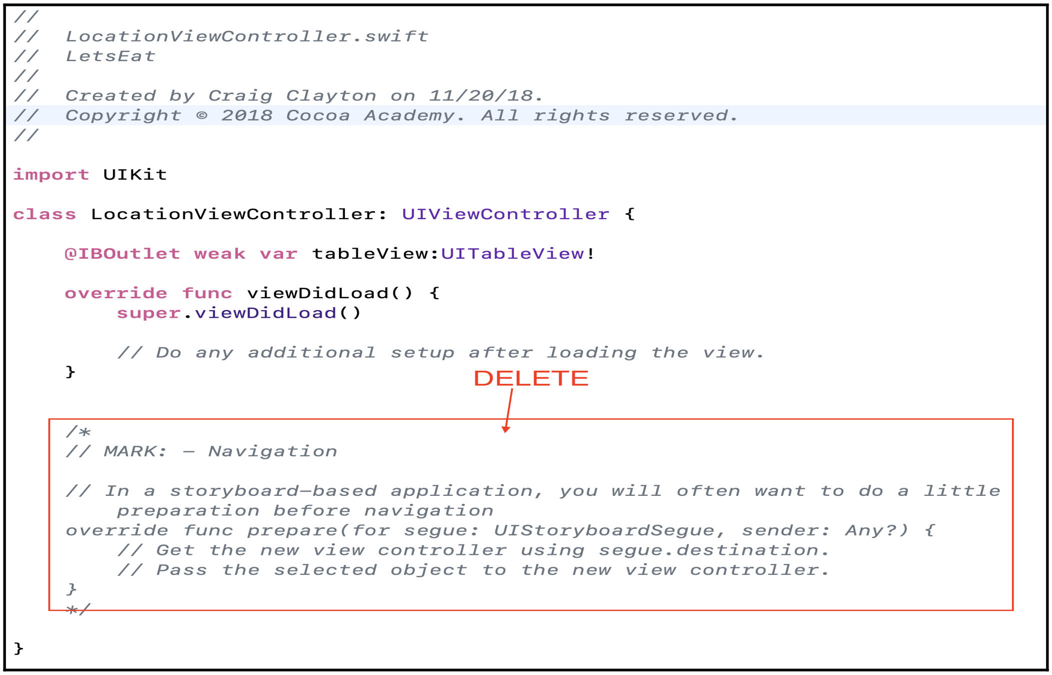
```
PlaygroundPage.current.liveView = TableViewExampleController()
```



Aspen
Boston
Charleston
Chicago
Houston
Las Vegas
Los Angeles
Miami
New Orleans
New York
Philadelphia
Portland
San Antonio
San Francisco
Washington District of Columbia

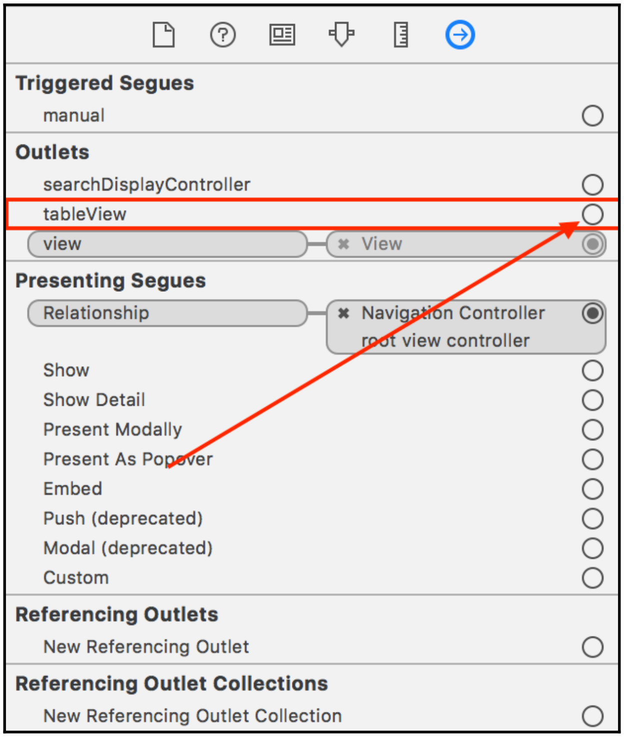


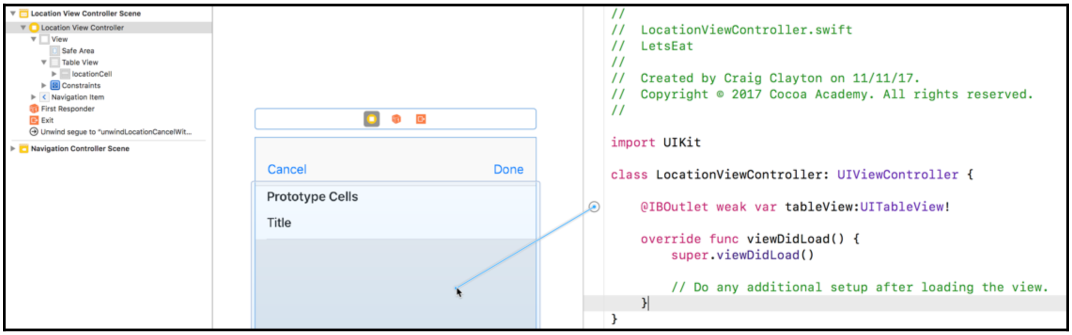
```
class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!
```



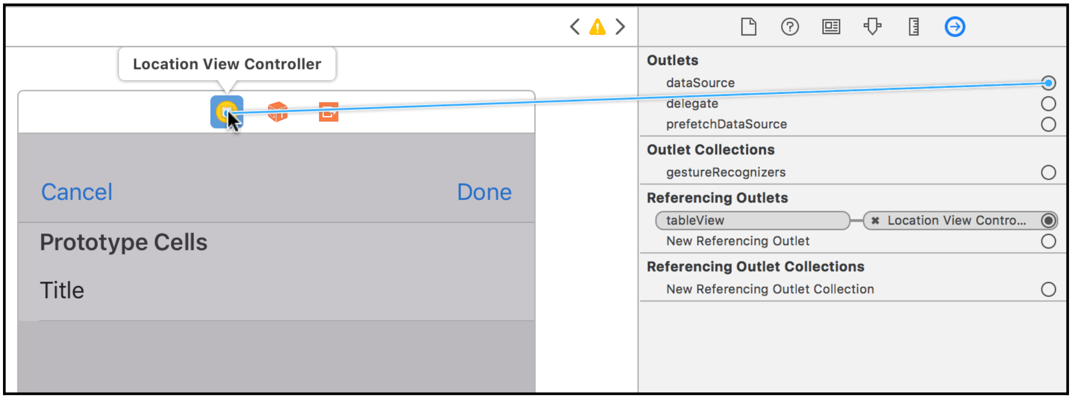
```
//
//  LocationViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 11/20/18.
//  Copyright © 2018 Cocoa Academy. All rights reserved.
//

import UIKit

class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!

    override func viewDidLoad() {
        super.viewDidLoad()

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


    /*
    // 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.destination.
        // Pass the selected object to the new view controller.
    }
    */

}
```

DELETE





```
//
//  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.
    }
}
```



Outlets

dataSource ●
delegate ○
prefetchDataSource ○

Outlet Collections

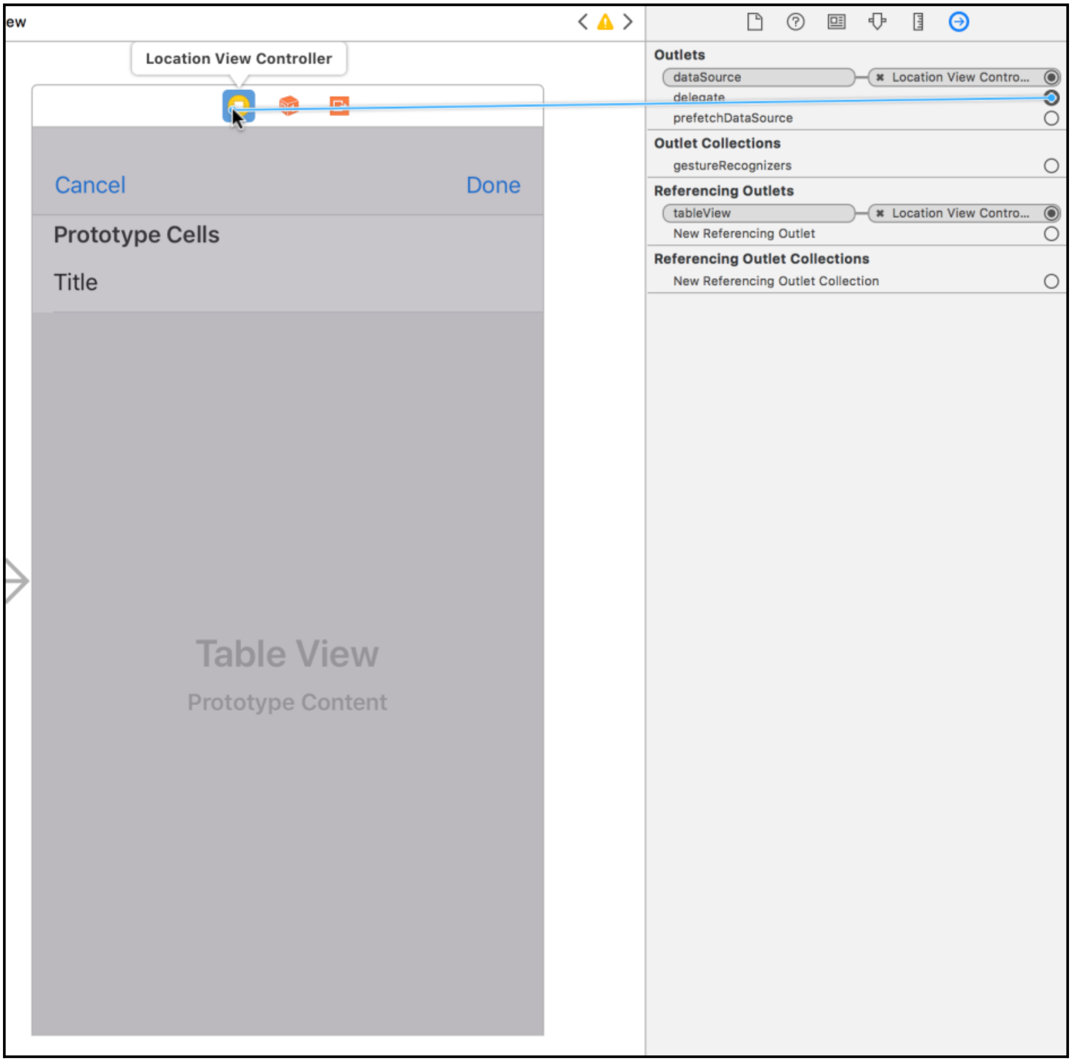
gestureRecognizers ○

Referencing Outlets

tableView — ✖ Location View Contro... ●
New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○



Outlets

dataSource — Location View Contro...

delegate

prefetchDataSource

Outlet Collections

gestureRecognizers

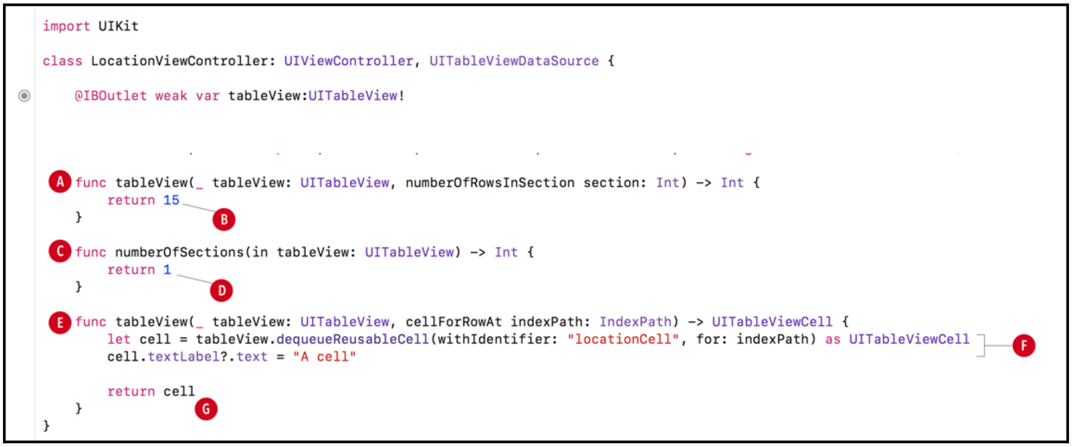
Referencing Outlets

tableView — Location View Contro...

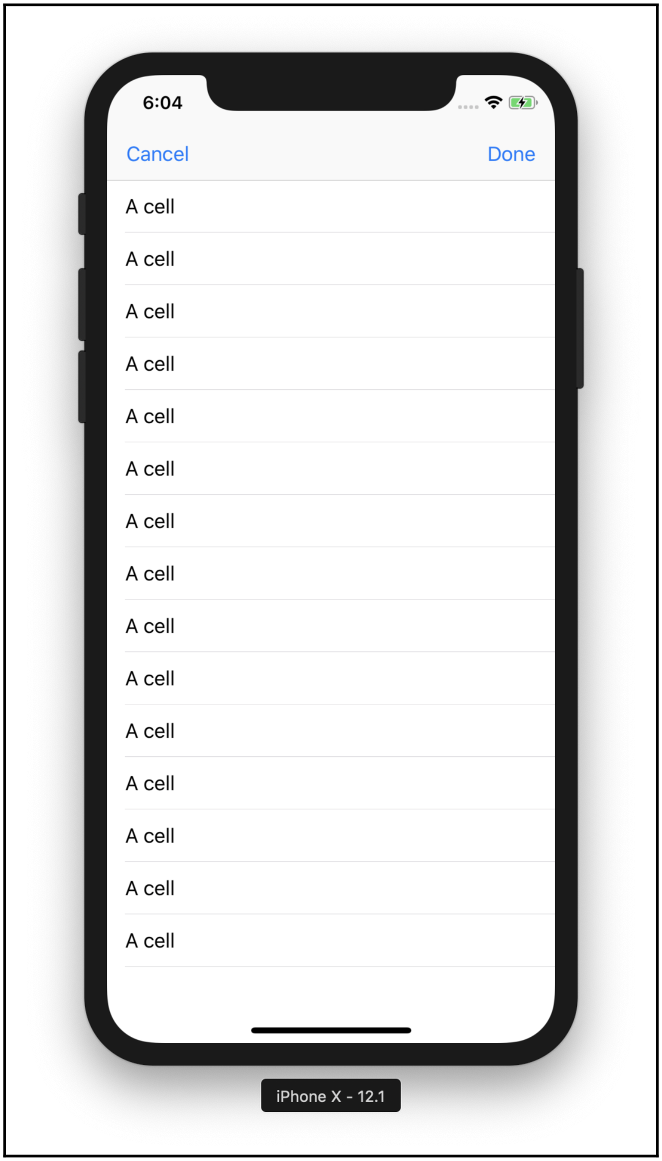
New Referencing Outlet

Referencing Outlet Collections

New Referencing Outlet Collection



```
import UIKit

class LocationViewController: UIViewController, UITableViewDataSource {

    @IBOutlet weak var tableView:UITableView!


    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
    }
}
```



A cell

A cell

A cell

A cell

A cell

A cell

A cell

A cell

A cell

A cell

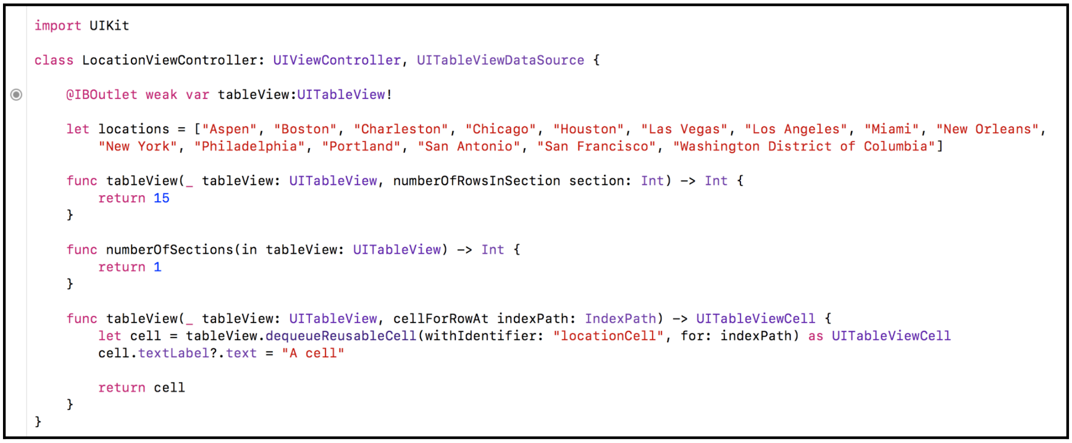
A cell

A cell

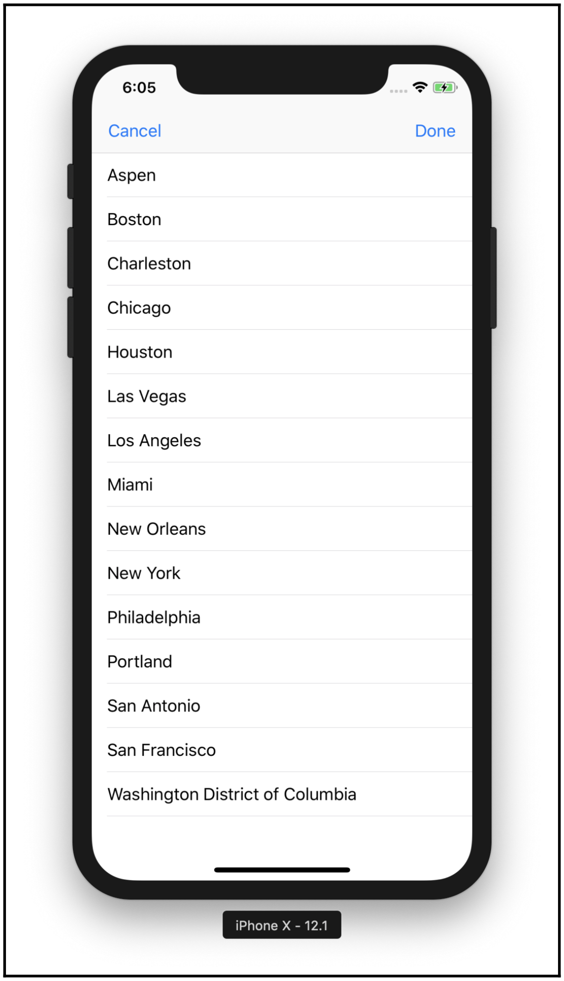
A cell

A cell

A cell



```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
    }
}
```



Aspen

Boston

Charleston

Chicago

Houston

Las Vegas

Los Angeles

Miami

New Orleans

New York

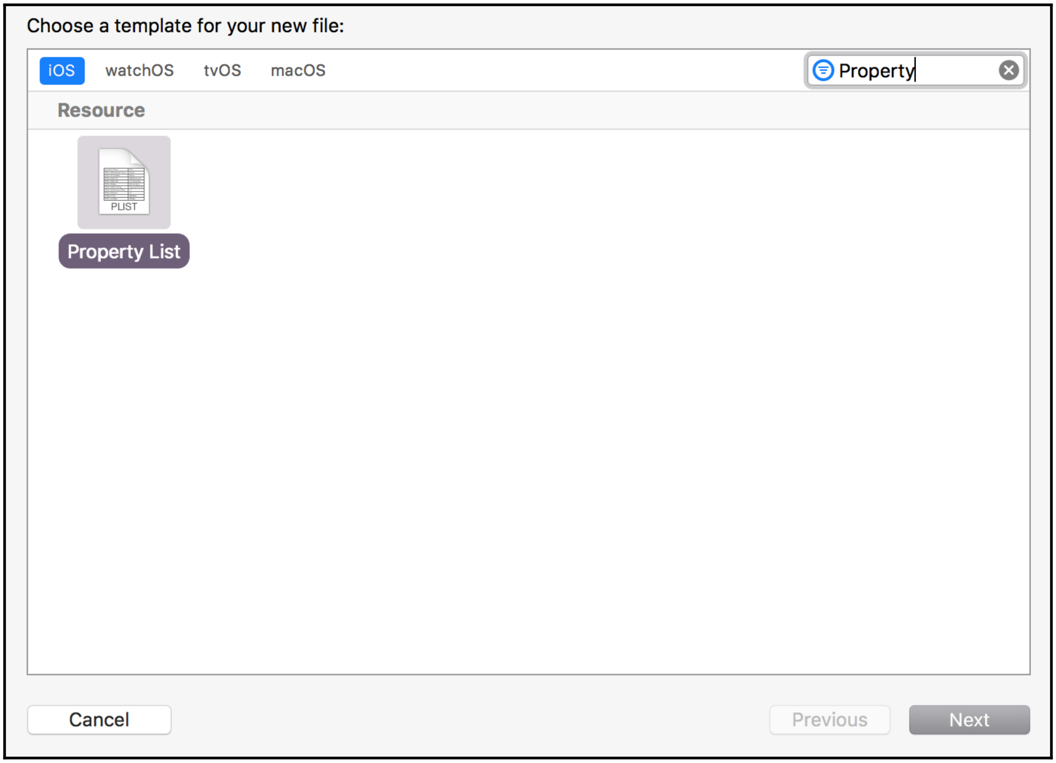
Philadelphia

Portland

San Antonio

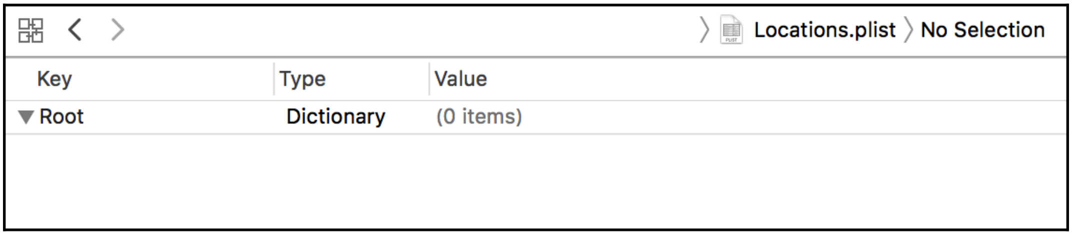
San Francisco

Washington District of Columbia



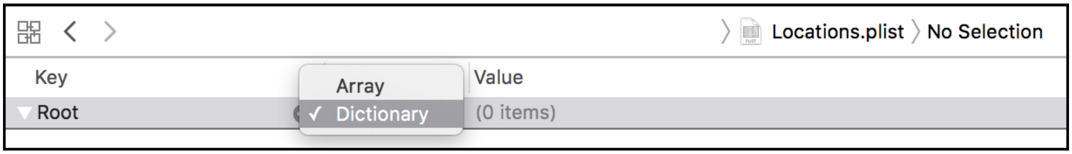
Choose a template for your new file:

Property



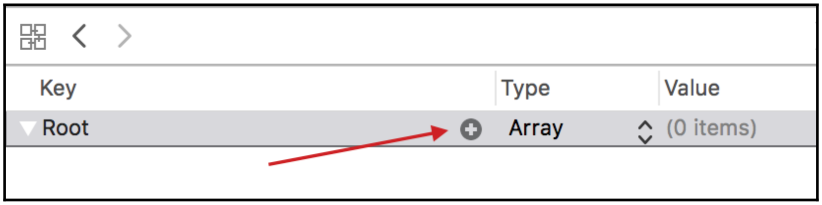
Locations.plist 〉 No Selection

Key	Type	Value
▼ Root	Dictionary	(0 items)

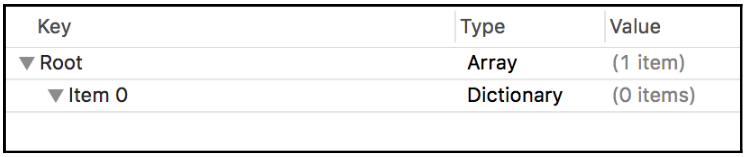


Locations.plist 〉 No Selection

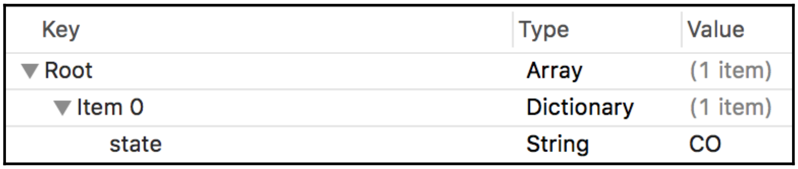
Key		Value
Root	✓ Dictionary	(0 items)



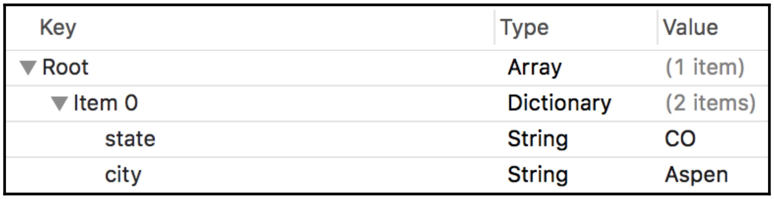
Key	Type	Value
▼ Root	Array	(0 items)



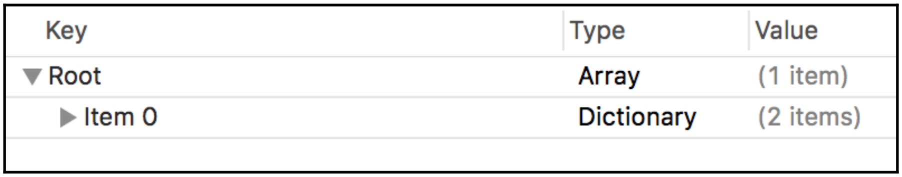
Key	Type	Value
▼ Root	Array	(1 item)
▼ Item 0	Dictionary	(0 items)



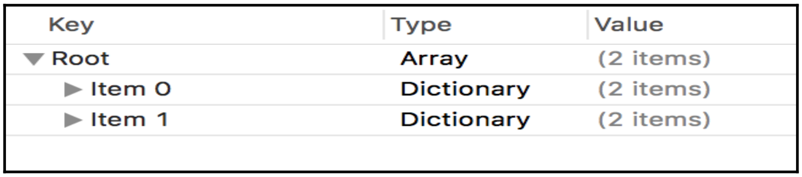
Key	Type	Value
▼ Root	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
state	String	CO



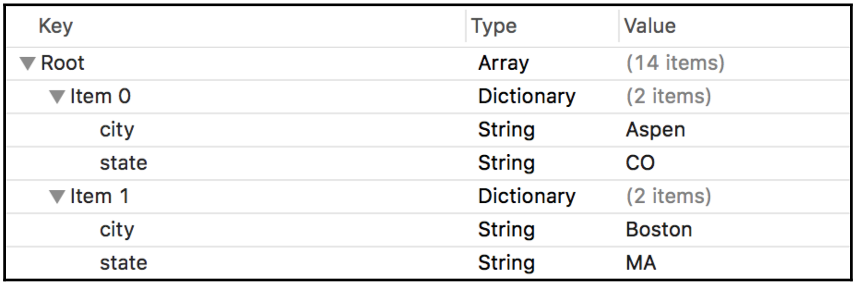
Key	Type	Value
▼ Root	Array	(1 item)
▼ Item 0	Dictionary	(2 items)
state	String	CO
city	String	Aspen



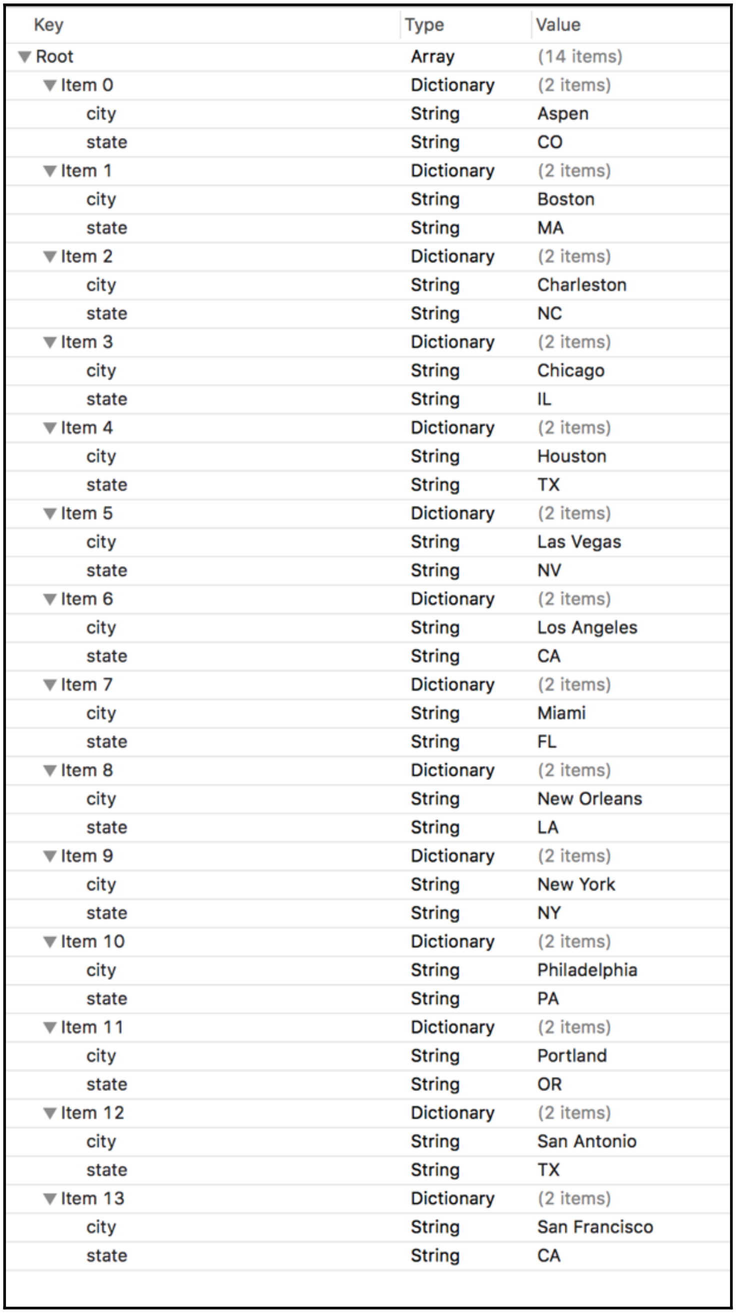
Key	Type	Value
▼ Root	Array	(1 item)
▶ Item 0	Dictionary	(2 items)



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

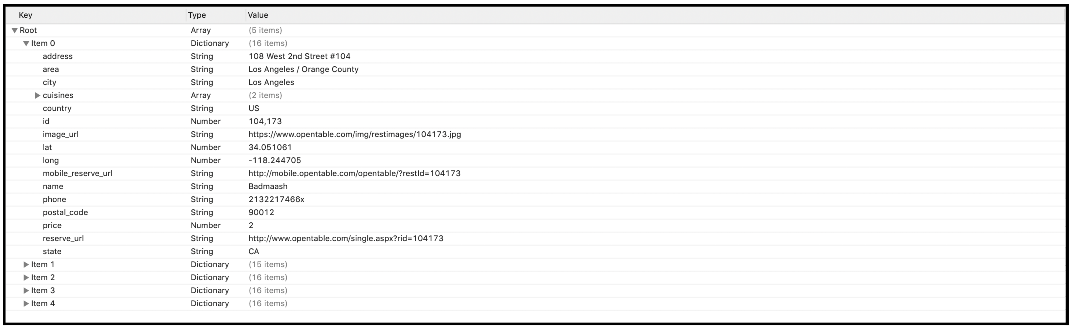


Key	Type	Value
▼ Root	Array	(14 items)
▼ Item 0	Dictionary	(2 items)
city	String	Aspen
state	String	CO
▼ Item 1	Dictionary	(2 items)
city	String	Boston
state	String	MA

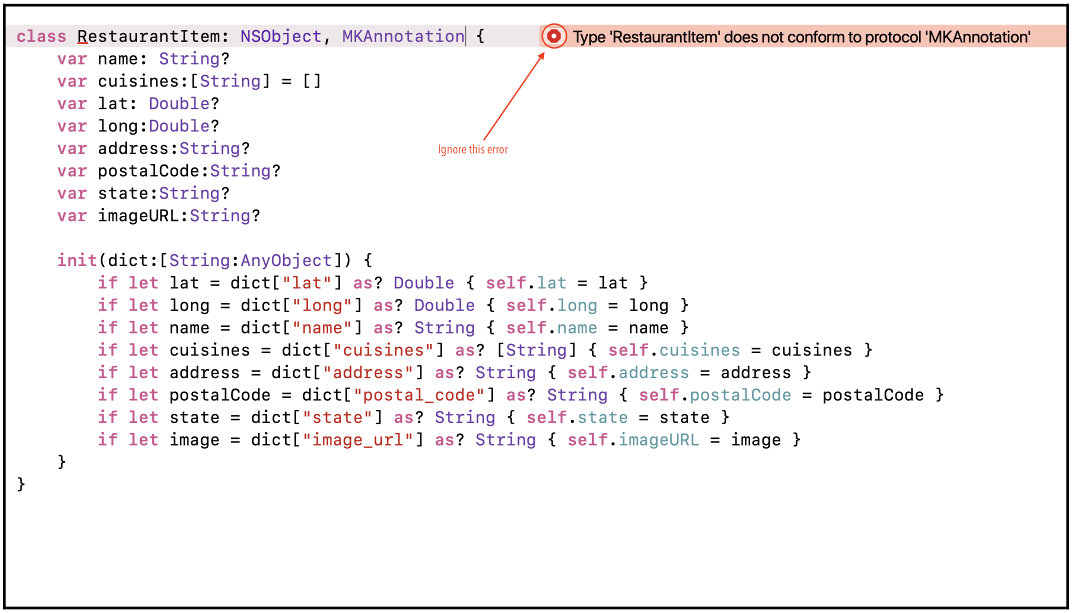


Key	Type	Value
▼ Root	Array	(14 items)
▼ Item 0	Dictionary	(2 items)
city	String	Aspen
state	String	CO
▼ Item 1	Dictionary	(2 items)
city	String	Boston
state	String	MA
▼ Item 2	Dictionary	(2 items)
city	String	Charleston
state	String	NC
▼ Item 3	Dictionary	(2 items)
city	String	Chicago
state	String	IL
▼ 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

Chapter 14: Where Are We?



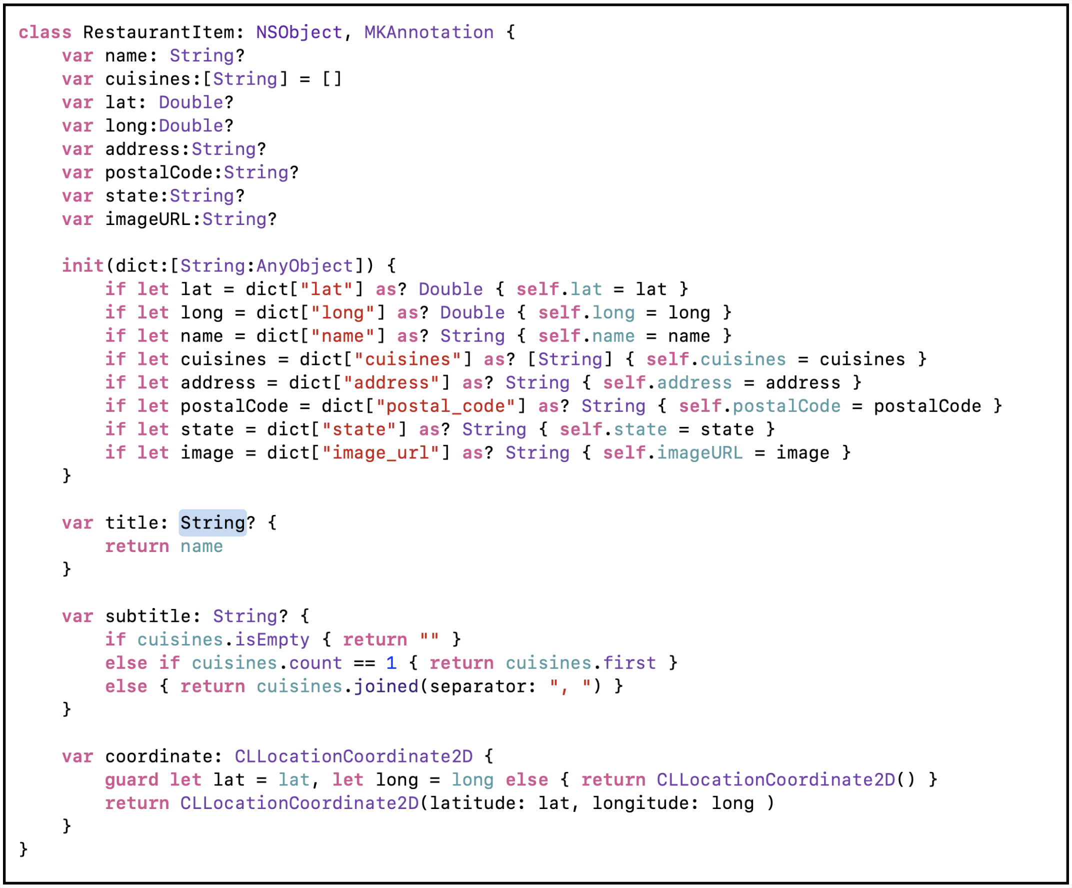
Key	Type	Value
▼ Root	Array	(5 items)
▼ Item 0	Dictionary	(16 items)
address	String	108 West 2nd Street #104
area	String	Los Angeles / Orange County
city	String	Los Angeles
▶ cuisines	Array	(2 items)
country	String	US
id	Number	104,173
image_url	String	https://www.opentable.com/img/restimages/104173.jpg
lat	Number	34.051061
long	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)



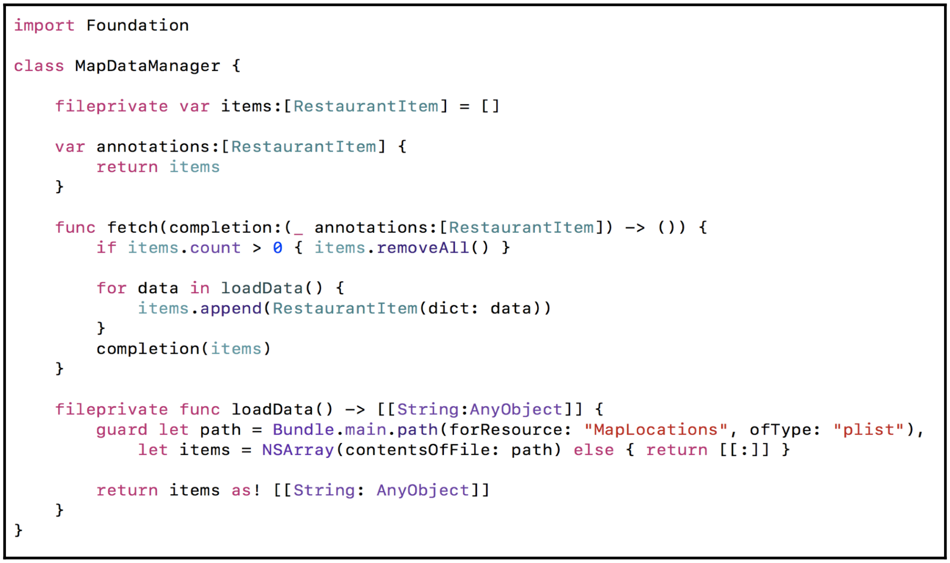
```
class RestaurantItem: NSObject, MKAnnotation {
    var name: String?
    var cuisines:[String] = []
    var lat: Double?
    var long:Double?
    var address:String?
    var postalCode:String?
    var state:String?
    var imageURL:String?

    init(dict:[String:AnyObject]) {
        if let lat = dict["lat"] as? Double { self.lat = lat }
        if let long = dict["long"] as? Double { self.long = 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 }
    }
}
```

Type 'RestaurantItem' does not conform to protocol 'MKAnnotation'

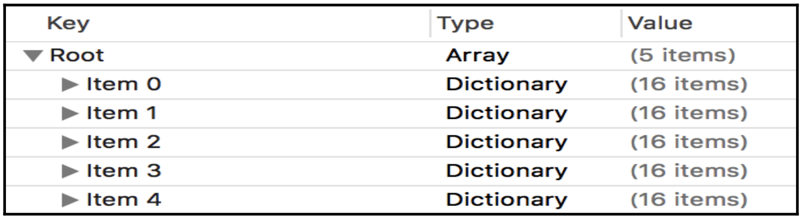
Ignore this error



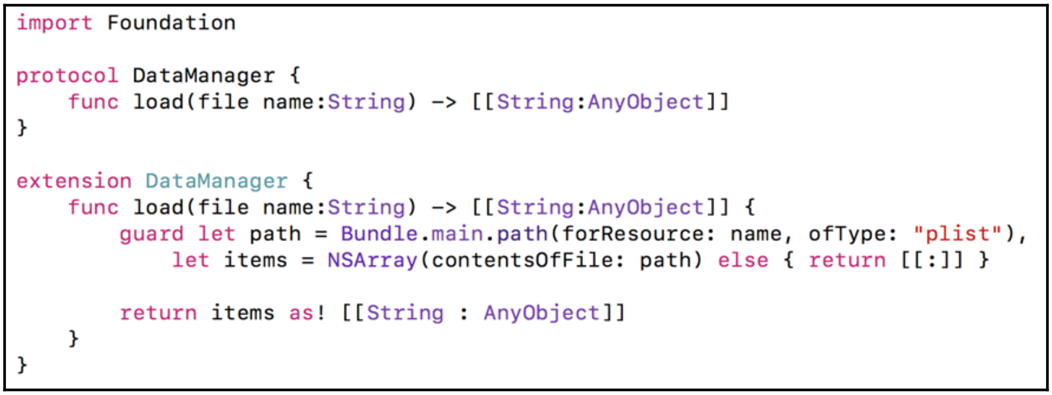
```
class RestaurantItem: NSObject, MKAnnotation {
    var name: String?
    var cuisines:[String] = []
    var lat: Double?
    var long:Double?
    var address:String?
    var postalCode:String?
    var state:String?
    var imageURL:String?

    init(dict:[String:AnyObject]) {
        if let lat = dict["lat"] as? Double { self.lat = lat }
        if let long = dict["long"] as? Double { self.long = 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 = lat, let long = long else { return CLLocationCoordinate2D() }
        return CLLocationCoordinate2D(latitude: lat, longitude: long )
    }
}
```



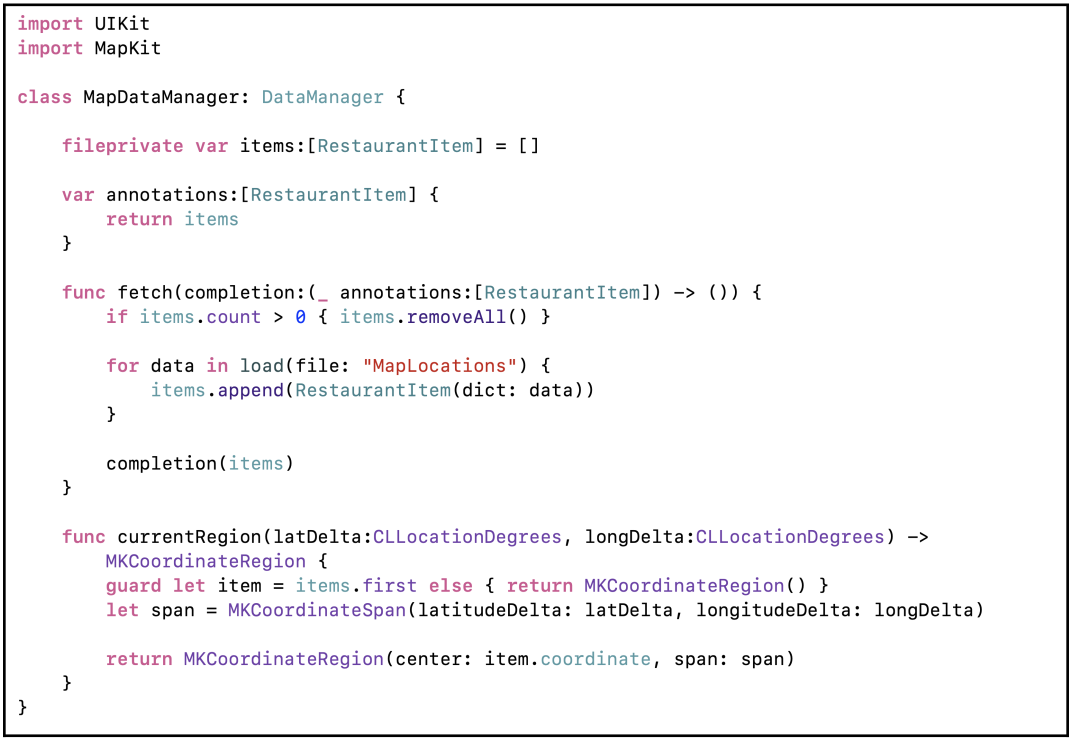
```
import Foundation

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]]
    }
}
```



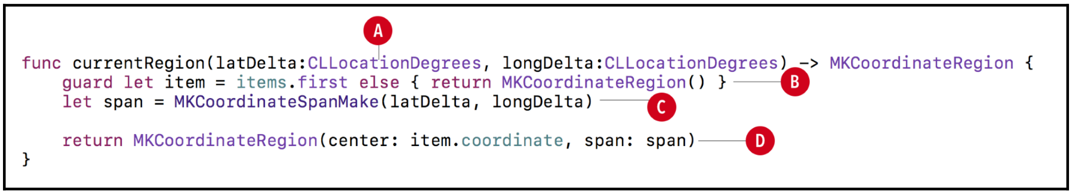
Key	Type	Value
▼ Root	Array	(5 items)
▶ Item 0	Dictionary	(16 items)
▶ Item 1	Dictionary	(16 items)
▶ Item 2	Dictionary	(16 items)
▶ Item 3	Dictionary	(16 items)
▶ Item 4	Dictionary	(16 items)



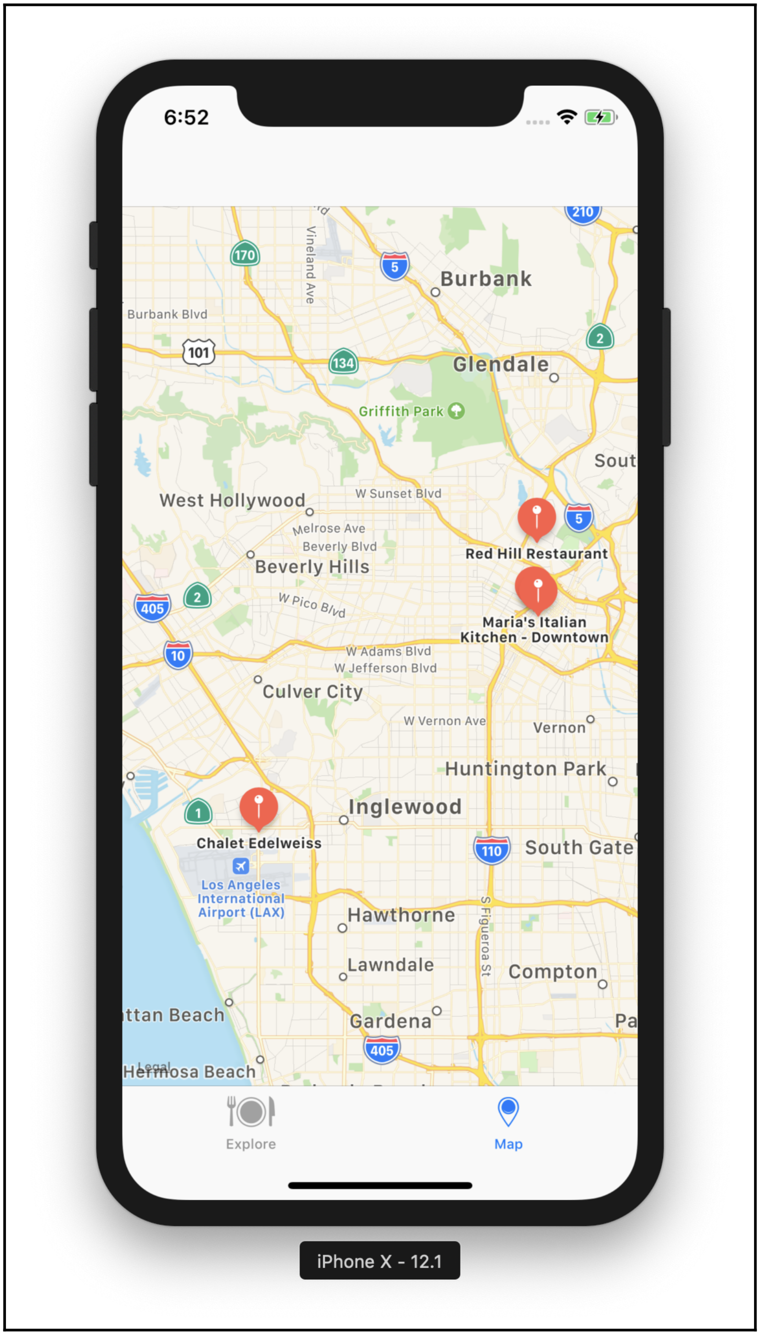
```
import Foundation

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]]
    }
}
```

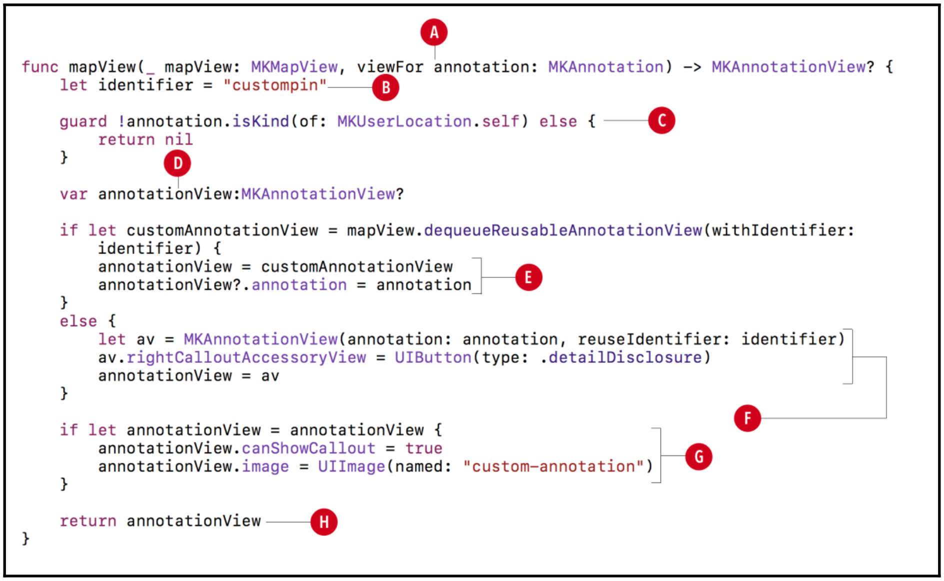


```
import UIKit
import MapKit

class MapDataManager: DataManager {

    fileprivate var items:[RestaurantItem] = []

    var annotations:[RestaurantItem] {
        return items
    }

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

        for data in load(file: "MapLocations") {
            items.append(RestaurantItem(dict: data))
        }

        completion(items)
    }

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

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

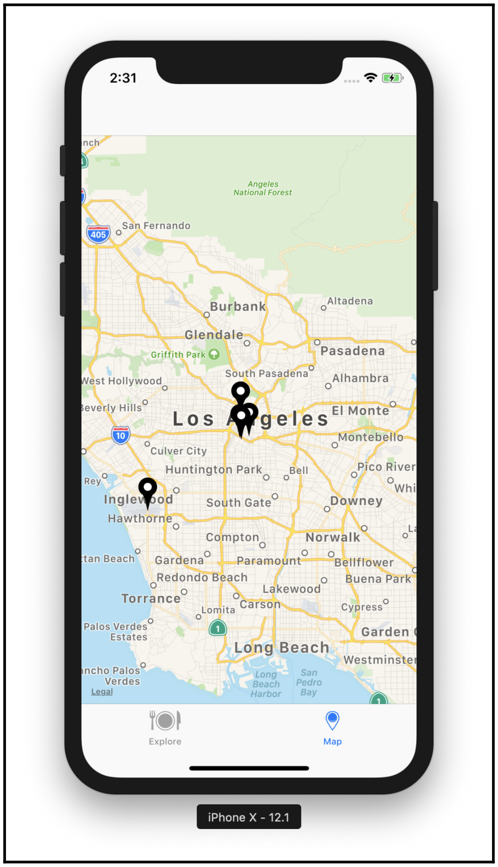


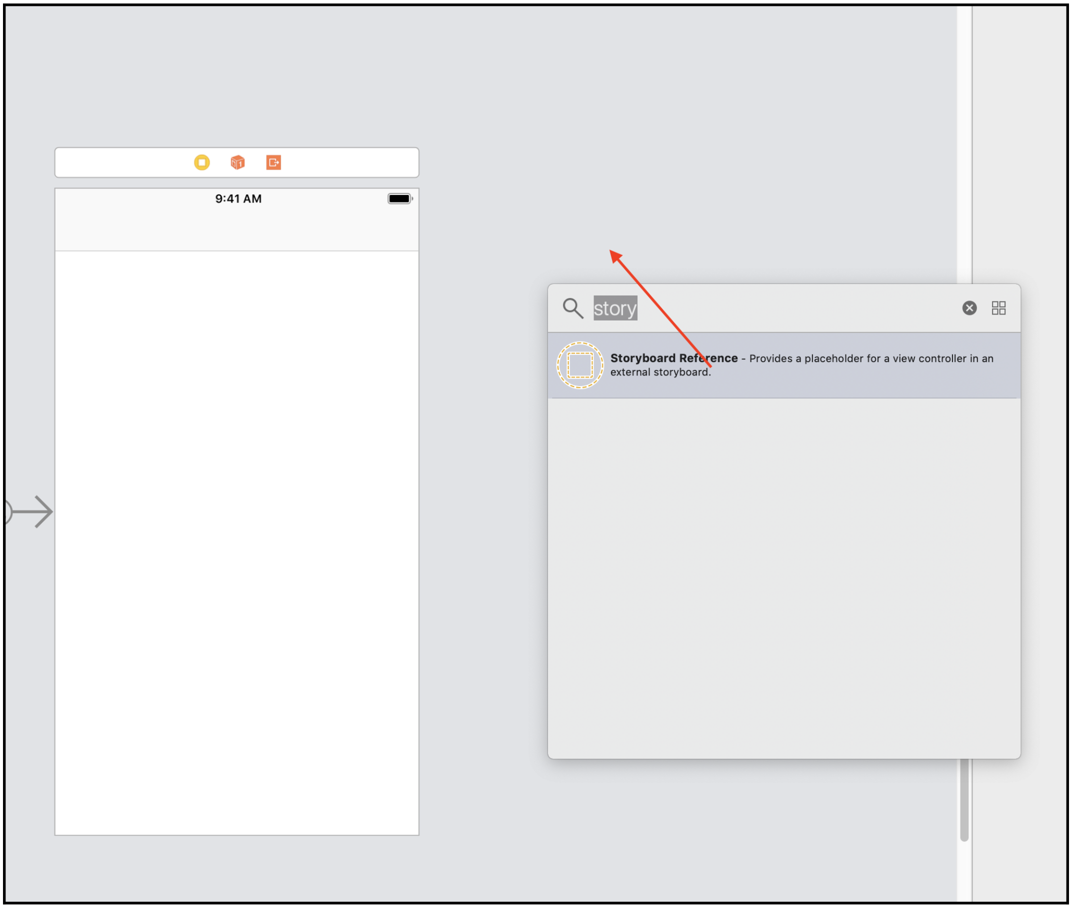
```
func currentRegion(latDelta:CLLocationDegrees, longDelta:CLLocationDegrees) -> MKCoordinateRegion {
    guard let item = items.first else { return MKCoordinateRegion() }
    let span = MKCoordinateSpanMake(latDelta, longDelta)

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


6:52
Burbank
Burbank Blvd
Vineland Ave
Glendale
Griffith Park
West Hollywood
W Sunset Blvd
Melrose Ave
Beverly Blvd
Beverly Hills
W Pico Blvd
Red Hill Restaurant
Maria's Italian Kitchen - Downtown
W Adams Blvd
W Jefferson Blvd
Culver City
W Vernon Ave
Vernon
Huntington Park
Inglewood
Chalet Edelweiss
Los Angeles International Airport (LAX)
South Gate
Hawthorne
S Figueroa St
Lawndale
Compton
Gardena
Hermosa Beach
Legal
Explore
Map
iPhone X - 12.1



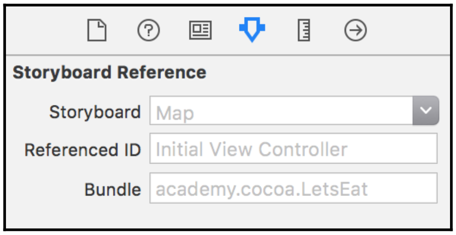
```swift
func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
    let identifier = "custompin"

    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
}
```


2:31
Angeles National Forest
San Fernando
405
Burbank
Altadena
Glendale
Pasadena
Griffith Park
South Pasadena
West Hollywood
Alhambra
Beverly Hills
Los Angeles
El Monte
10
Montebello
Culver City
Huntington Park
Bell
Pico River
Rey
Inglewood
South Gate
Downey
Hawthorne
Compton
Norwalk
tan Beach
Gardena
Paramount
Bellflower
Redondo Beach
Buena Park
Lakewood
Torrance
Carson
Lomita
Cypress
Palos Verdes Estates
1
Garden
Long Beach
Westminster
ncho Palos Verdes
Long Beach Harbor
San Pedro Bay
Legal
Explore
Map
iPhone X - 12.1



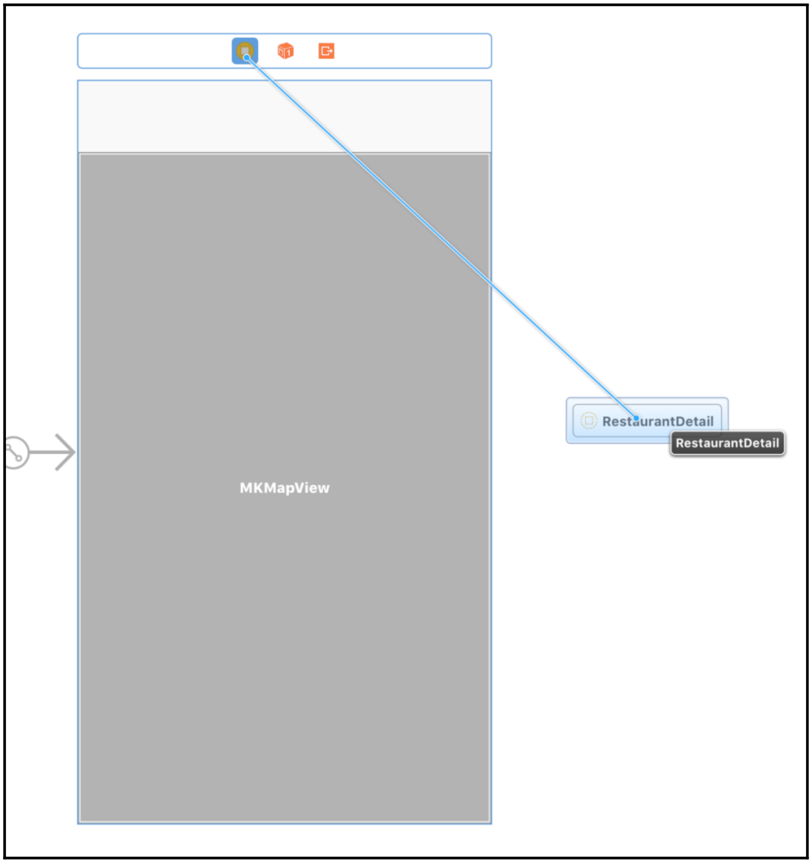
story

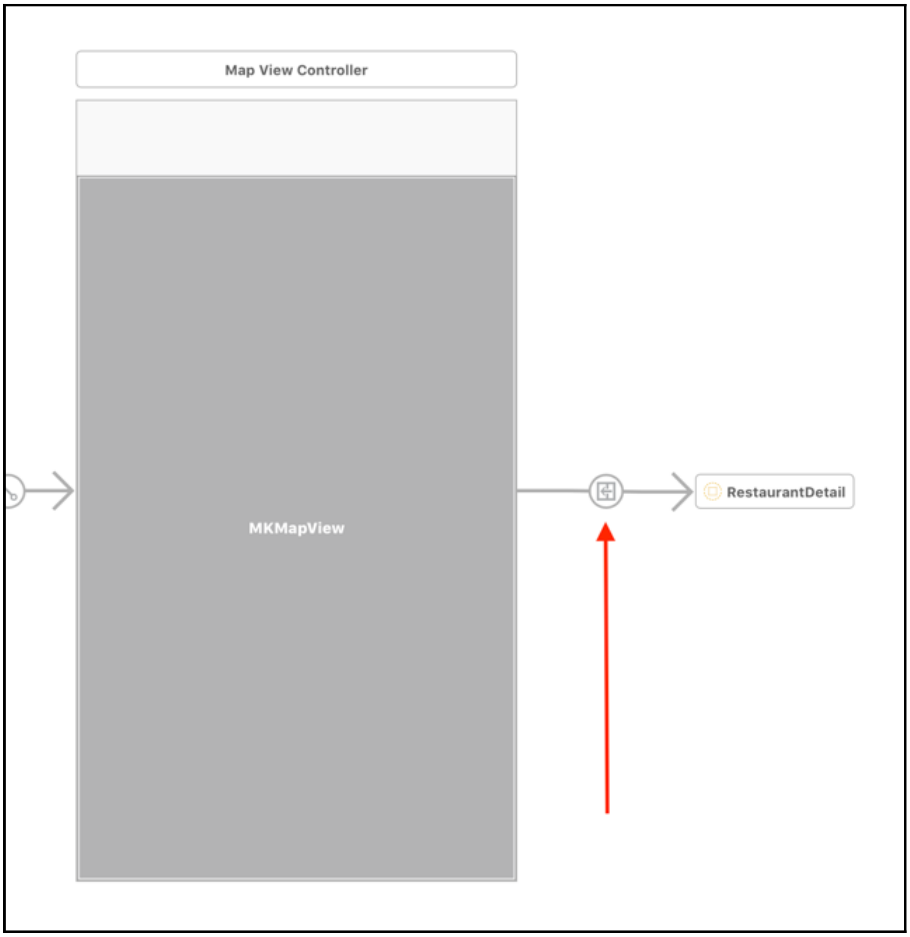
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

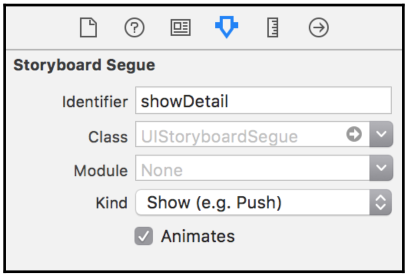


Storyboard Reference

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


MKMapView
RestaurantDetail
RestaurantDetail




Storyboard Segue

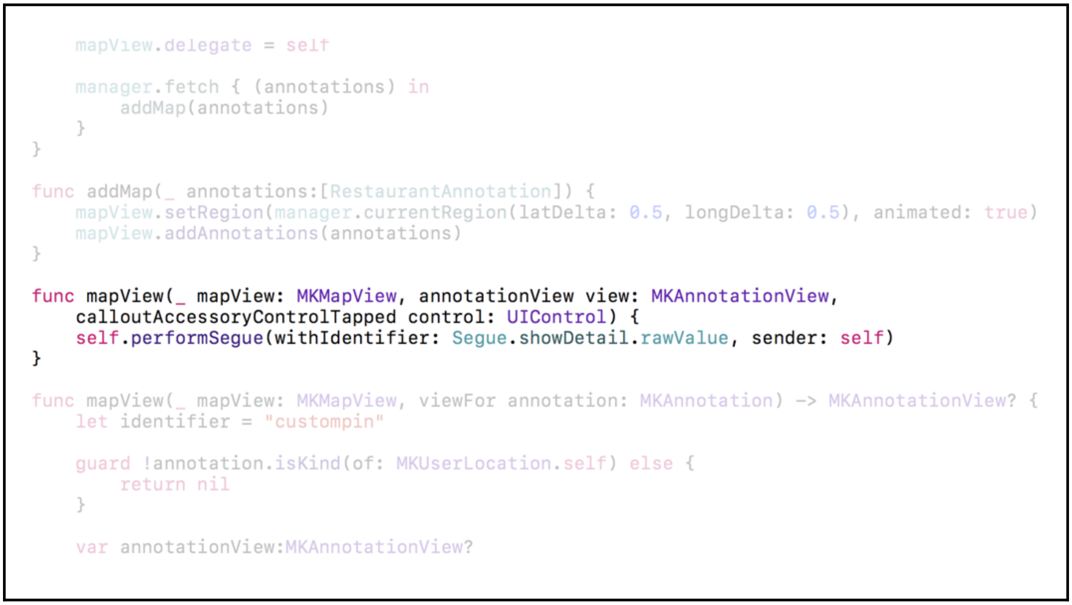
Identifier showDetail

Class UIStoryboardSegue

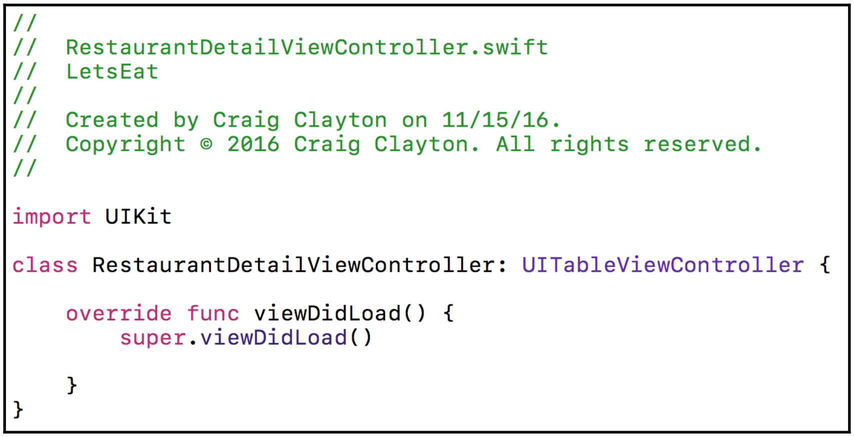
Module None

Kind Show (e.g. Push)

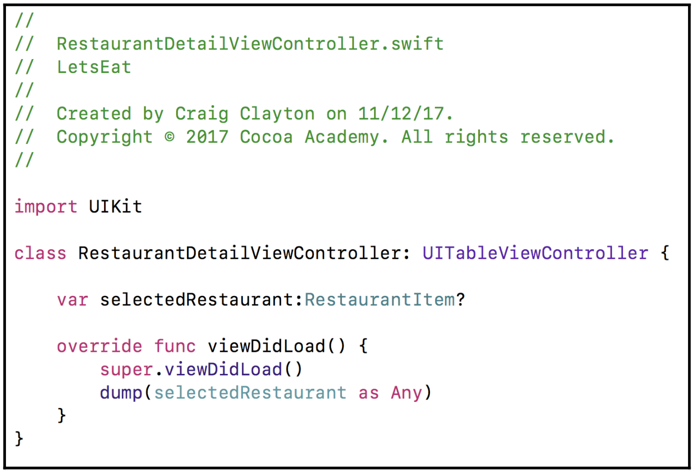
☑ Animates



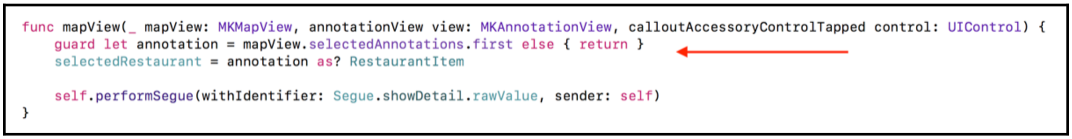
```
    mapView.delegate = self

    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?
```



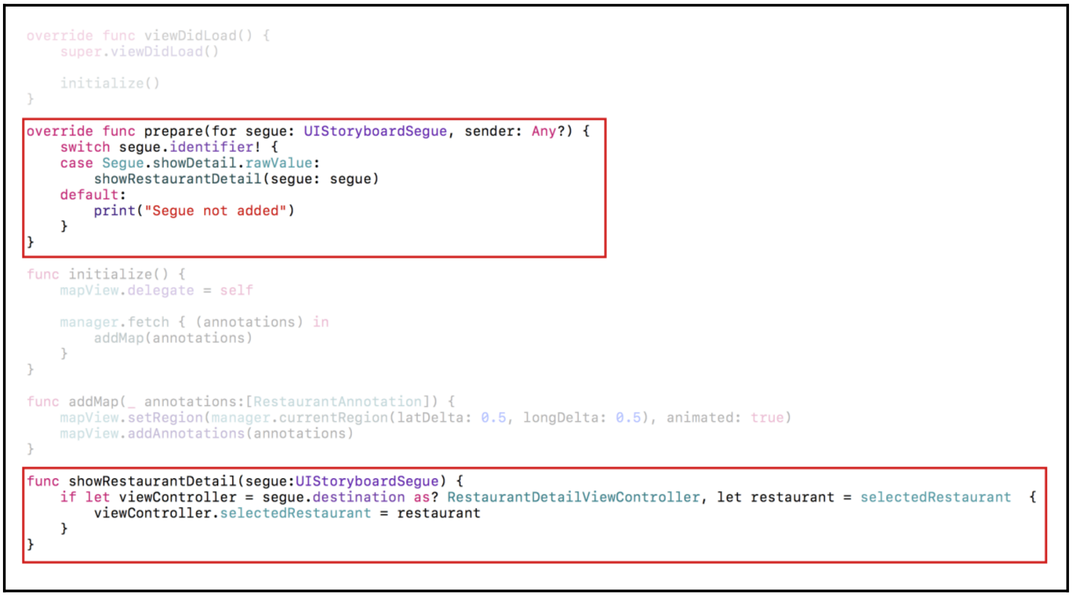
```
//
//  RestaurantDetailViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 11/15/16.
//  Copyright © 2016 Craig Clayton. All rights reserved.
//

import UIKit

class RestaurantDetailViewController: UITableViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

    }
}
```



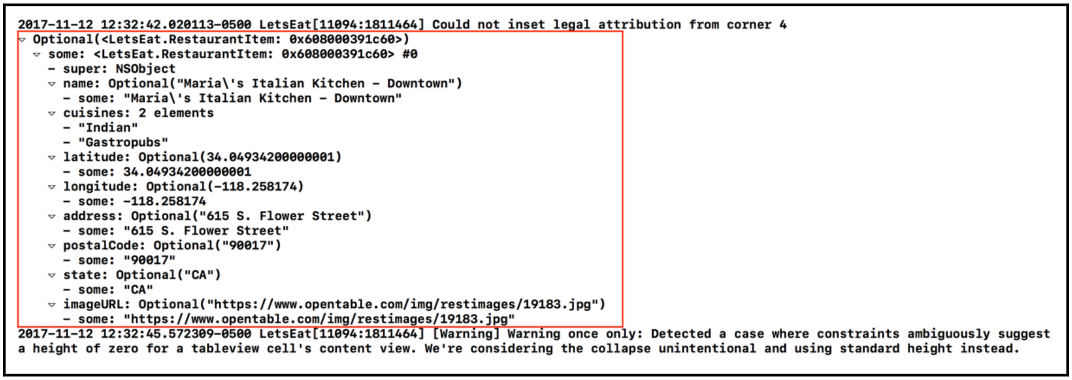
```
//
//  RestaurantDetailViewController.swift
//  LetsEat
//
//  Created by Craig Clayton on 11/12/17.
//  Copyright © 2017 Cocoa Academy. All rights reserved.
//

import UIKit

class RestaurantDetailViewController: UITableViewController {

    var selectedRestaurant:RestaurantItem?

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



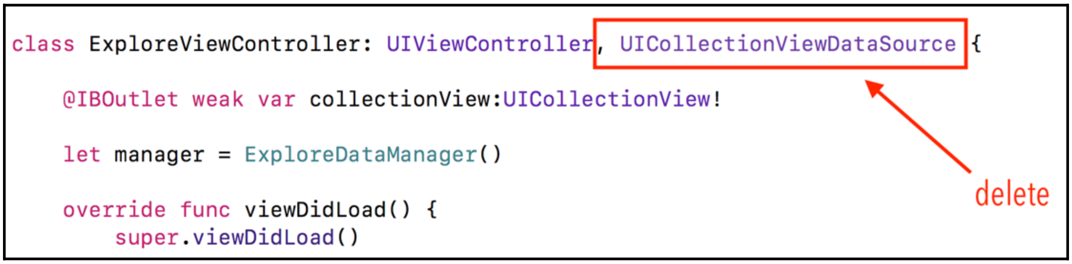
```
func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
    guard let annotation = mapView.selectedAnnotations.first else { return }
    selectedRestaurant = annotation as? RestaurantItem

    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
    }
}
```

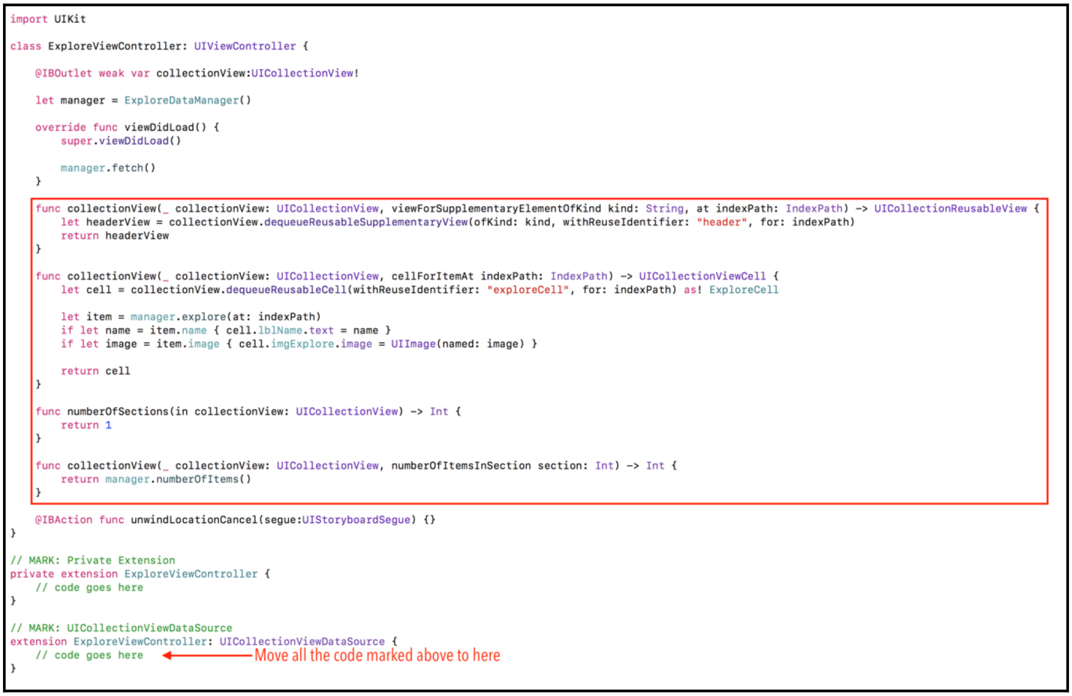


```
2017-11-12 12:32:42.020113-0500 LetsEat[11094:1811464] Could not inset legal attribution from corner 4
▿ Optional(<LetsEat.RestaurantItem: 0x608000391c60>)
  ▿ some: <LetsEat.RestaurantItem: 0x608000391c60> #0
    - super: NSObject
    ▿ name: Optional("Maria\'s Italian Kitchen - Downtown")
      - 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.
```



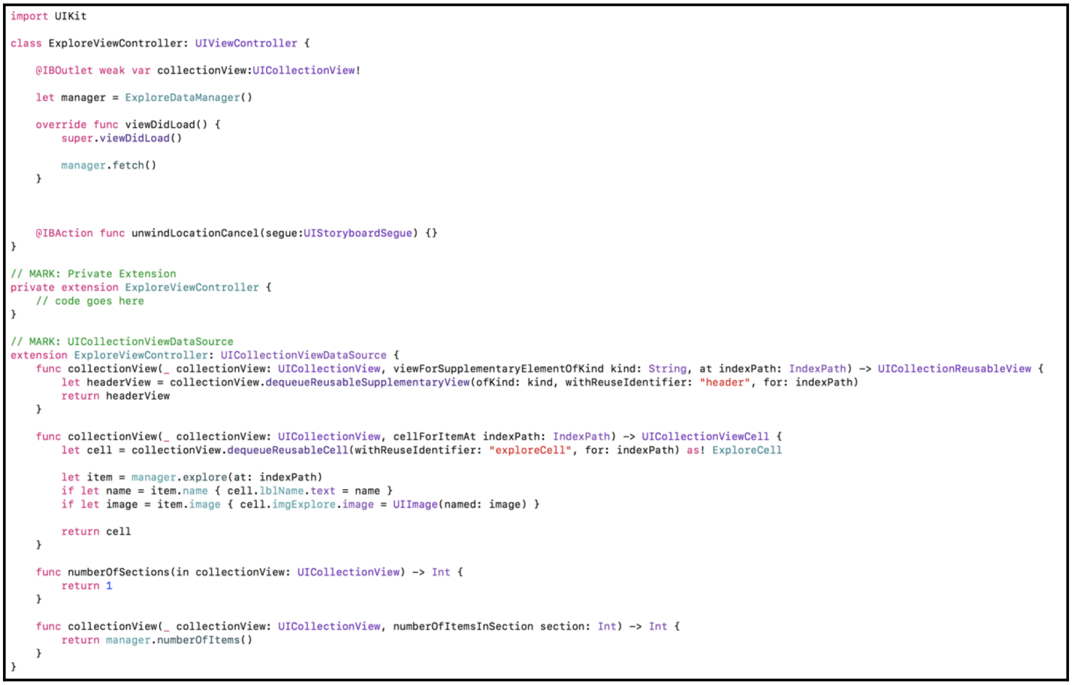
```
class ExploreViewController: UIViewController, UICollectionViewDataSource {

    @IBOutlet weak var collectionView:UICollectionView!

    let manager = ExploreDataManager()

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

delete

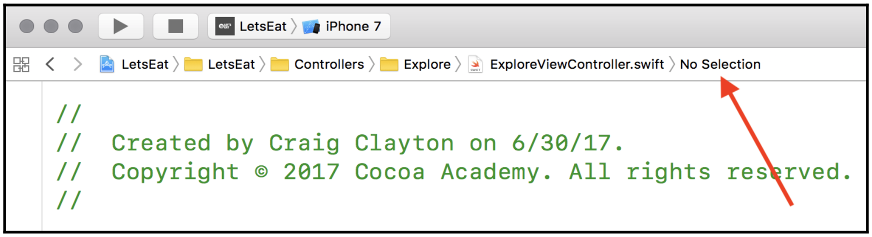


```
import UIKit

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
}
```

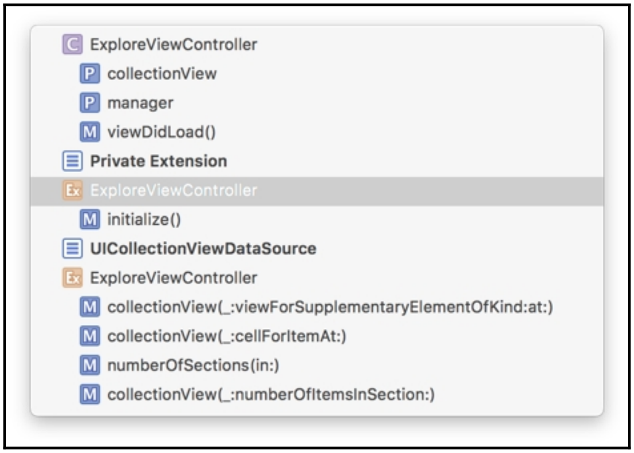
Move all the code marked above to here



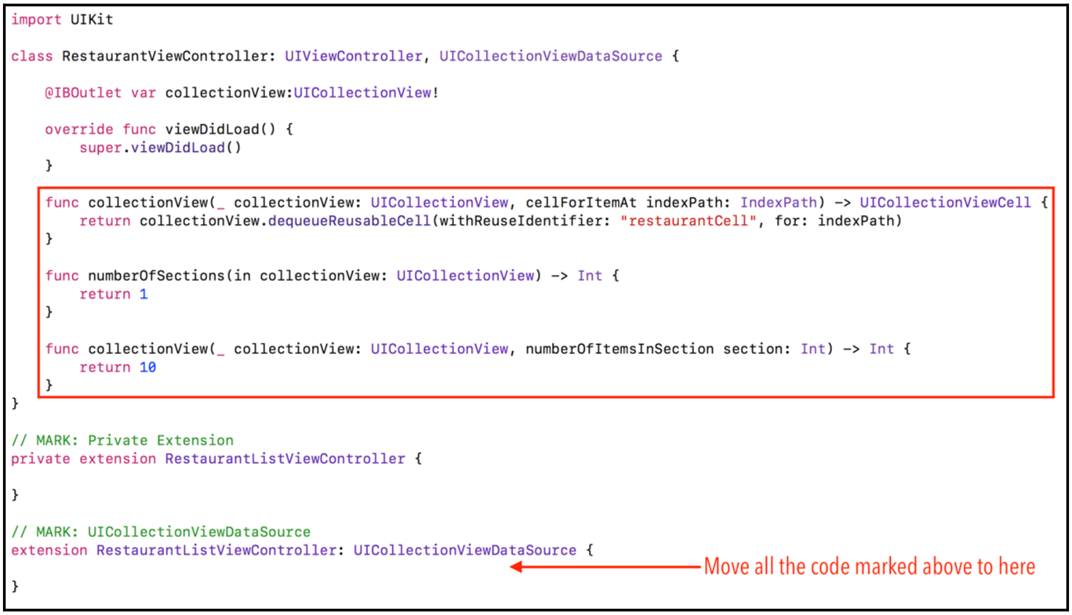
```
import UIKit

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()
    }
}
```



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

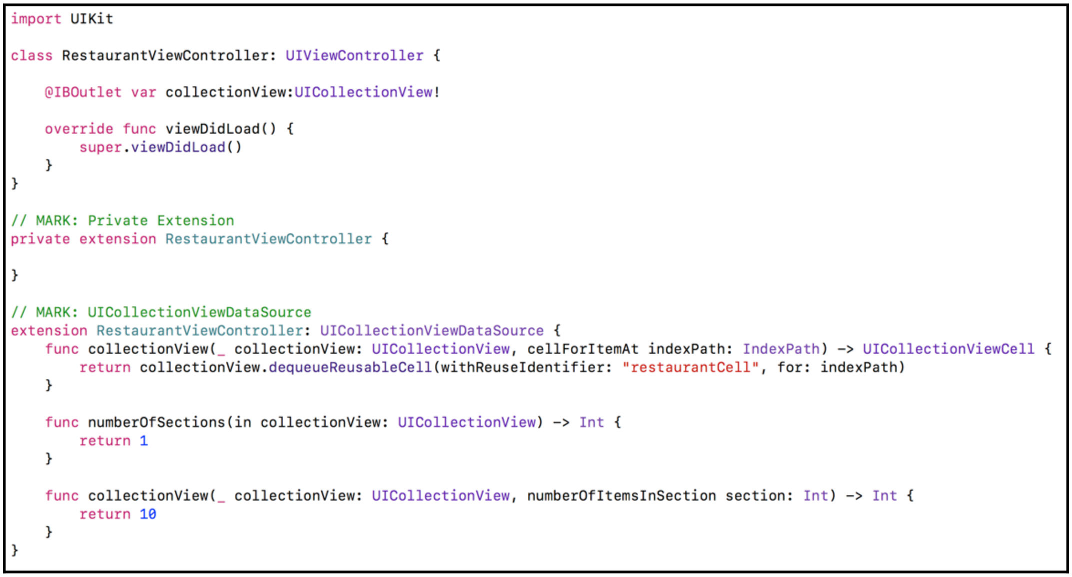


- C ExploreViewController
    - P collectionView
    - P manager
    - M viewDidLoad()
- **Private Extension**
- Ex ExploreViewController
    - M initialize()
- **UICollectionViewDataSource**
- Ex ExploreViewController
    - M collectionView(_:viewForSupplementaryElementOfKind:at:)
    - M collectionView(_:cellForItemAt:)
    - M numberOfSections(in:)
    - M collectionView(_:numberOfItemsInSection:)

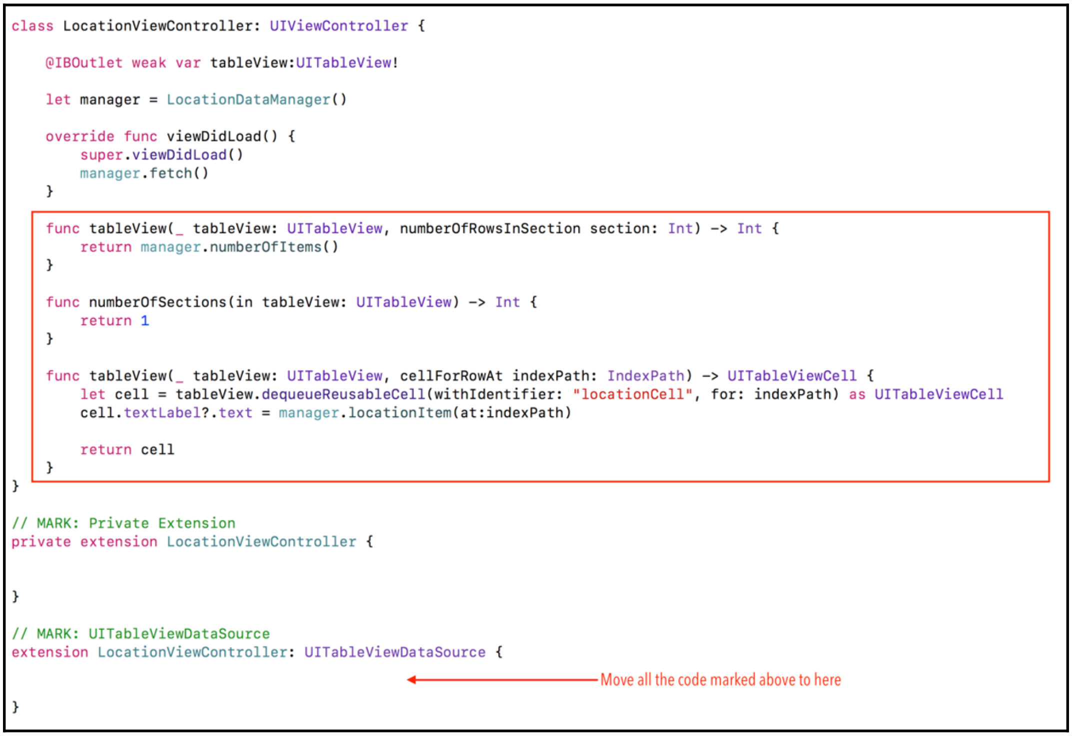


```
import UIKit

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

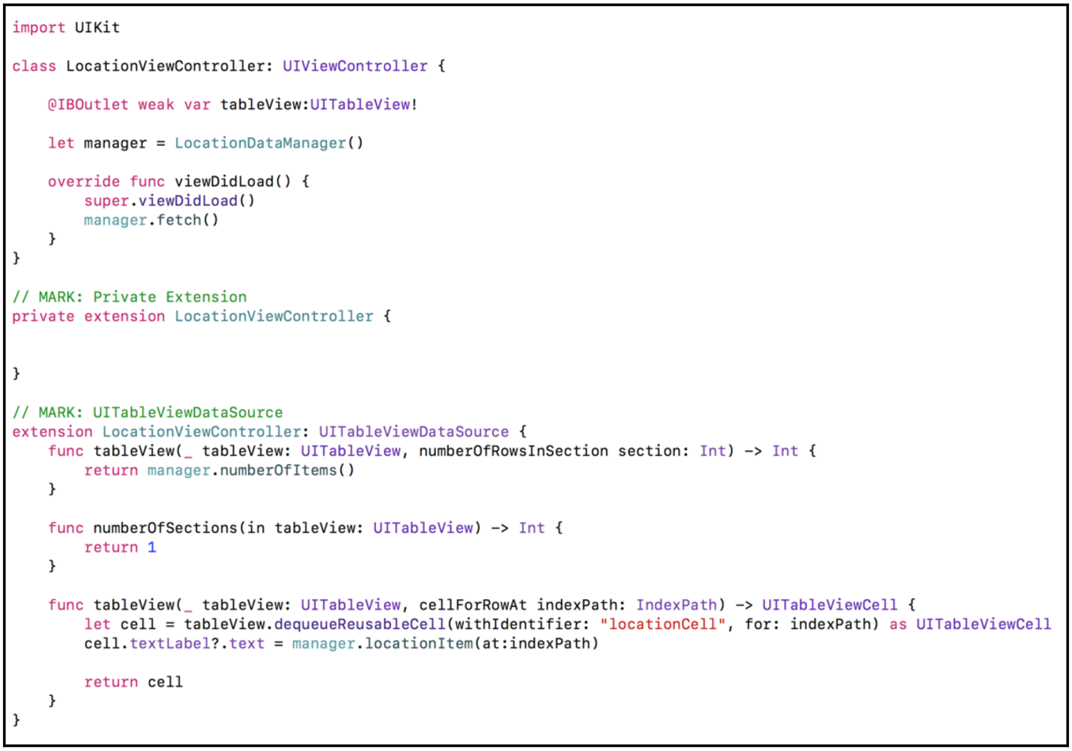


```
import UIKit

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
    }
}
```

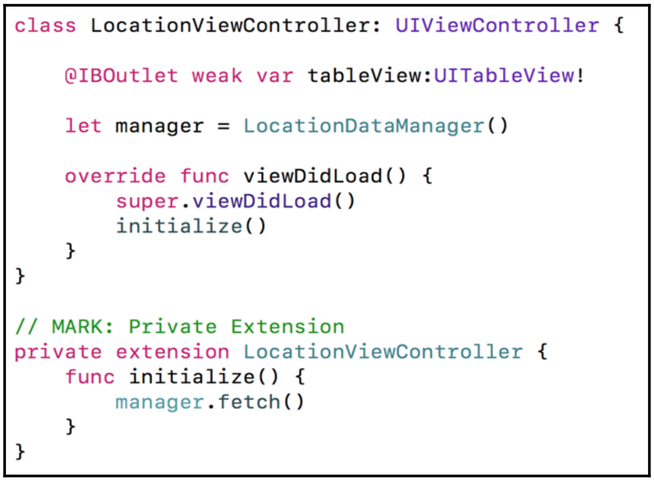


```
class LocationViewController: UIViewController {

    @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 {

}
```

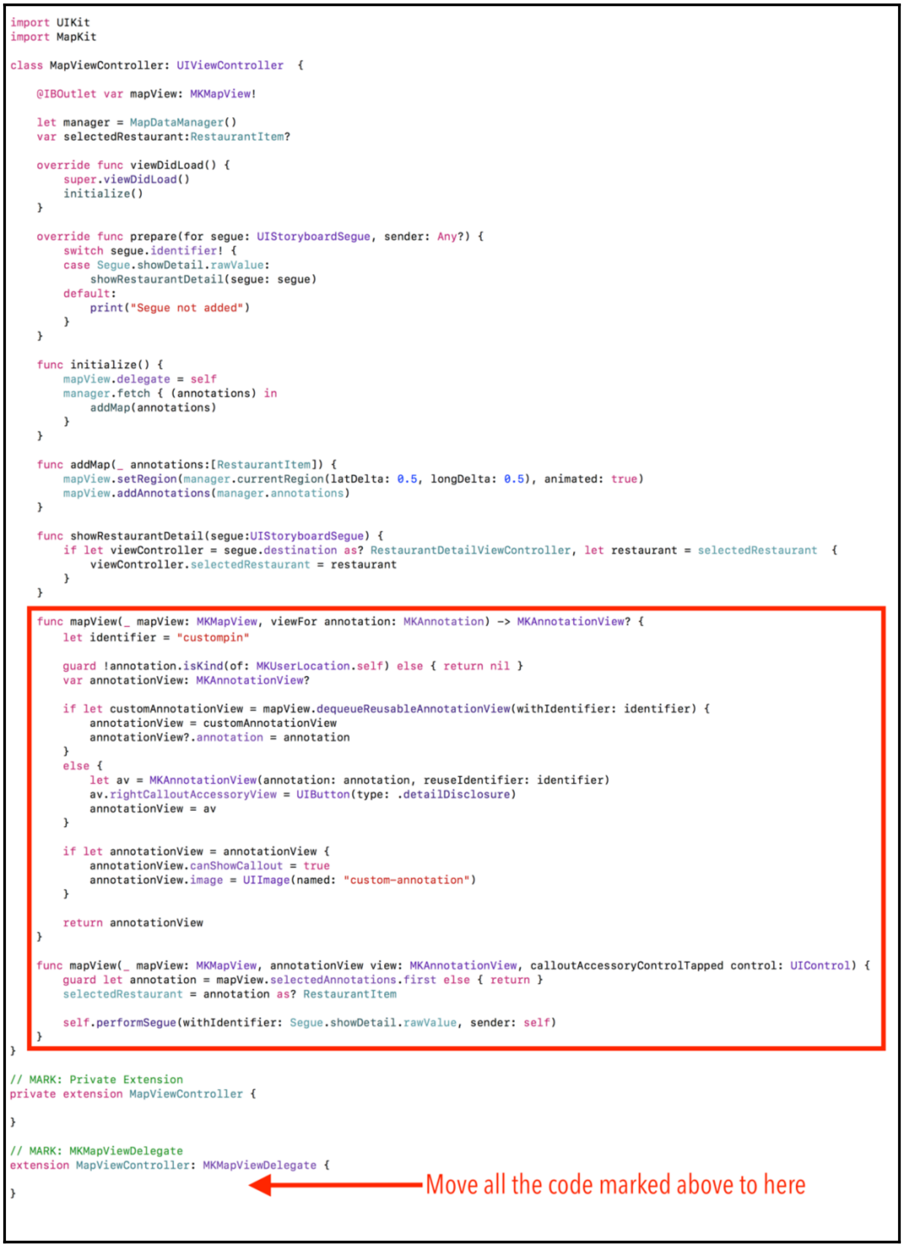
Move all the code marked above to here



```swift
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
    }
}
```

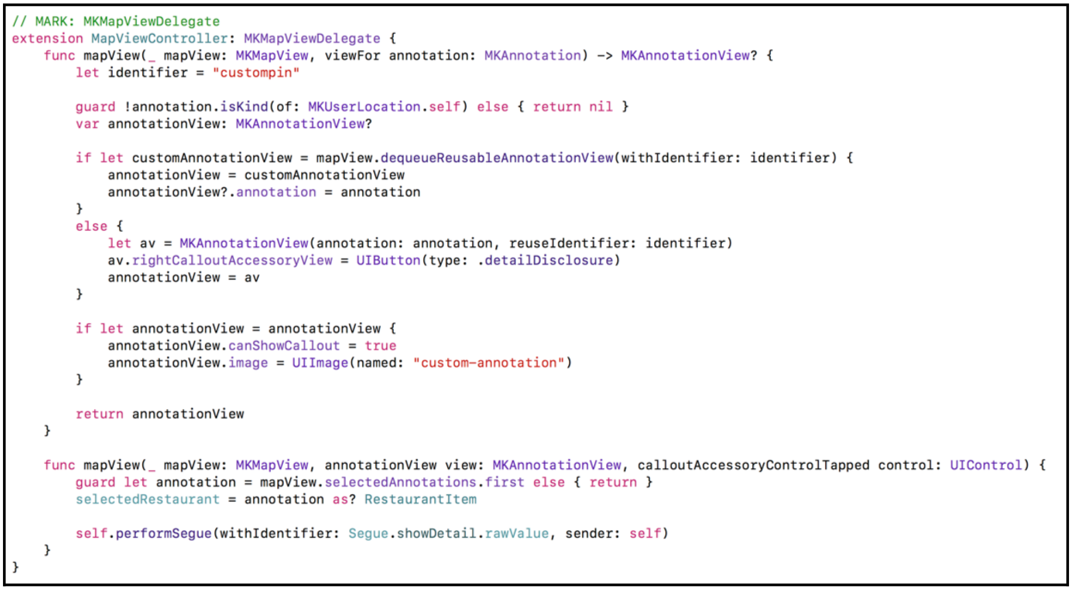


```swift
class LocationViewController: UIViewController {

    @IBOutlet weak var tableView:UITableView!

    let manager = LocationDataManager()

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

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

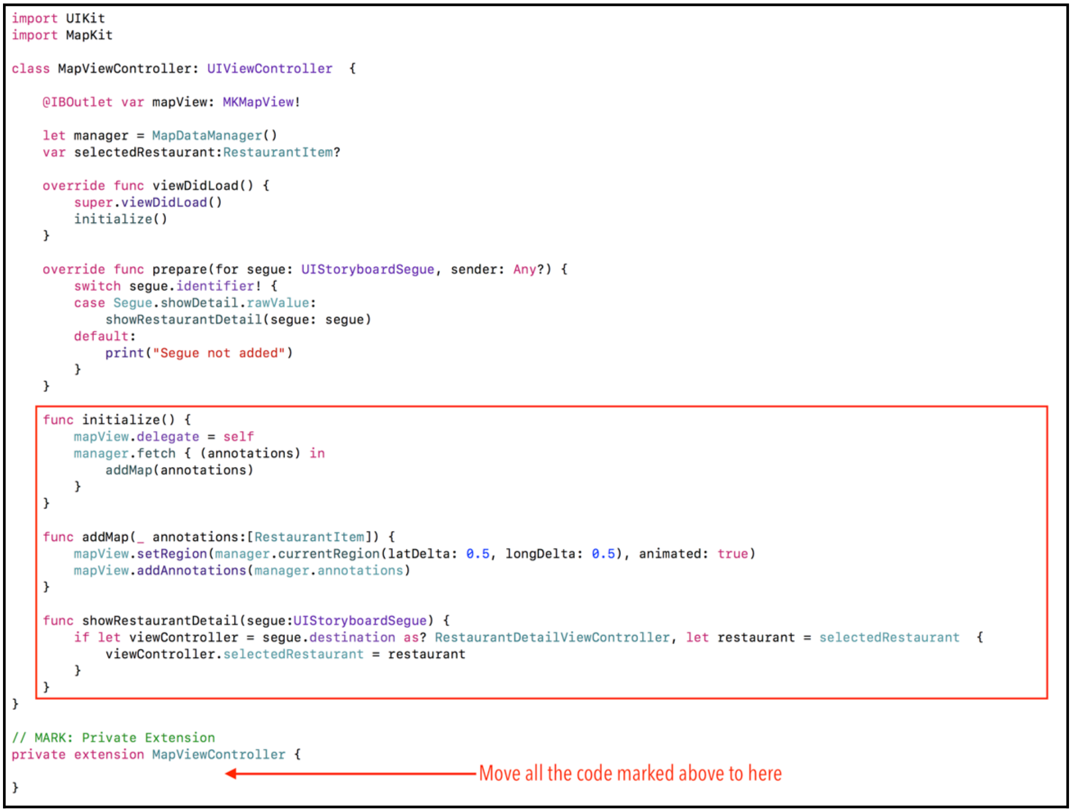


```swift
import UIKit
import MapKit

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 {

}
```

Move all the code marked above to here

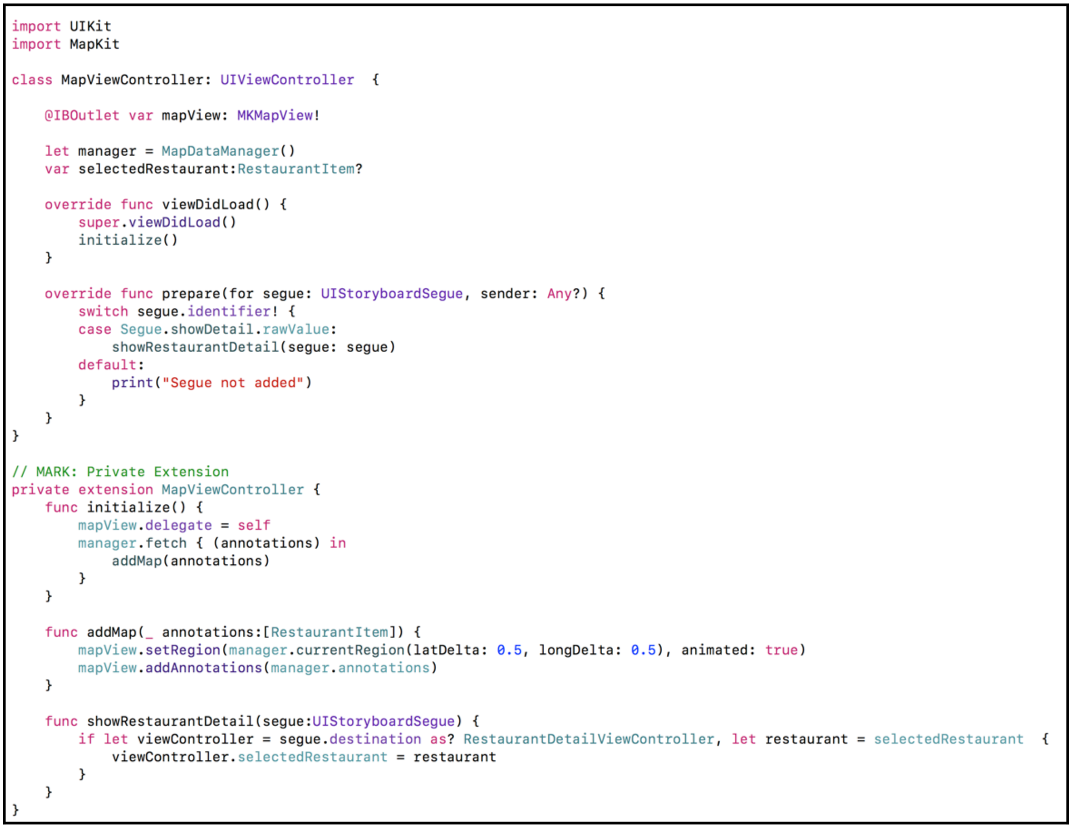


```
// MARK: MKMapViewDelegate
extension MapViewController: MKMapViewDelegate {
    func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
        let identifier = "custompin"

        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)
    }
}
```



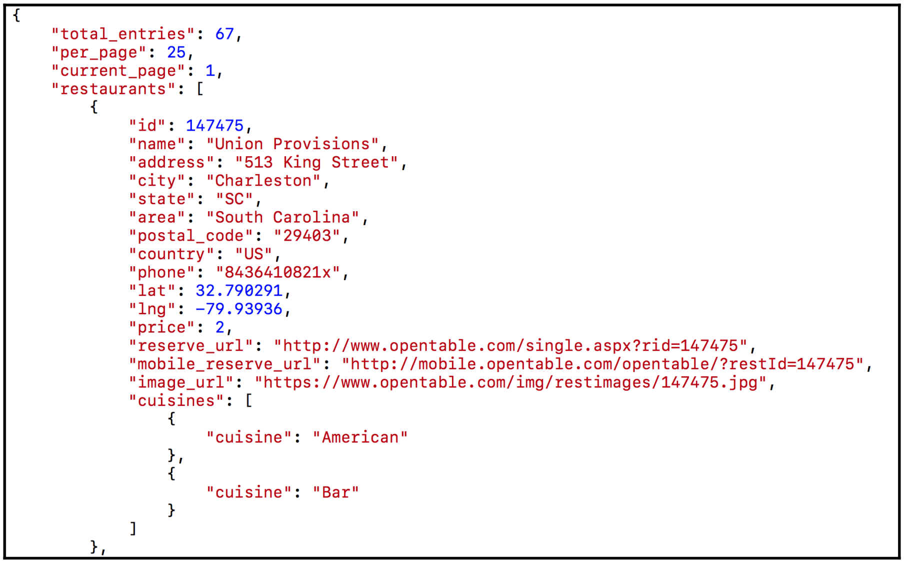
```swift
import UIKit
import MapKit

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
        }
    }
}

// MARK: Private Extension
private extension MapViewController {

}
```

Move all the code marked above to here

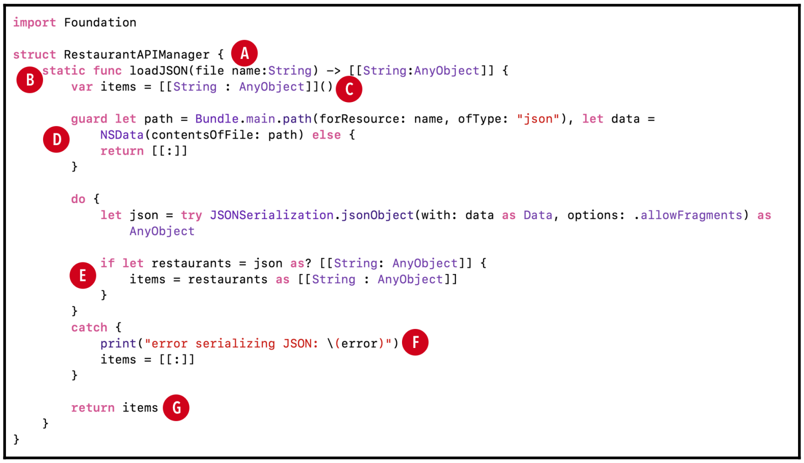


```
import UIKit
import MapKit

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
        }
    }
}
```

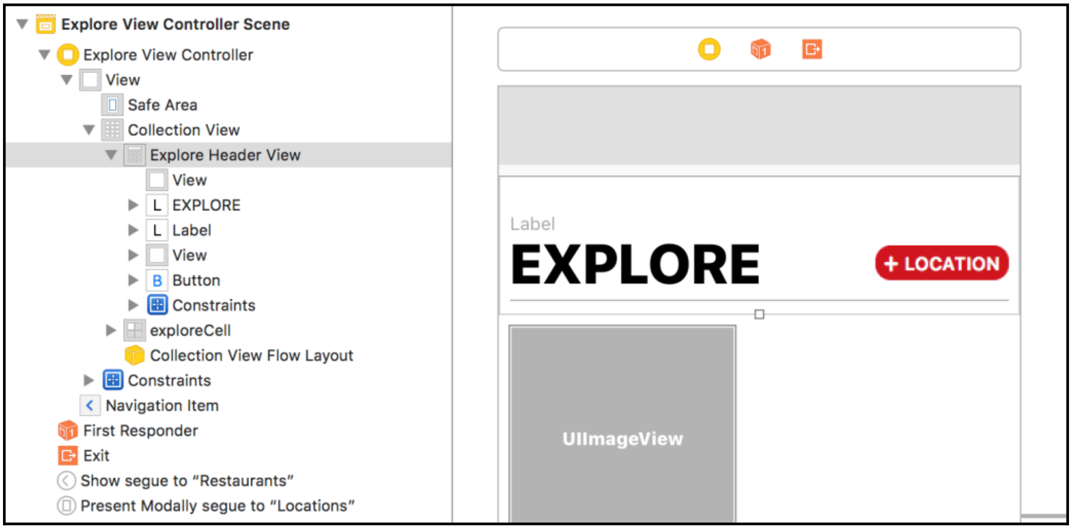
Chapter15: Working with an API

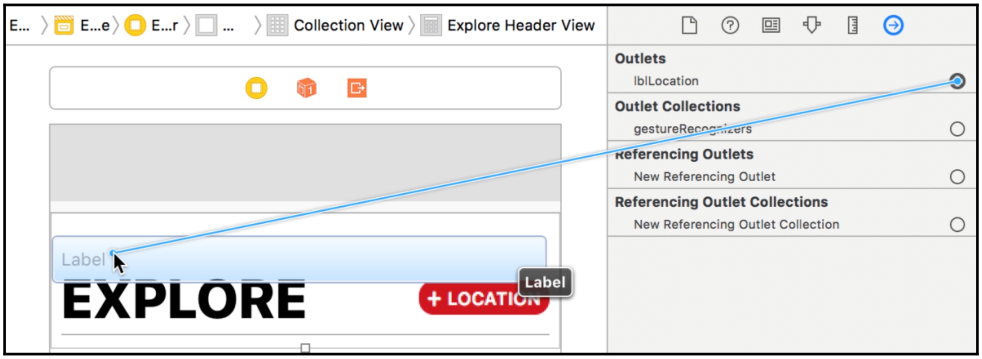


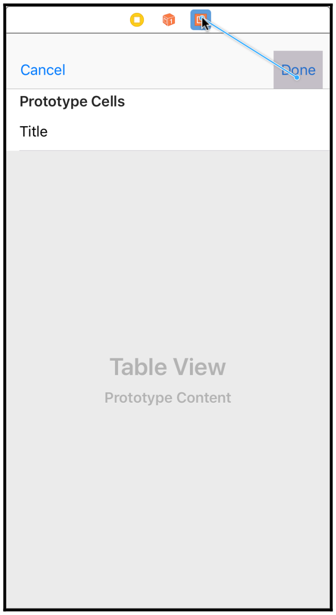
```
{
    "total_entries": 67,
    "per_page": 25,
    "current_page": 1,
    "restaurants": [
        {
            "id": 147475,
            "name": "Union Provisions",
            "address": "513 King Street",
            "city": "Charleston",
            "state": "SC",
            "area": "South 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"
                }
            ]
        },
```

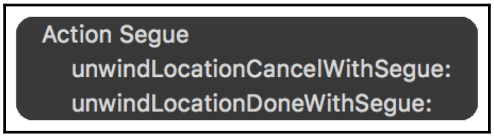


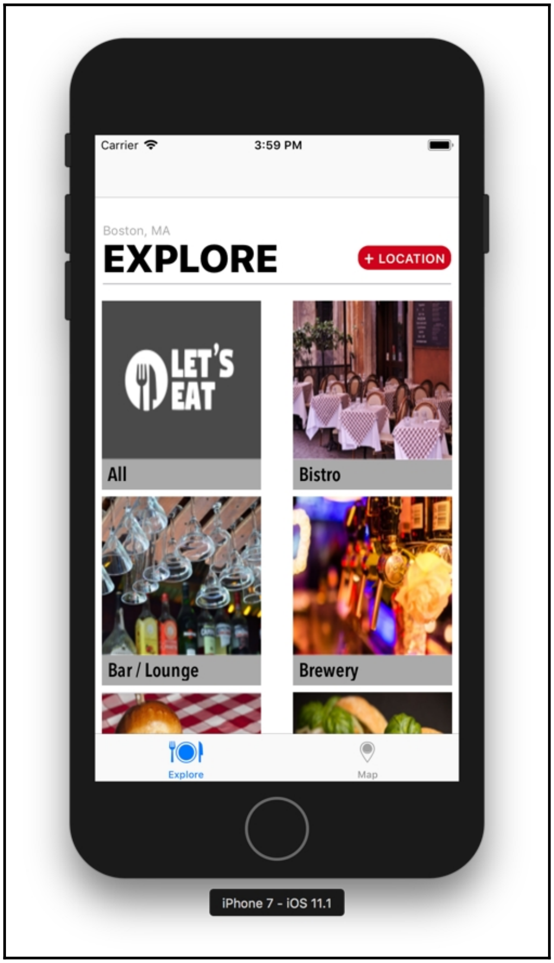
```
import Foundation

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 as? [[String: AnyObject]] {
                items = restaurants as [[String : AnyObject]]
            }
        }
        catch {
            print("error serializing JSON: \(error)")
            items = [[:]]
        }

        return items
    }
}
```

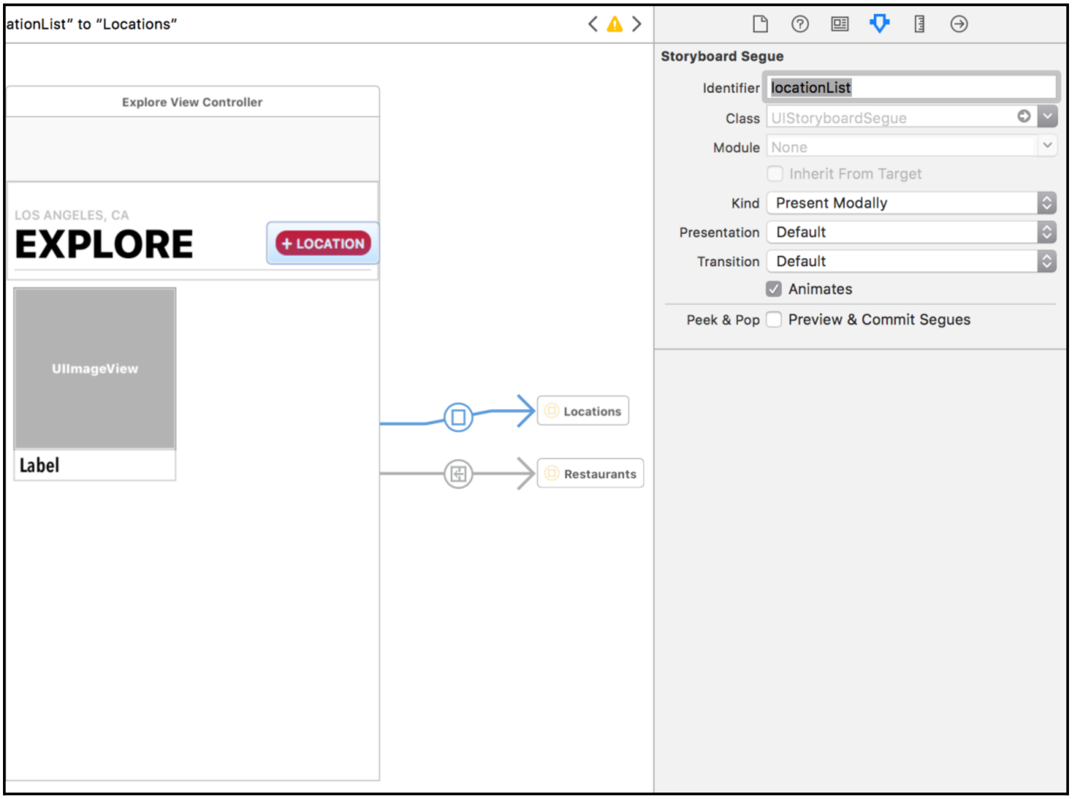

Explore View Controller Scene
Explore View Controller
View
Safe Area
Collection View
Explore Header View
View
EXPLORE
Label
View
Button
Constraints
exploreCell
Collection View Flow Layout
Constraints
Navigation Item
First Responder
Exit
Show segue to "Restaurants"
Present Modally segue to "Locations"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


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



Storyboard Segue

Identifier locationList

Class UIStoryboardSegue

Module None

□ Inherit From Target

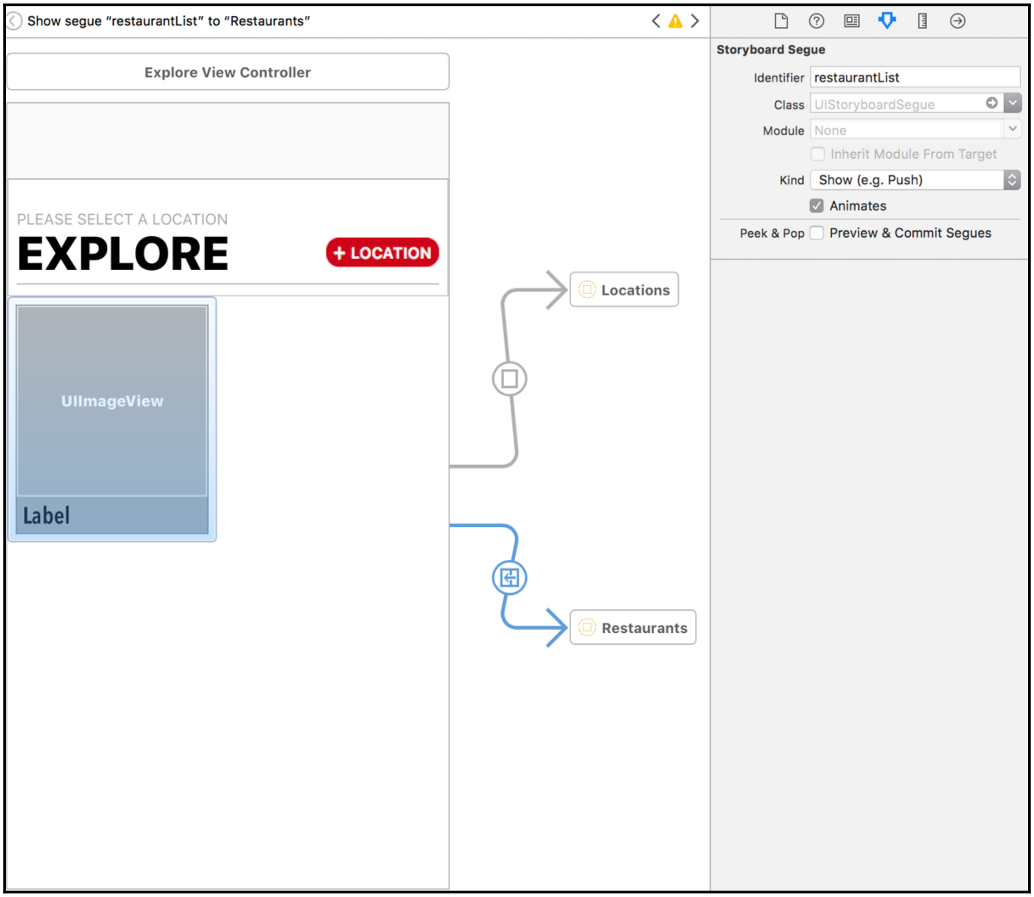
Kind Present Modally

Presentation Default

Transition Default

☑ Animates

Peek & Pop □ Preview & Commit Segues



Storyboard Segue

Identifier restaurantList

Class UIStoryboardSegue

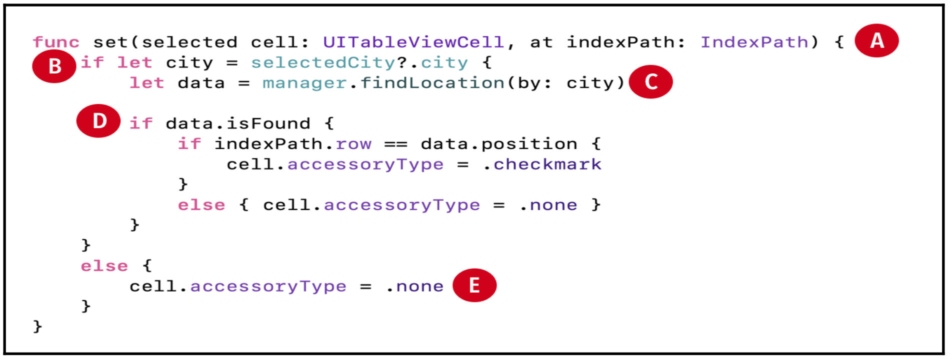
Module None

☐ Inherit Module From Target

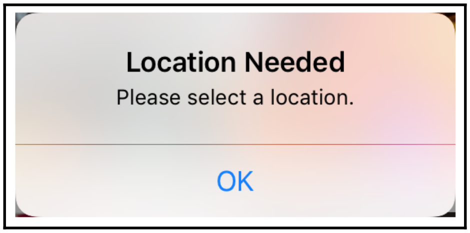
Kind Show (e.g. Push)

☑ Animates

Peek & Pop ☐ Preview & Commit Segues

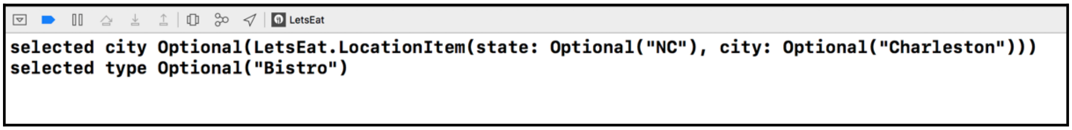


```
func set(selected cell: UITableViewCell, at indexPath: IndexPath) {
    if let city = selectedCity?.city {
        let data = manager.findLocation(by: city)

        if data.isFound {
            if indexPath.row == data.position {
                cell.accessoryType = .checkmark
            }
            else { cell.accessoryType = .none }
        }
    }
    else {
        cell.accessoryType = .none
    }
}
```

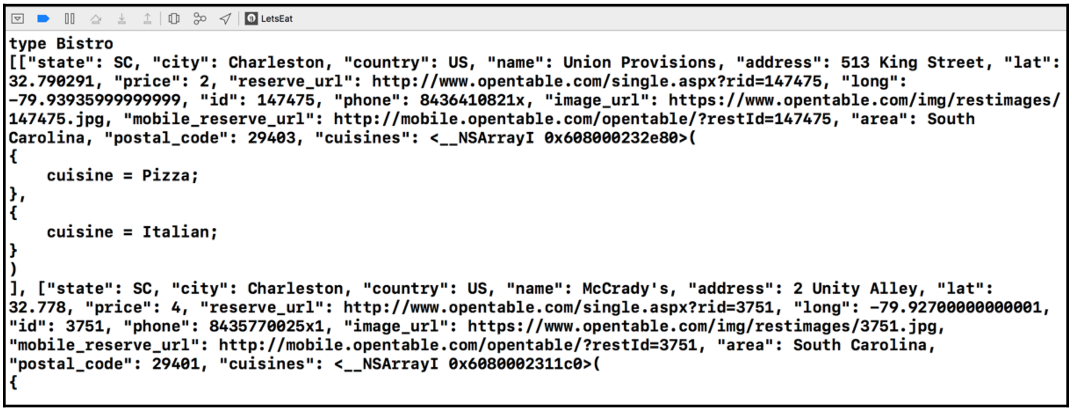


Location Needed

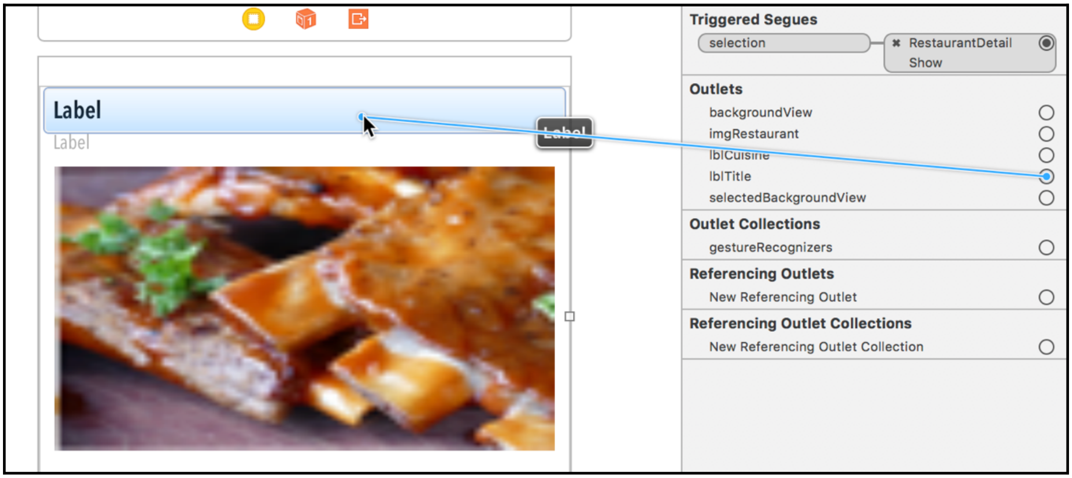
Please select a location.

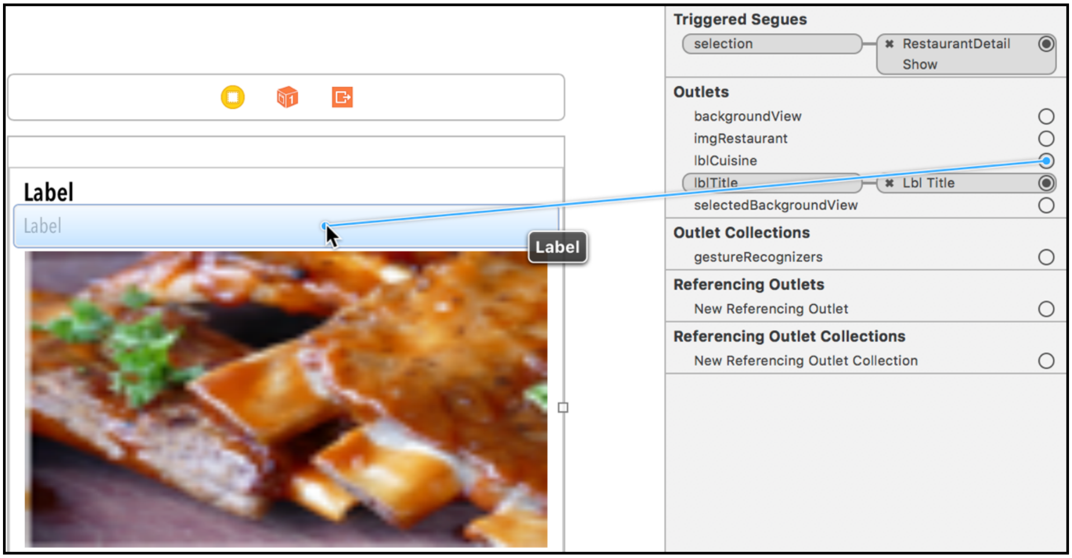


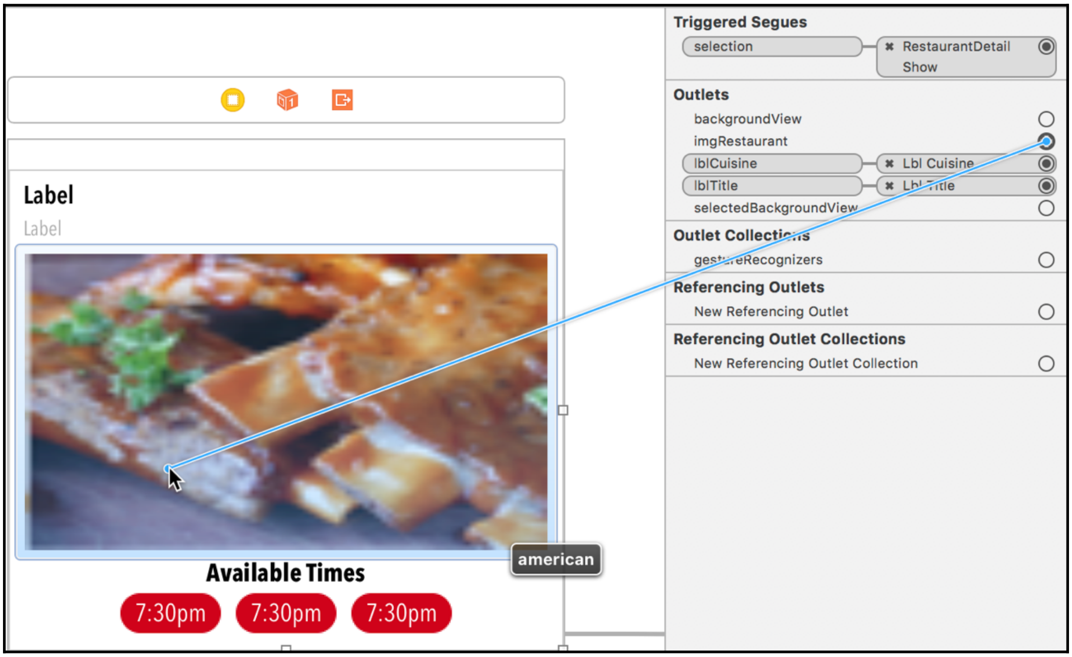
```
selected city Optional(LetsEat.LocationItem(state: Optional("NC"), city: Optional("Charleston")))
selected type Optional("Bistro")
```

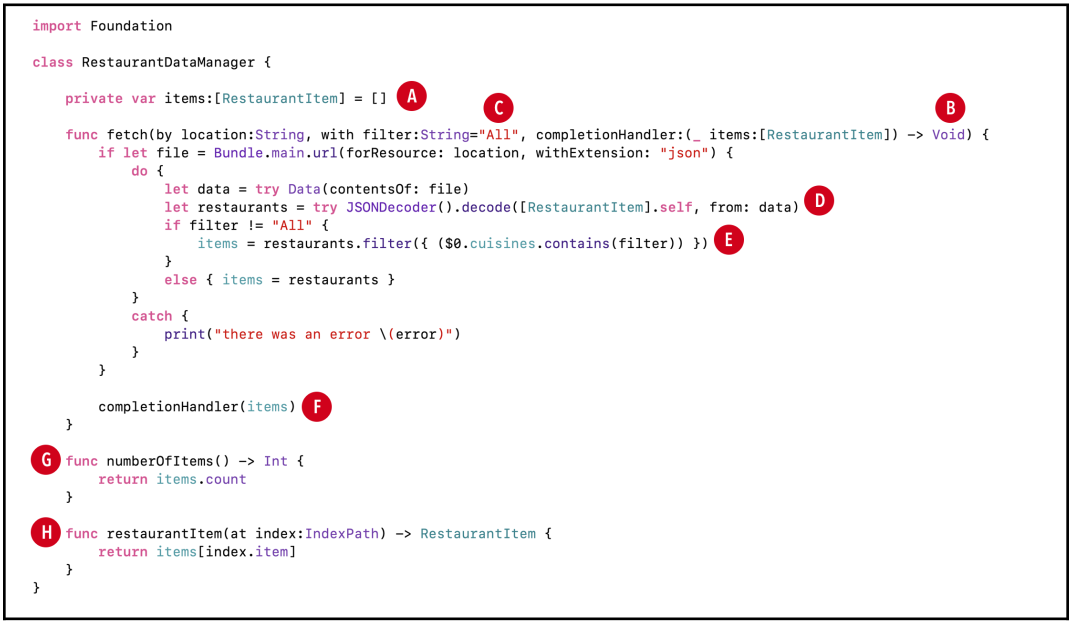


```
type Bistro
[["state": SC, "city": Charleston, "country": US, "name": Union Provisions, "address": 513 King Street, "lat":
32.790291, "price": 2, "reserve_url": http://www.opentable.com/single.aspx?rid=147475, "long":
-79.93935999999999, "id": 147475, "phone": 8436410821x, "image_url": https://www.opentable.com/img/restimages/
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>(
{
```

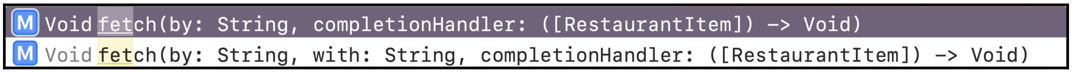



Label
Label
Label
Triggered Segues
selection
✖ RestaurantDetail
Show
Outlets
backgroundView
imgRestaurant
lblCuisine
lblTitle
✖ Lbl Title
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

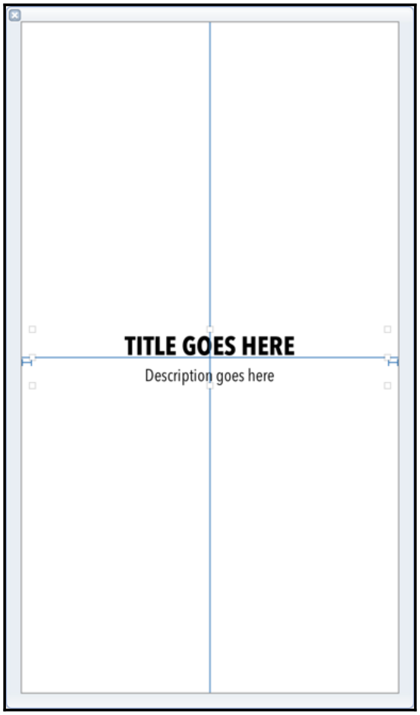

Label
Label
american
Available Times
7:30pm
7:30pm
7:30pm
Triggered Segues
selection
✖ RestaurantDetail
Show
Outlets
backgroundView
imgRestaurant
lblCuisine
✖ Lbl Cuisine
lblTitle
✖ Lbl Title
selectedBackgroundView
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing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



```
import Foundation

class RestaurantDataManager {

    private var items:[RestaurantItem] = []

    func fetch(by location:String, with filter:String="All", completionHandler:(_ items:[RestaurantItem]) -> Void) {
        if let file = Bundle.main.url(forResource: location, withExtension: "json") {
            do {
                let data = try Data(contentsOf: file)
                let restaurants = try JSONDecoder().decode([RestaurantItem].self, from: data)
                if filter != "All" {
                    items = restaurants.filter({ ($0.cuisines.contains(filter)) })
                }
                else { items = restaurants }
            }
            catch {
                print("there was an error \(error)")
            }
        }

        completionHandler(items)
    }

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

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

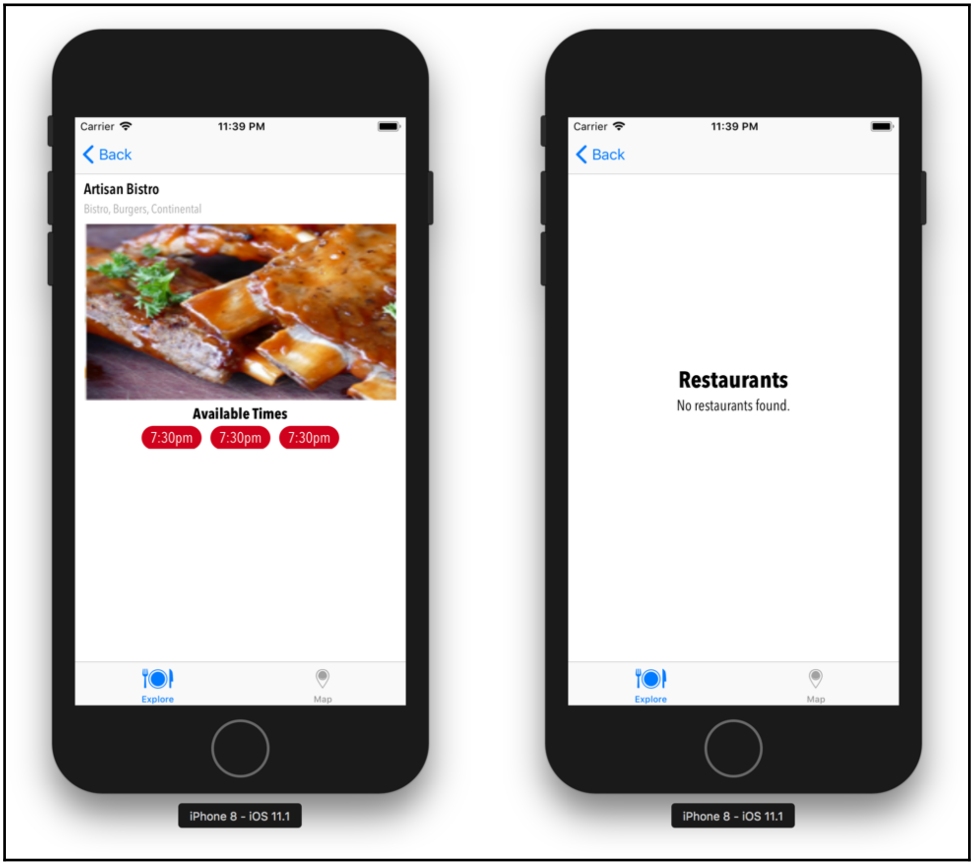


```
M Void fetch(by: String, completionHandler: ([RestaurantItem]) -> Void)
M Void fetch(by: String, with: String, completionHandler: ([RestaurantItem]) -> Void)
```



TITLE GOES HERE

Description goes here



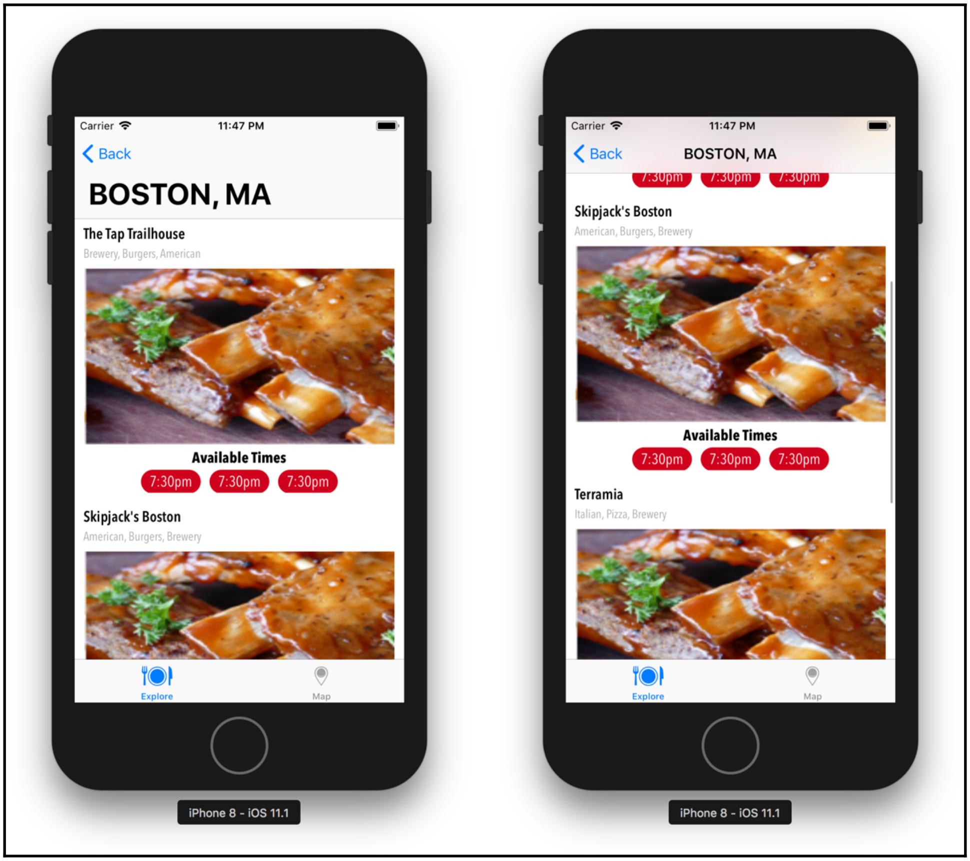
Artisan Bistro

Bistro, Burgers, Continental

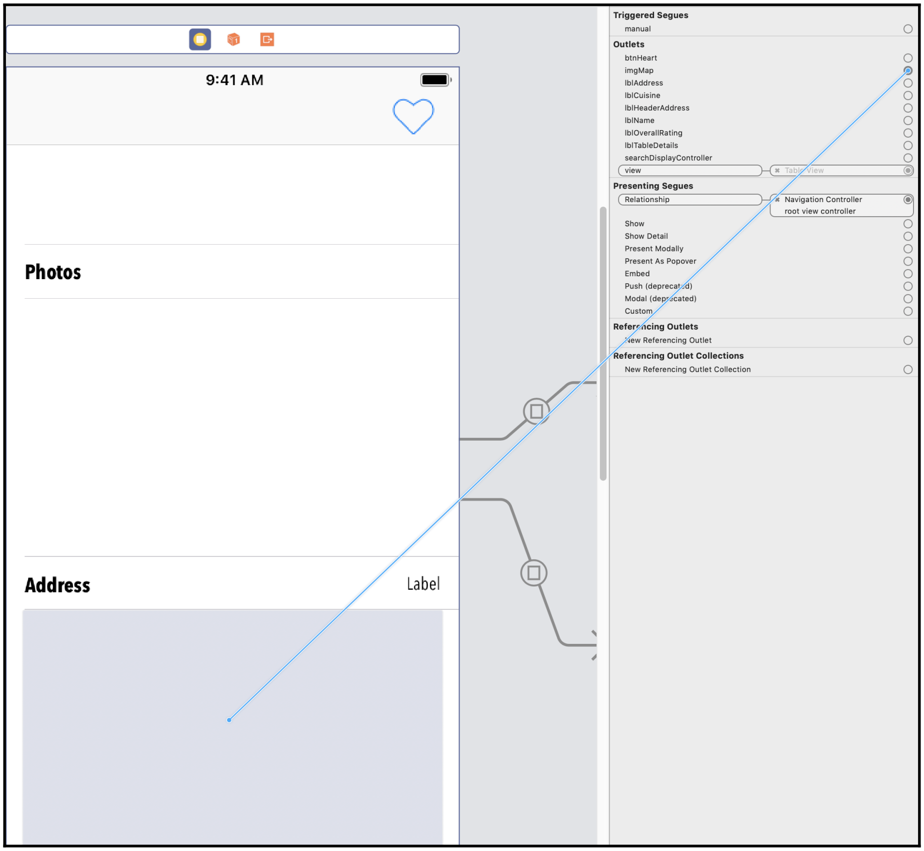
Available Times

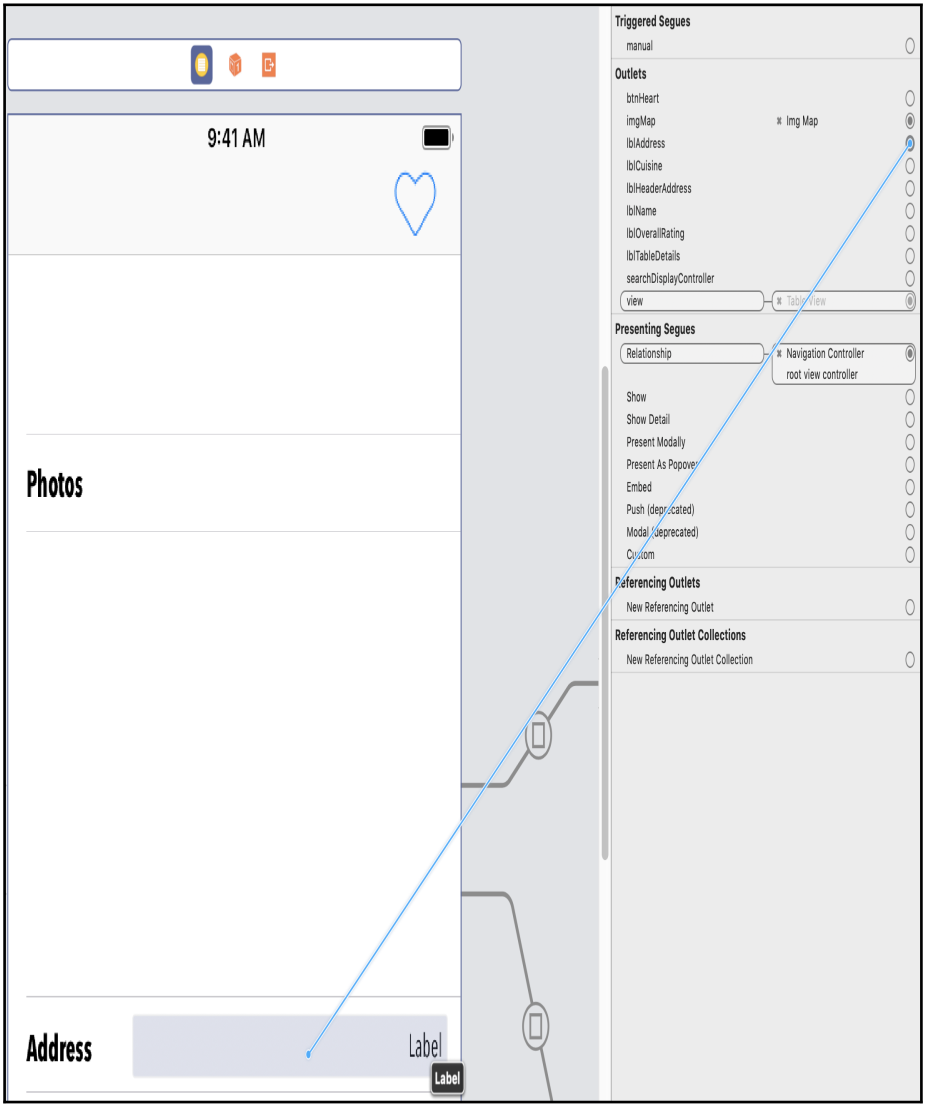
Restaurants

No restaurants found.

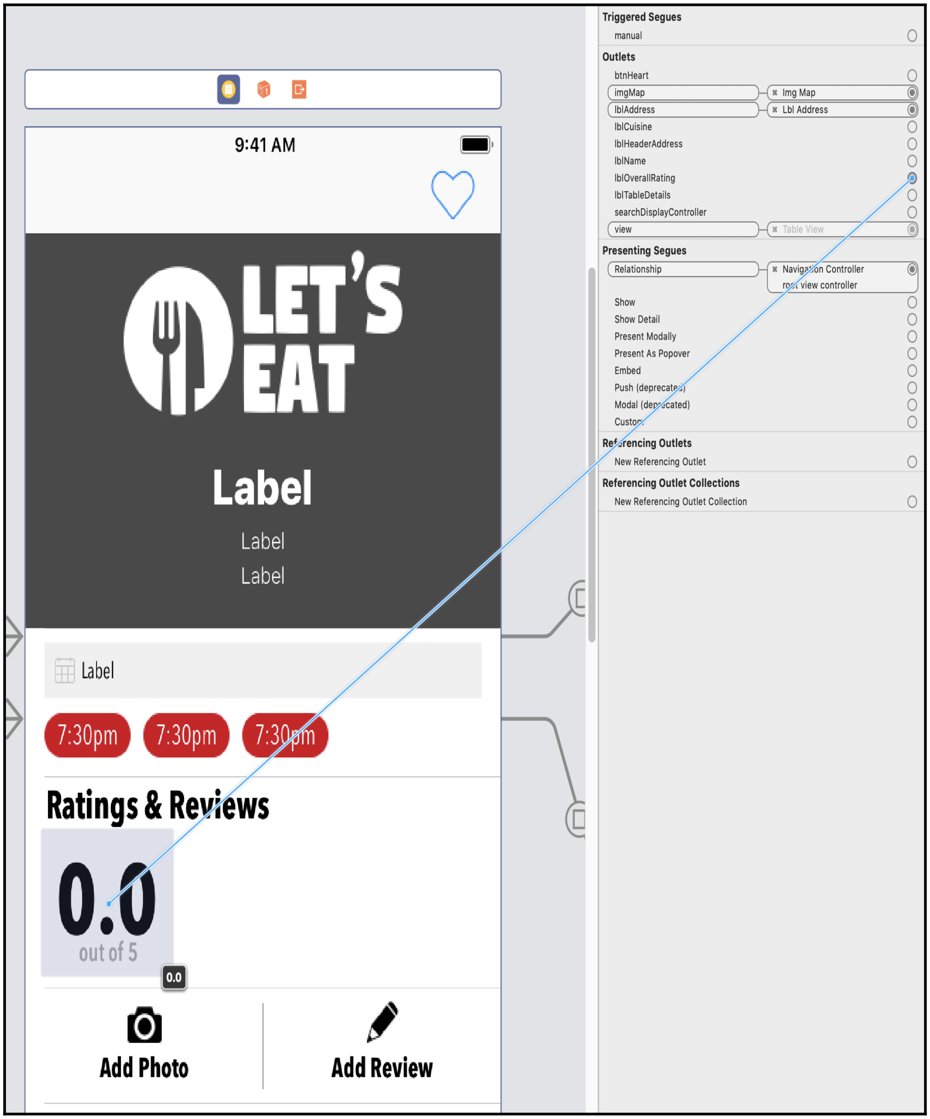


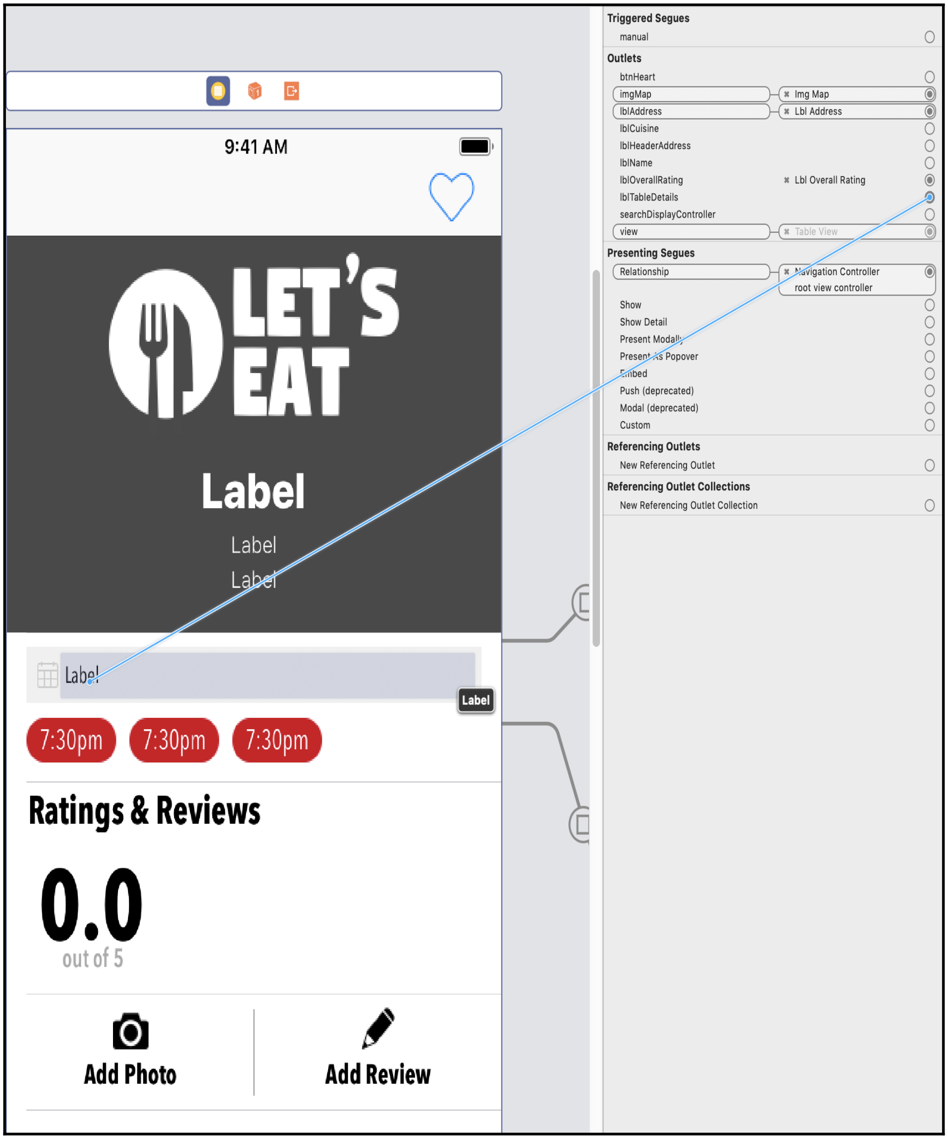
Chapter 16:
Displaying Data in Restaurant Detail


9:41 AM
Photos
Address
Label
Triggered Segues
manual
Outlets
btnHeart
imgMap
lblAddress
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
searchDisplayController
view
Table 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



9:41 AM
Photos
Address
Label
Label
Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
searchDisplayController
view
Table 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


9:41 AM
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
Triggered Segues
manual
Outlets
btnHeart
imgMap
Img Map
lblAddress
Lbl Address
lblCuisine
lblHeaderAddress
lblName
lblOverallRating
lblTableDetails
searchDisplayController
view
Table 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



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 — Navigation Controller root view controller ●

Show ○

Show Detail ○

Present Modally ○

Present As Popover ○

Embed ○

Push (deprecated) ○

Modal (deprecated) ○

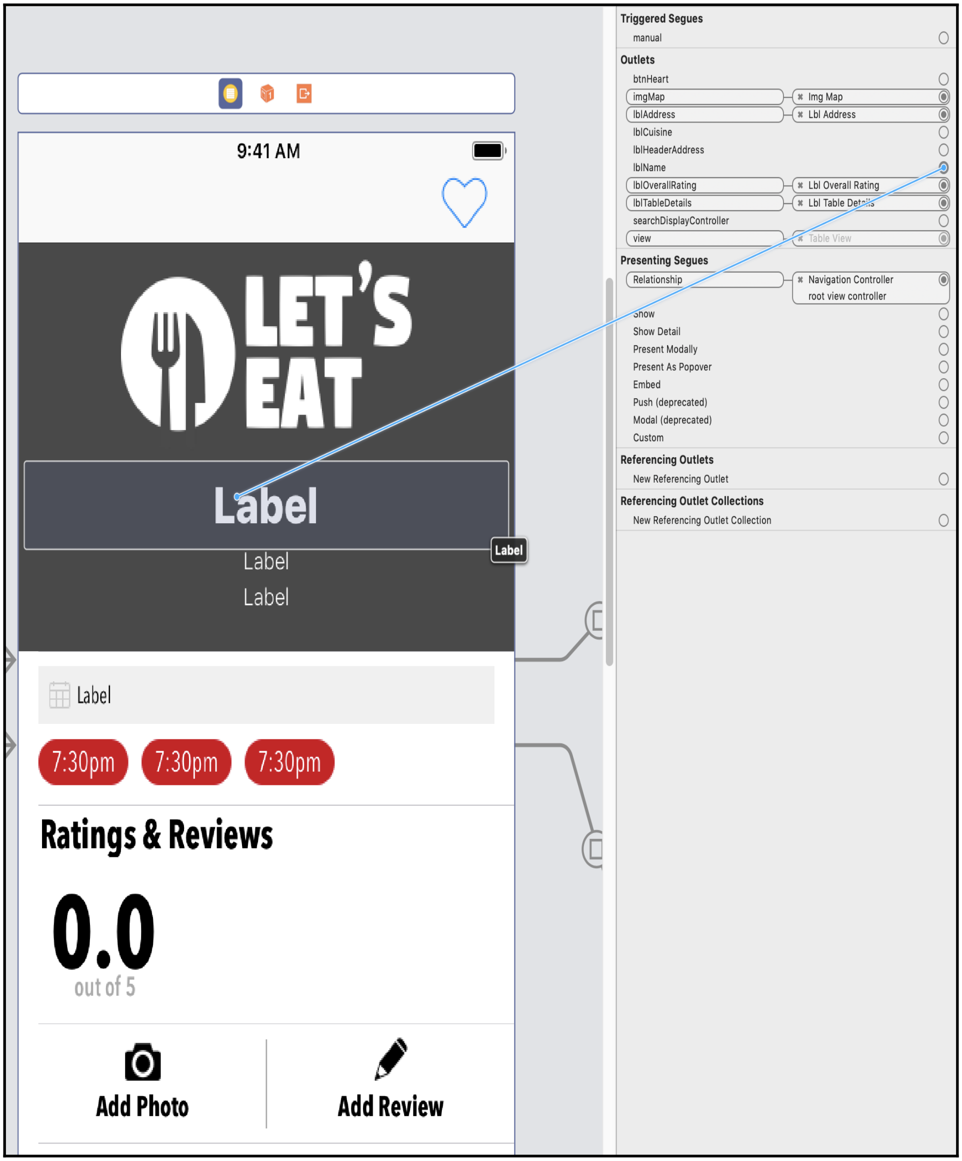
Custom ○

Referencing Outlets

New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○



Triggered Segues

manual ○

Outlets

btnHeart ○	
imgMap	Img Map ◉
lblAddress	Lbl Address ◉
lblCuisine ○	
lblHeaderAddress ○	
lblName ◉	
lblOverallRating	Lbl Overall Rating ◉
lblTableDetails	Lbl Table Details ◉
searchDisplayController ○	
view	Table View ◉

Presenting Segues

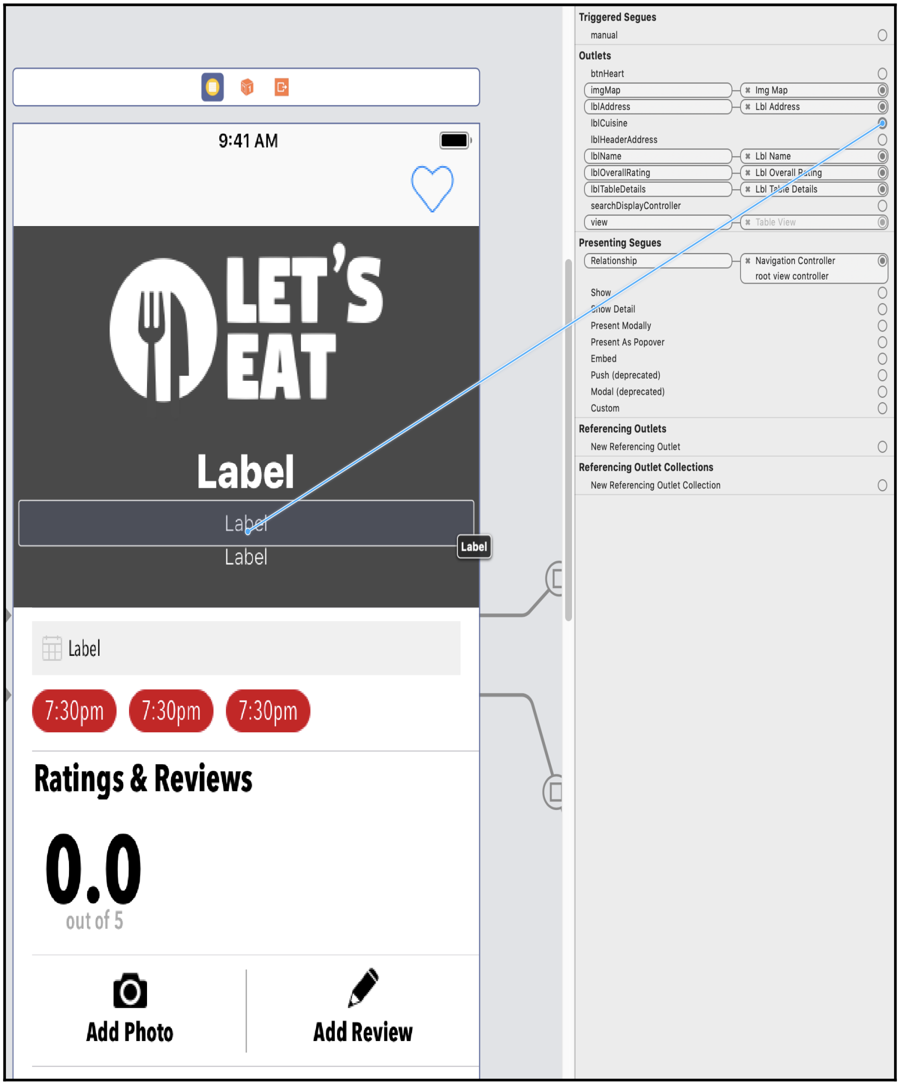
Relationship	Navigation Controller root view controller ◉
Show ○	
Show Detail ○	
Present Modally ○	
Present As Popover ○	
Embed ○	
Push (deprecated) ○	
Modal (deprecated) ○	
Custom ○	

Referencing Outlets

New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○



Triggered Segues

manual

Outlets

btnHeart

imgMap Img Map

lblAddress Lbl Address

lblCuisine

lblHeaderAddress

lblName Lbl Name

lblOverallRating Lbl Overall Rating

lblTableDetails Lbl Table Details

searchDisplayController

view Table View

Presenting Segues

Relationship Navigation Controller root view controller

Show

Show Detail

Present Modally

Present As Popover

Embed

Push (deprecated)

Modal (deprecated)

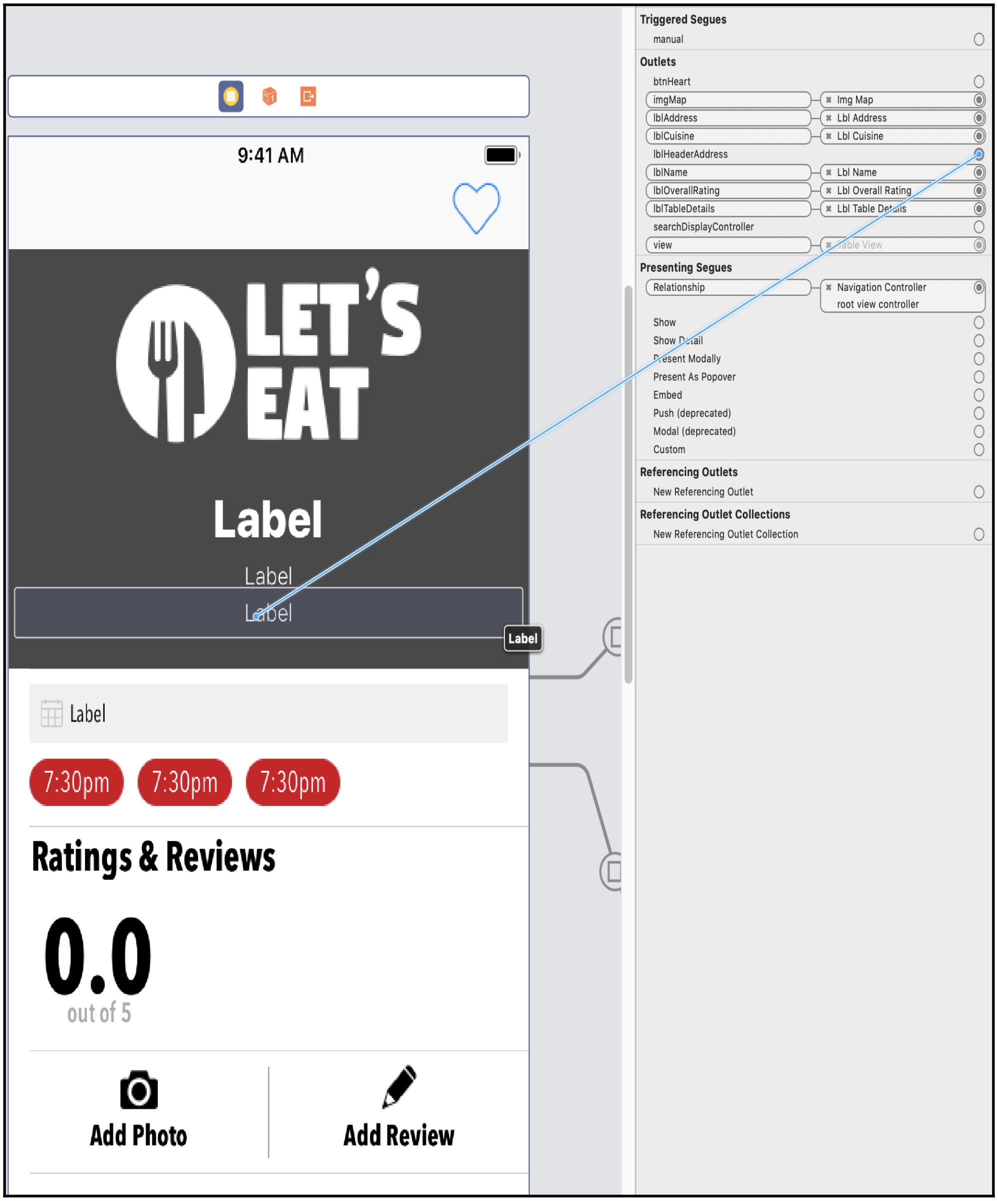
Custom

Referencing Outlets

New Referencing Outlet

Referencing Outlet Collections

New Referencing Outlet Collection



Triggered Segues

manual ○

Outlets

btnHeart	○
imgMap	Img Map ●
lblAddress	Lbl Address ●
lblCuisine	Lbl Cuisine ●
lblHeaderAddress	●
lblName	Lbl Name ●
lblOverallRating	Lbl Overall Rating ●
lblTableDetails	Lbl Table Details ●
searchDisplayController	○
view	Table View ●

Presenting Segues

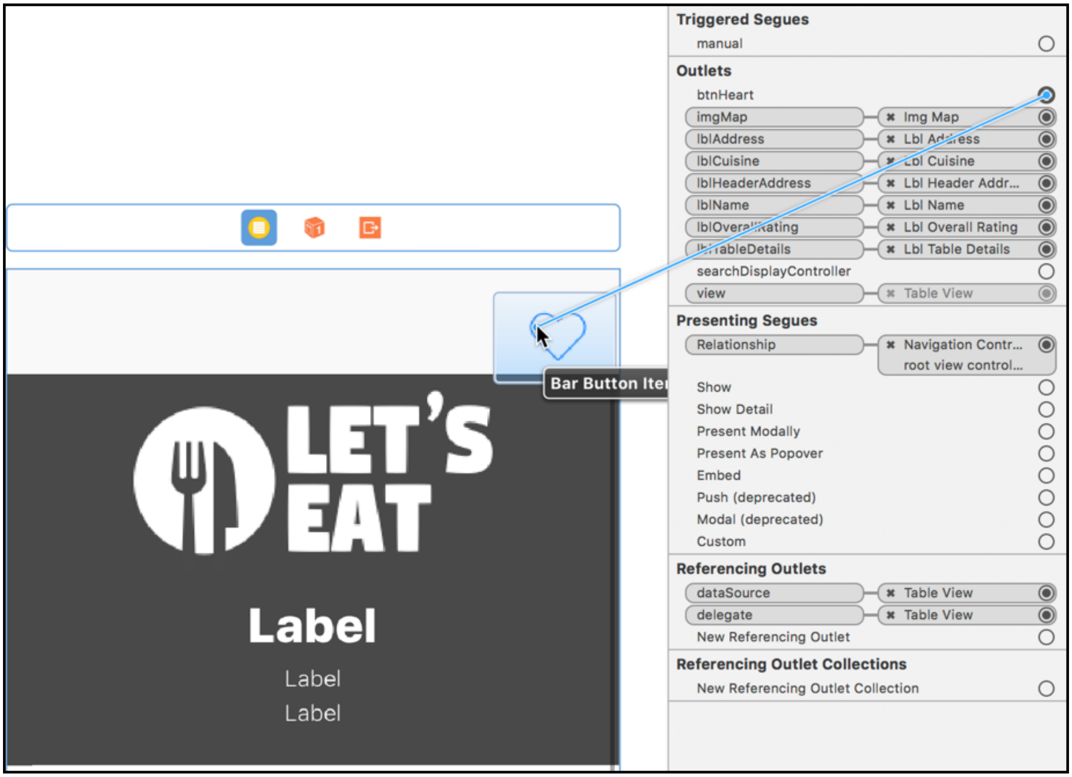
Relationship	Navigation Controller root view controller ●
Show	○
Show Detail	○
Present Modally	○
Present As Popover	○
Embed	○
Push (deprecated)	○
Modal (deprecated)	○
Custom	○

Referencing Outlets

New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○



Triggered Segues

manual ○

Outlets

btnHeart ●
imgMap — Img Map ●
lblAddress — Lbl Address ●
lblCuisine — Lbl Cuisine ●
lblHeaderAddress — Lbl Header Addr... ●
lblName — Lbl Name ●
lblOverallRating — Lbl Overall Rating ●
lblTableDetails — Lbl Table Details ●
searchDisplayController ○
view — Table View ●

Presenting Segues

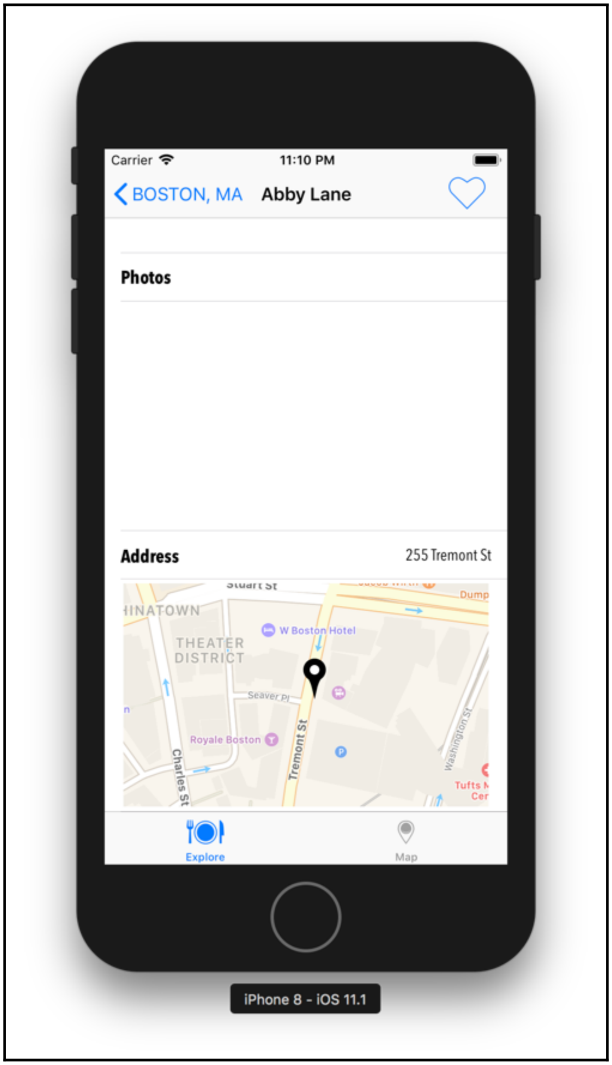
Relationship — Navigation Contr... root view control... ●
Show ○
Show Detail ○
Present Modally ○
Present As Popover ○
Embed ○
Push (deprecated) ○
Modal (deprecated) ○
Custom ○

Referencing Outlets

dataSource — Table View ●
delegate — Table View ●
New Referencing Outlet ○

Referencing Outlet Collections

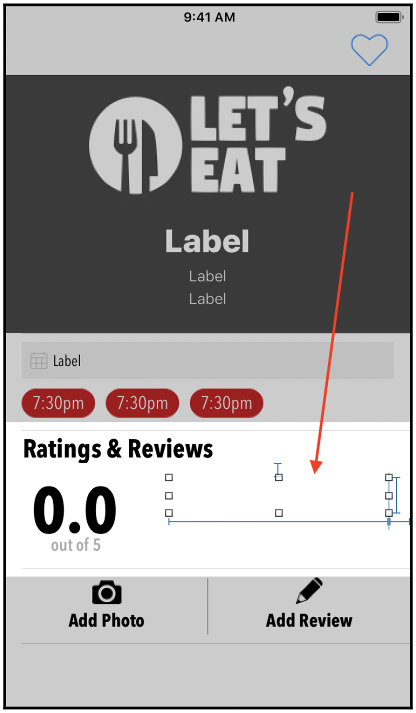
New Referencing Outlet Collection ○

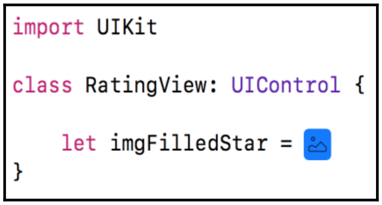


Photos

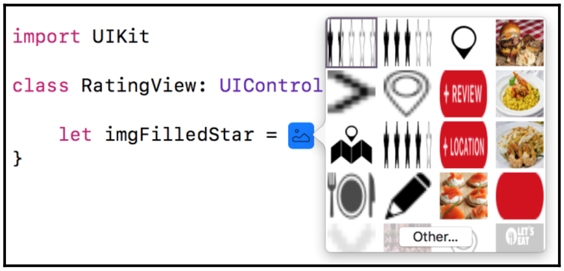
Address 255 Tremont St

Chapter 17: Foodie Reviews

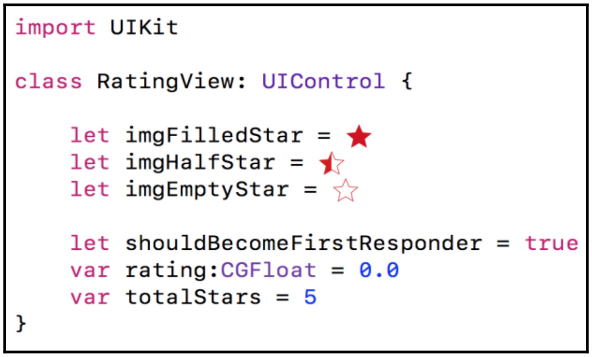




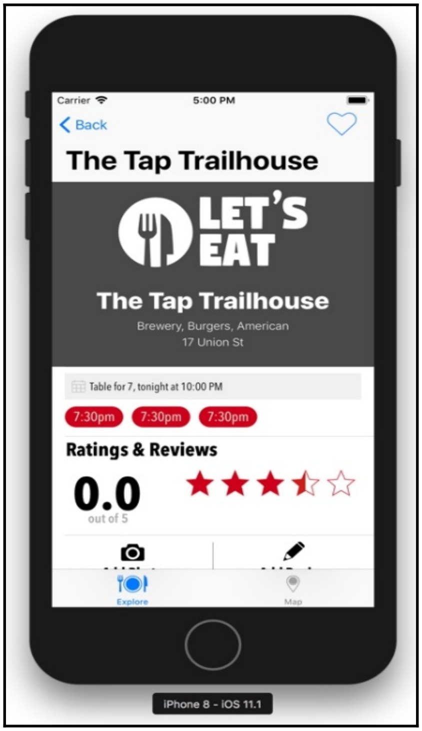
```
import UIKit

class RatingView: UIControl {

    let imgFilledStar = 
}
```

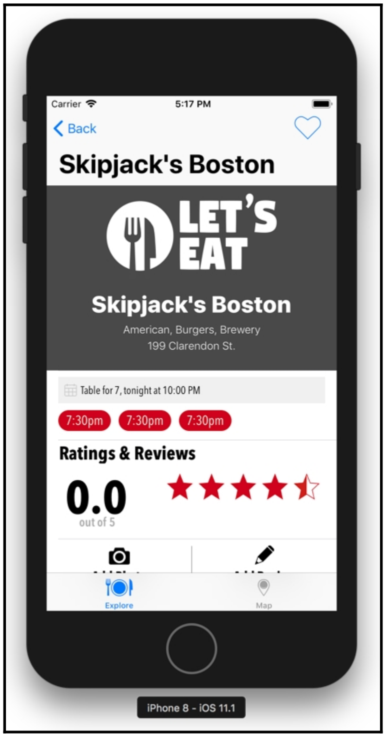


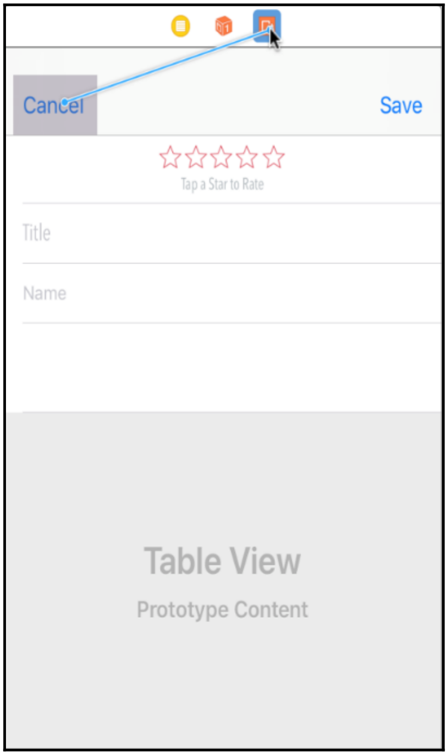
```
import UIKit

class RatingView: UIControl

    let imgFilledStar = 
}
```



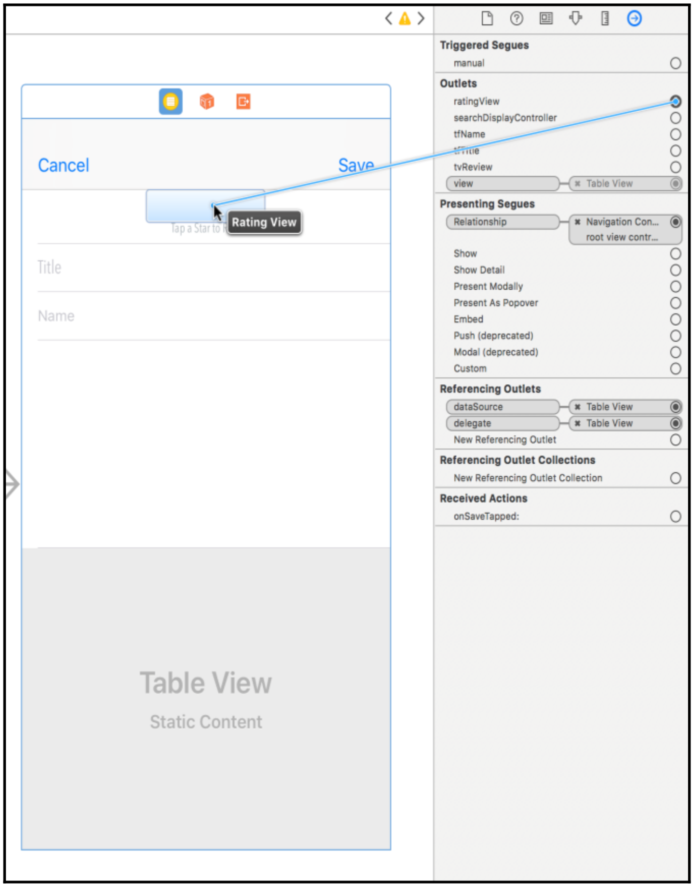
```swift
import UIKit

class RatingView: UIControl {

    let imgFilledStar = ★
    let imgHalfStar = ⯪
    let imgEmptyStar = ☆

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

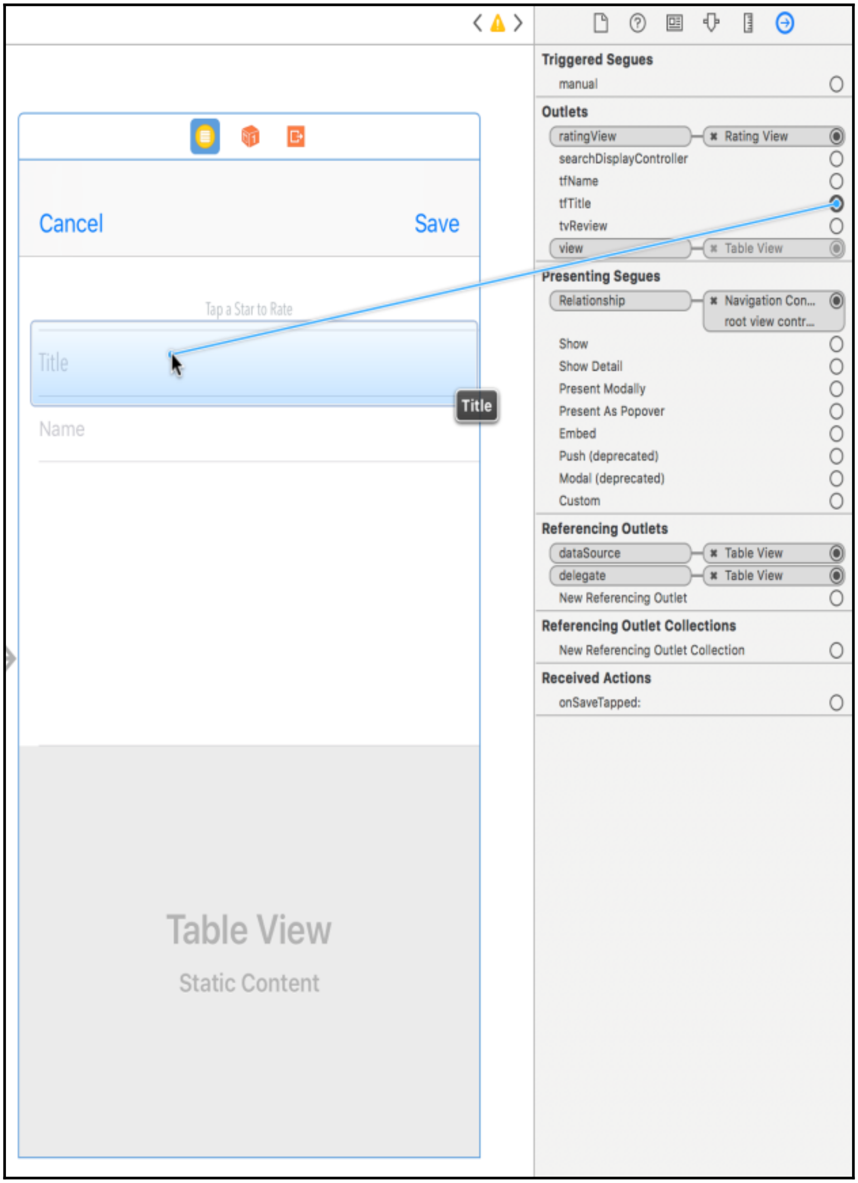


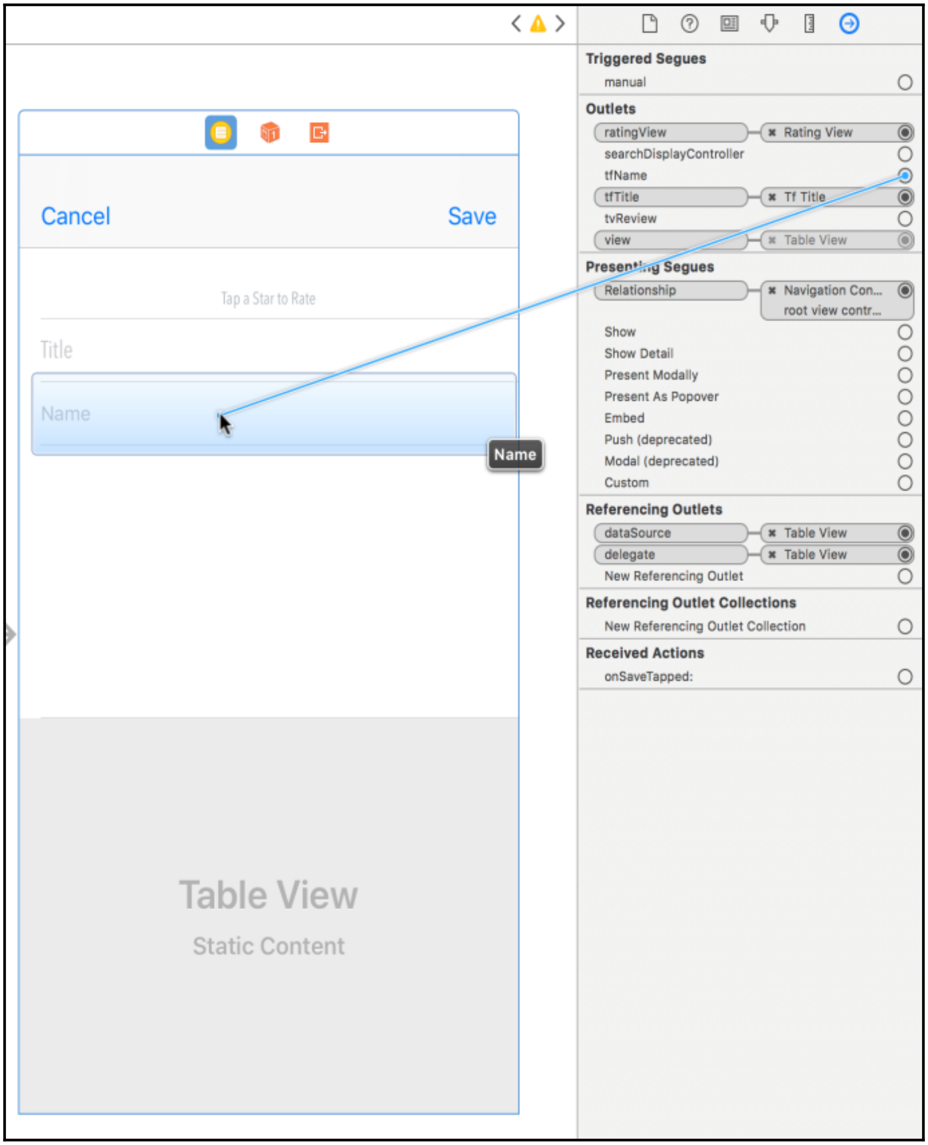

Carrier
5:17 PM
Back
Skipjack's Boston
LET'S EAT
Skipjack's Boston
American, Burgers, Brewery
199 Clarendon St.
Table for 7, tonight at 10:00 PM
7:30pm
7:30pm
7:30pm
Ratings & Reviews
0.0
out of 5
Explore
Map
iPhone 8 - iOS 11.1



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


Cancel
Save
Tap a Star to
Rating View
Title
Name
Table View
Static Content
Triggered Segues
manual
Outlets
ratingView
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
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:



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

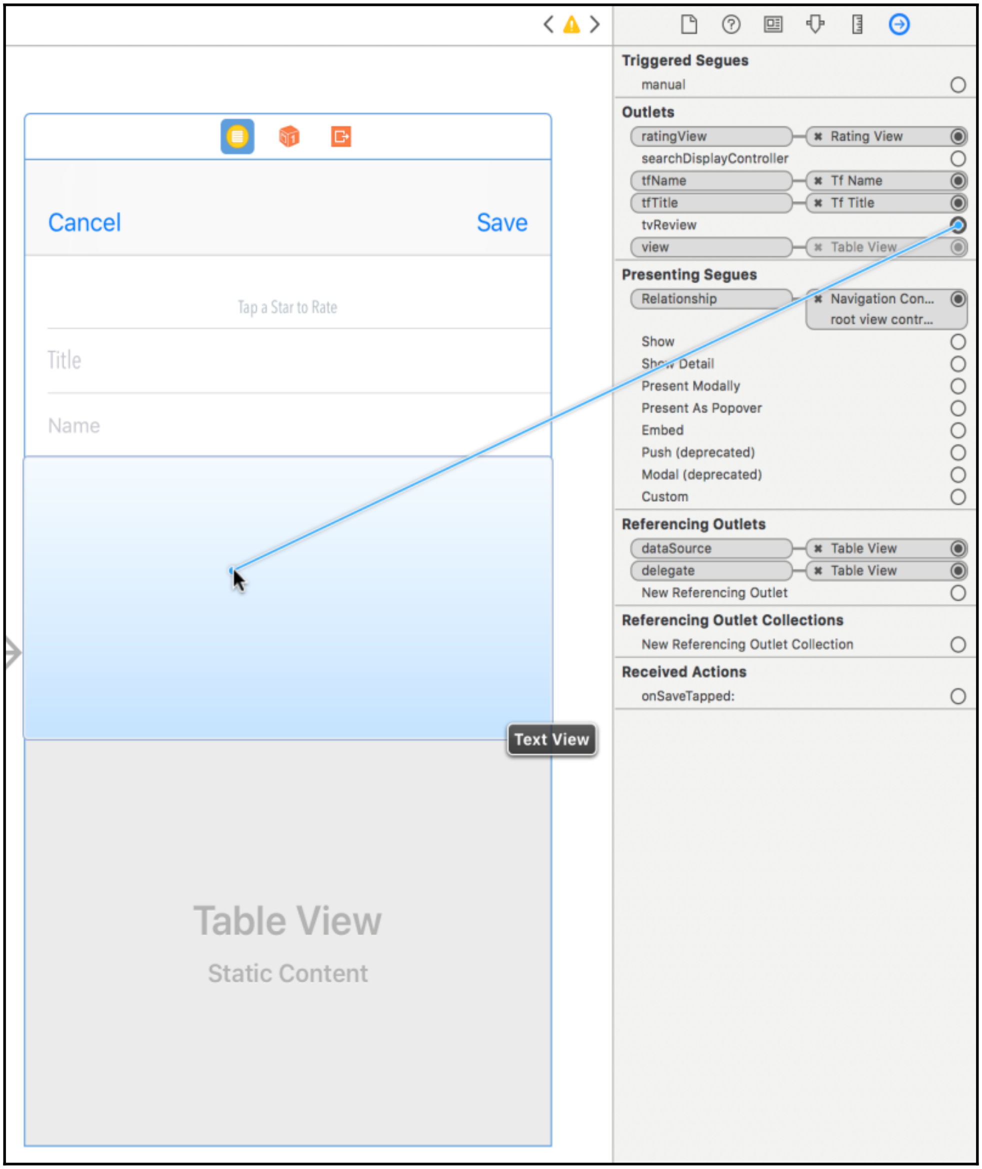
dataSource — Table View ●
delegate — Table View ●
New Referencing Outlet ○

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

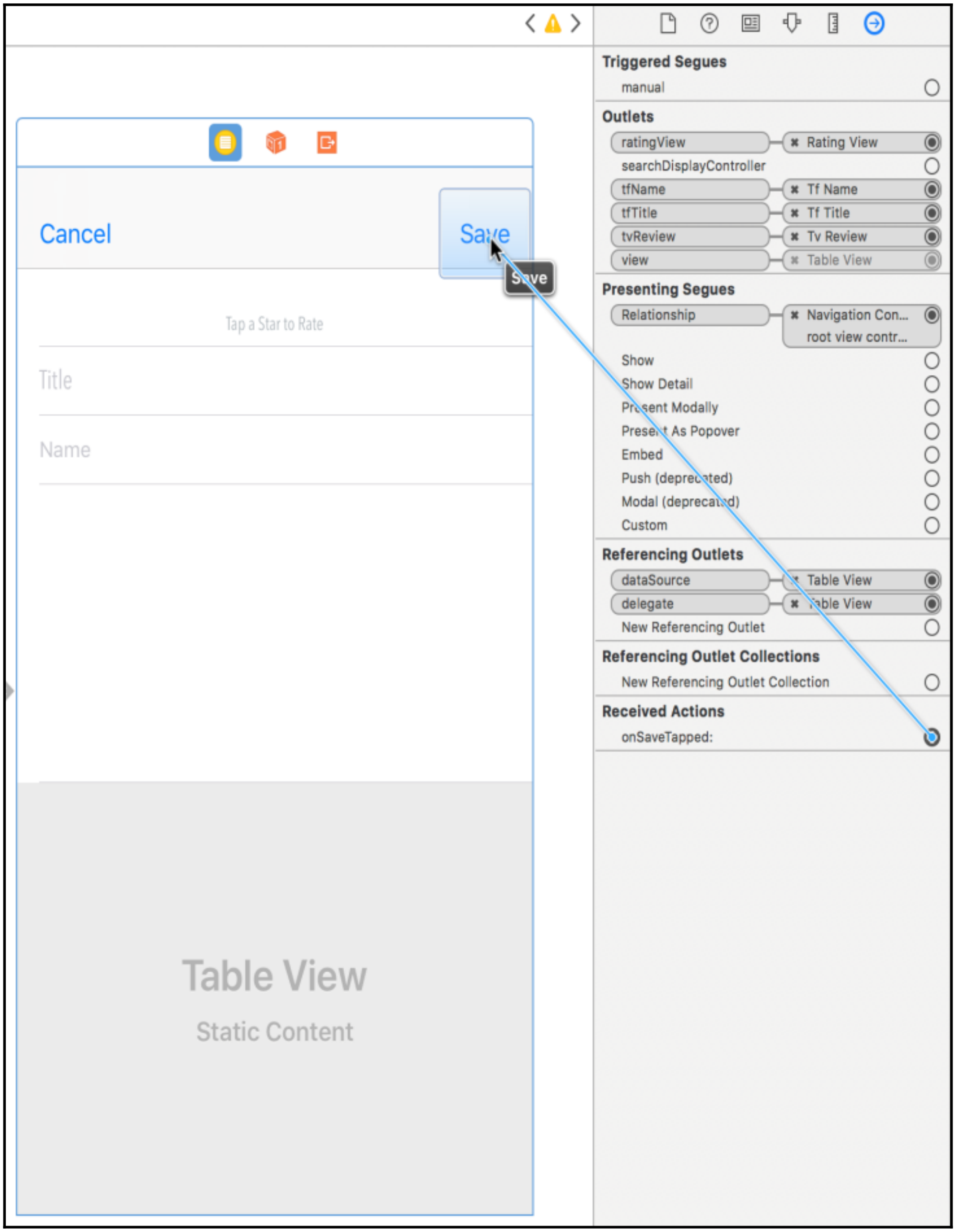
dataSource — Table View ●
delegate — Table View ●
New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○

Received Actions

onSaveTapped: ○



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 — Table View ●

New Referencing Outlet ○

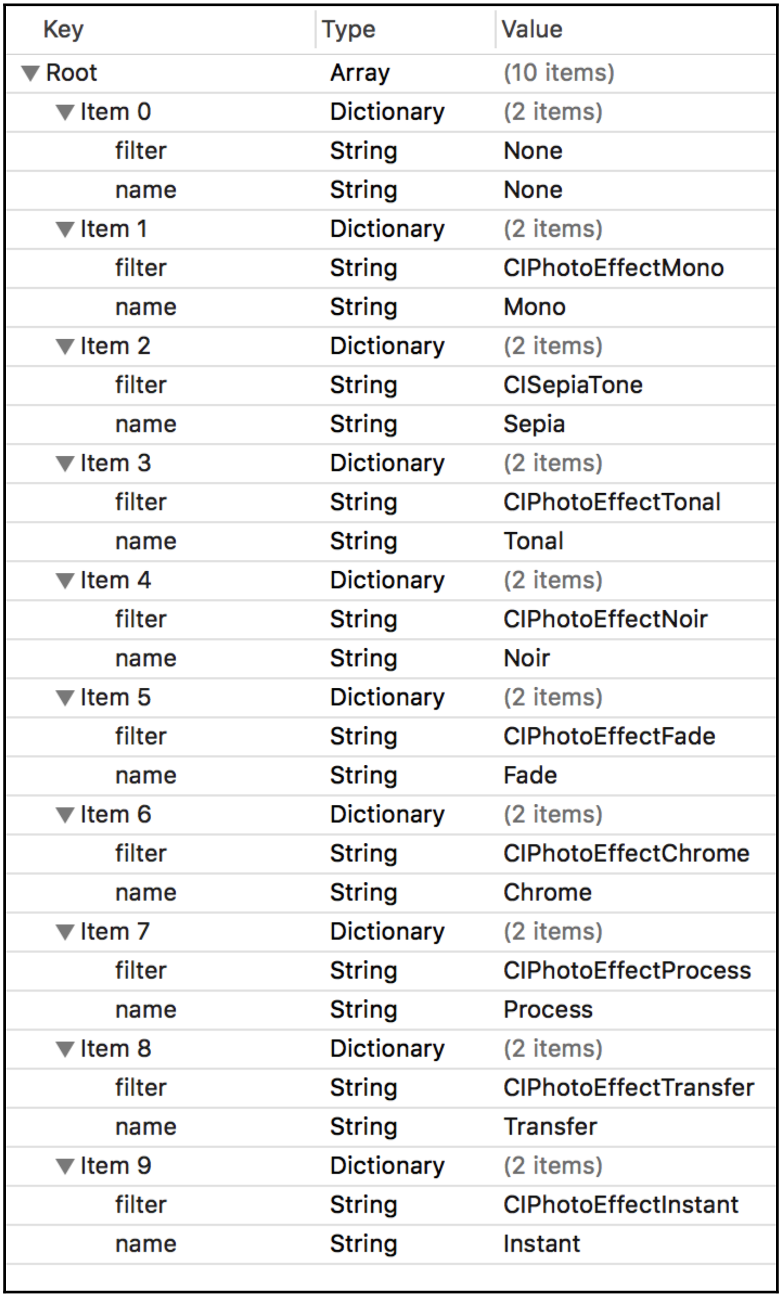
Referencing Outlet Collections

New Referencing Outlet Collection ○

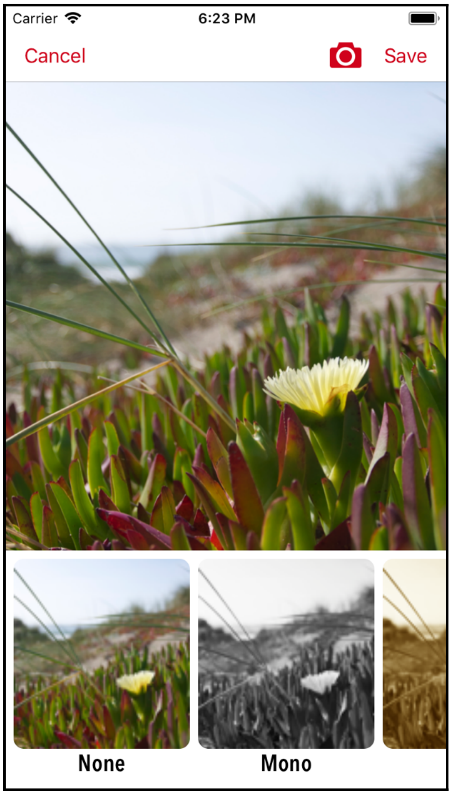
Received Actions

onSaveTapped: ○

Chapter 18: 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

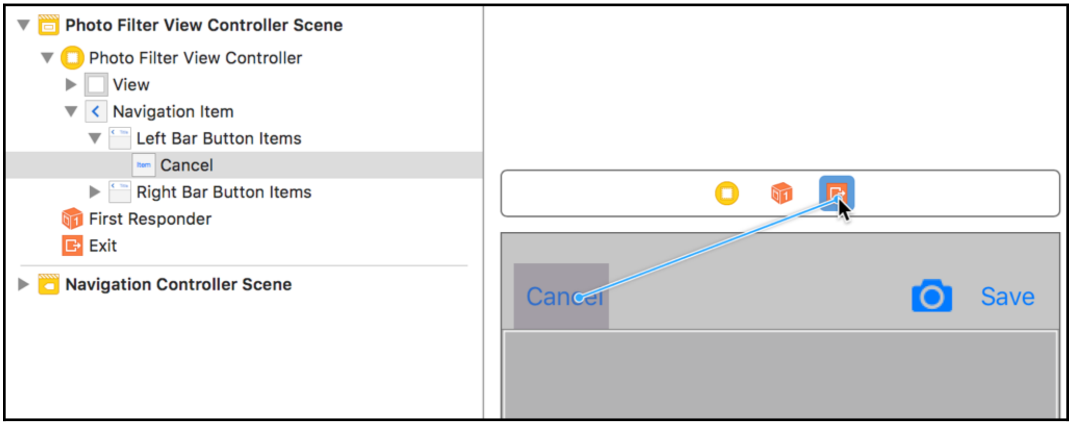
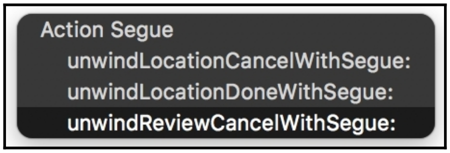
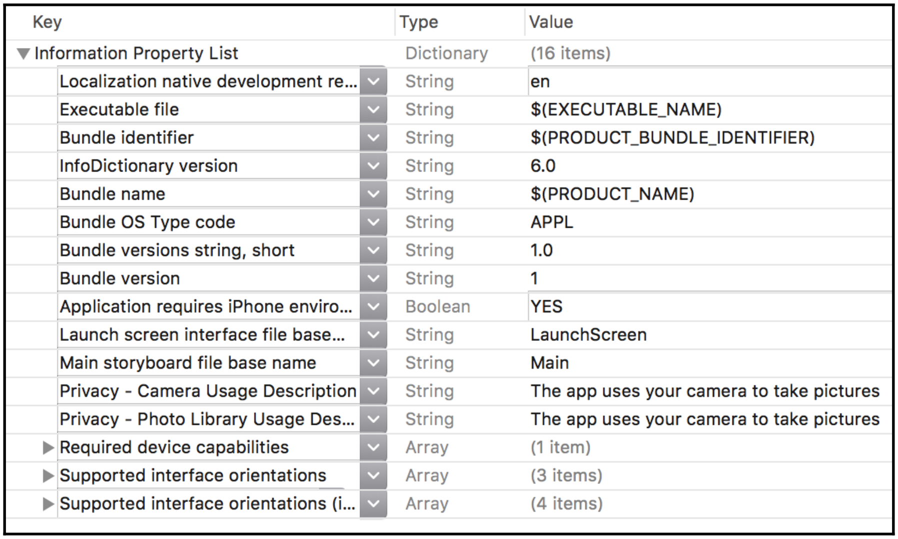
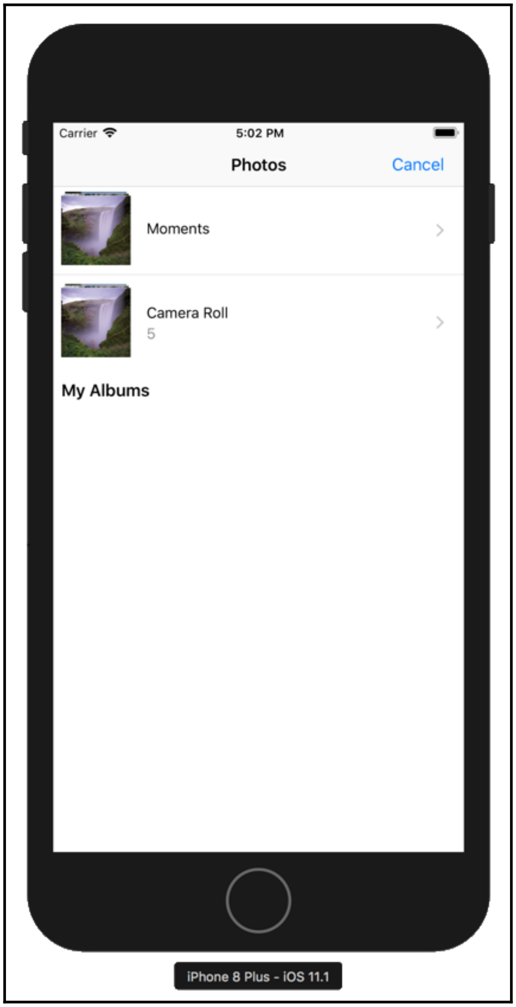


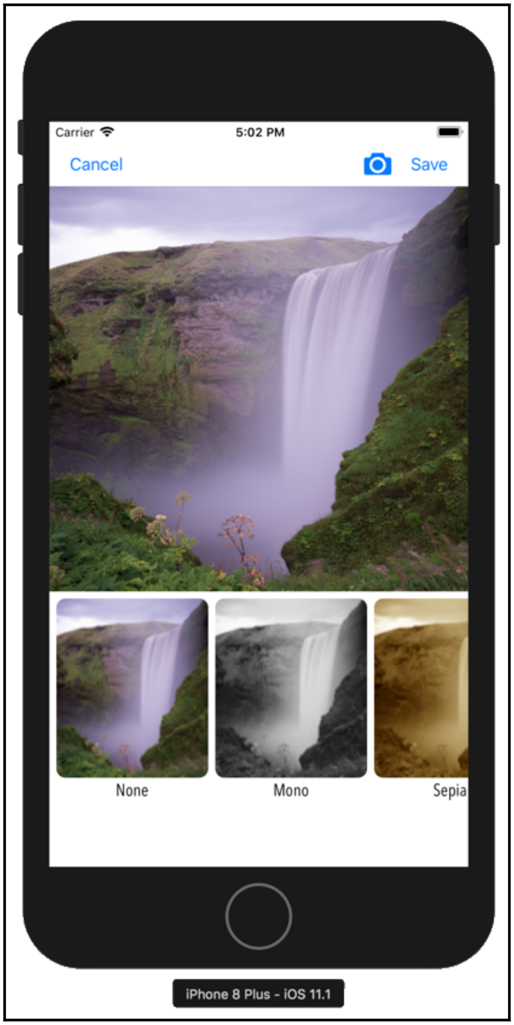
Photo Filter View Controller Scene
Photo Filter View Controller
View
Navigation Item
Left Bar Button Items
Cancel
Right Bar Button Items
First Responder
Exit
Navigation Controller Scene
Cancel
Save


Action Segue
unwindLocationCancelWithSegue:
unwindLocationDoneWithSegue:
unwindReviewCancelWithSegue:

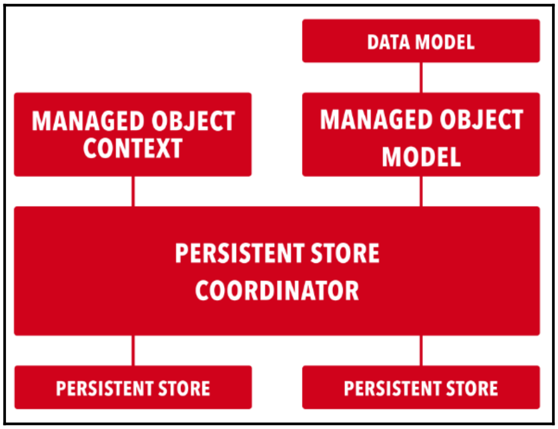


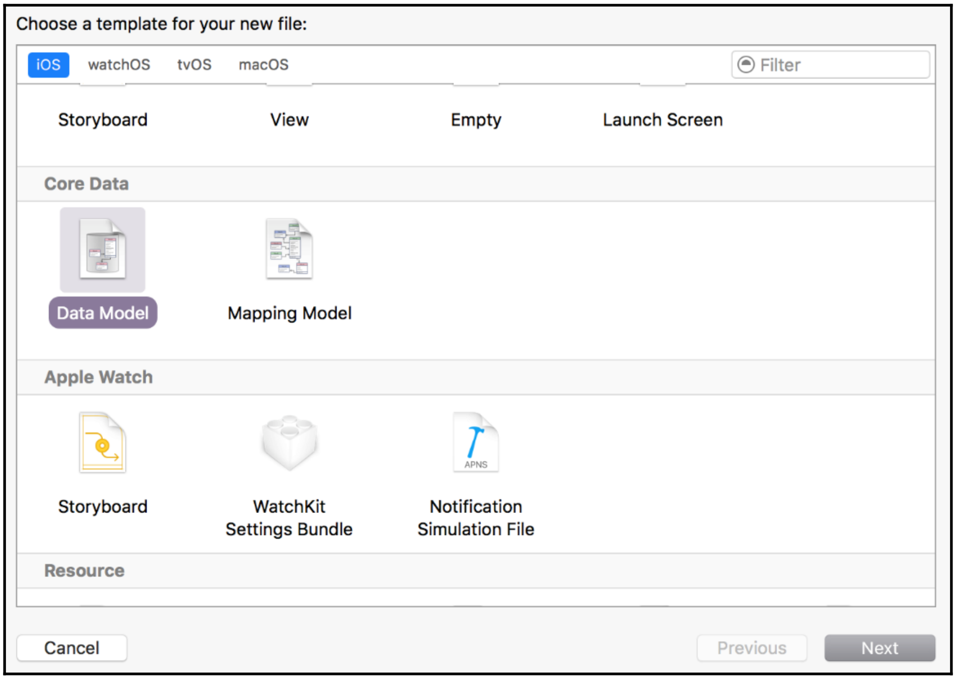
Key	Type	Value
▼ Information Property List	Dictionary	(16 items)
Localization native development re...	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	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)

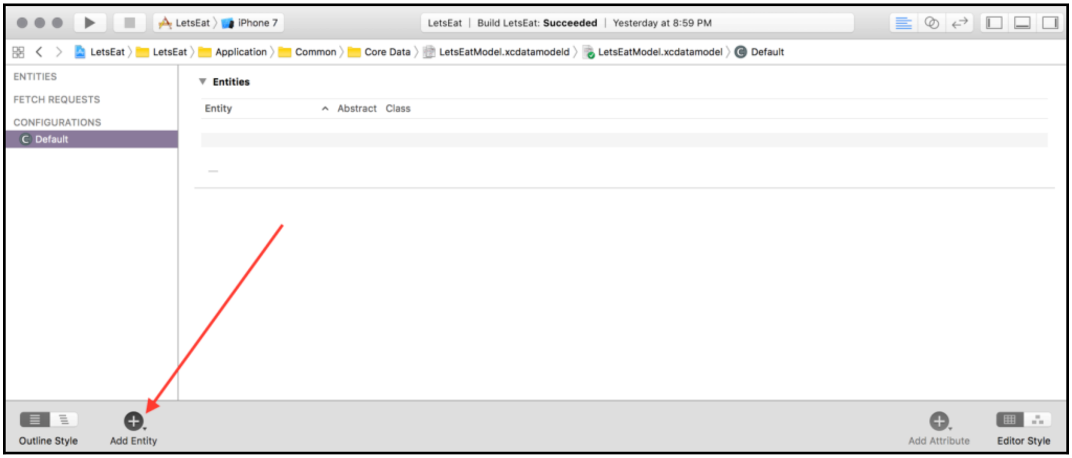

Carrier
5:02 PM
Photos
Cancel
Moments
Camera Roll
5
My Albums
iPhone 8 Plus - iOS 11.1

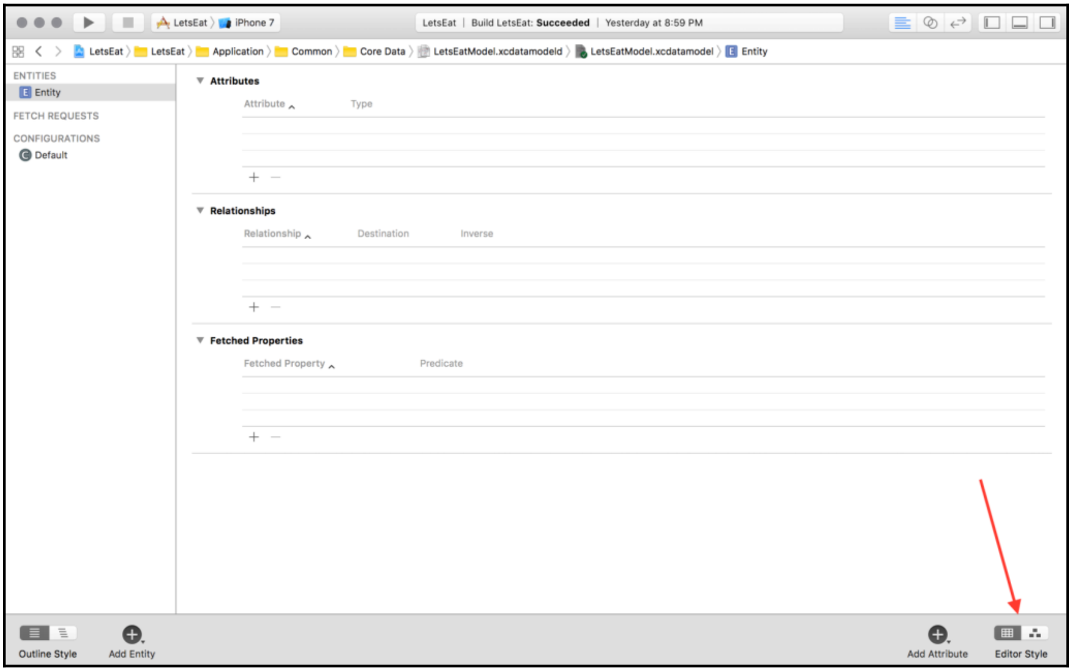

Carrier
5:02 PM
Cancel
Save
None
Mono
Sepia
iPhone 8 Plus - iOS 11.1

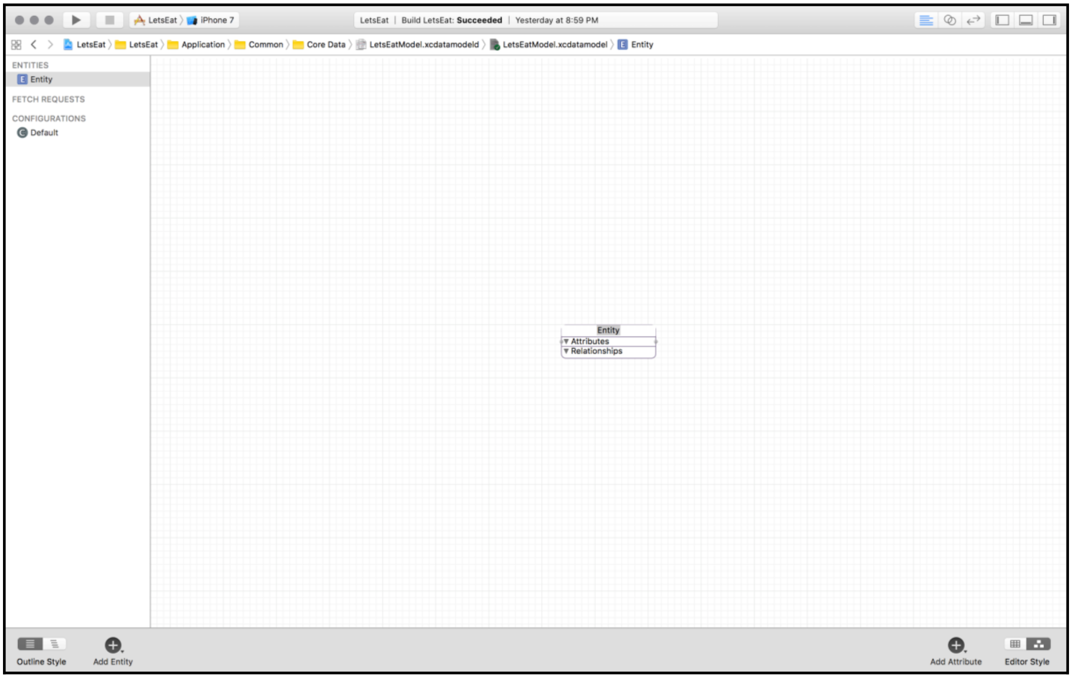
Chapter 19: Understanding Core Data

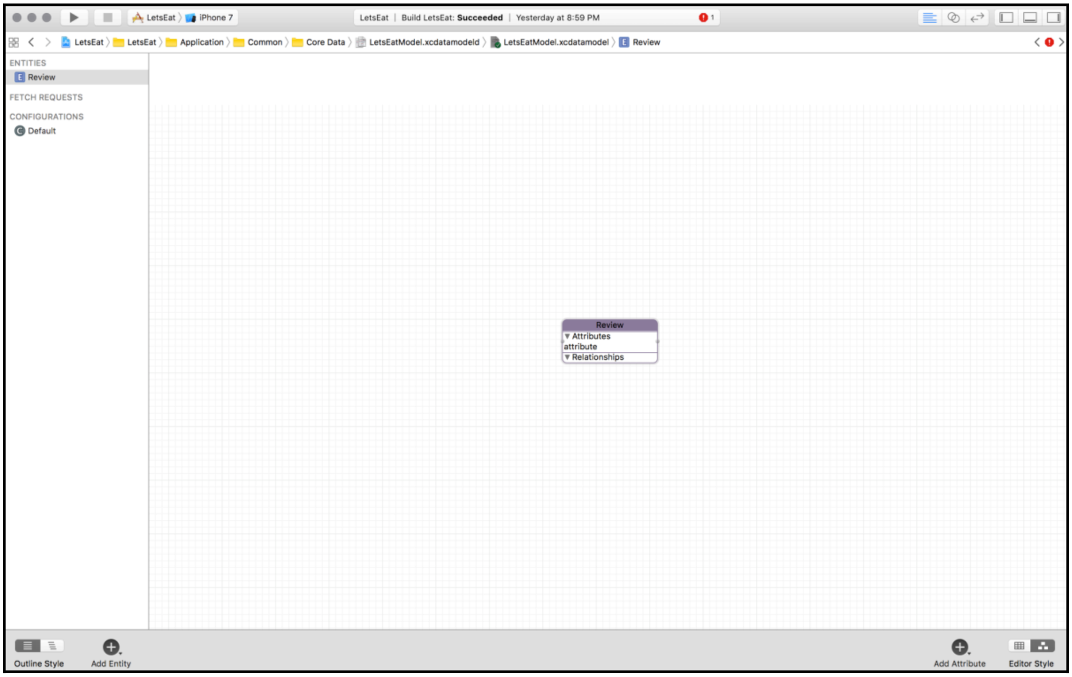


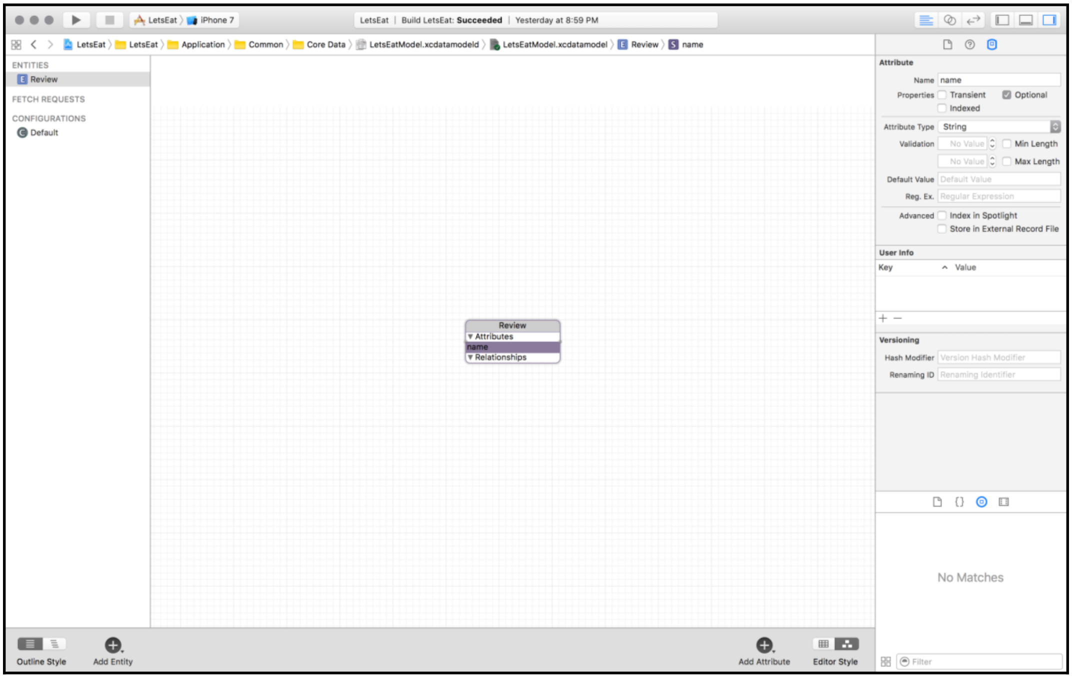

Choose a template for your new file:


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
1
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
Outline Style
Add Entity
Add Attribute
Editor Style
No Matches
Filter

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

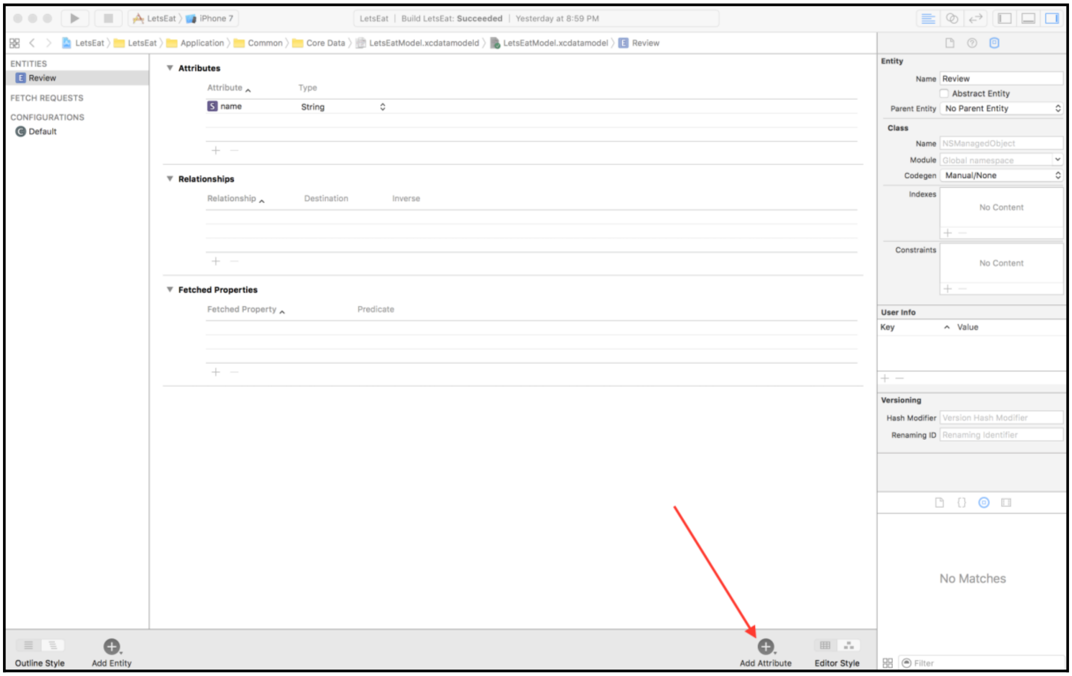
User Info

Key	Value

Versioning

Hash Modifier: Version Hash Modifier

Renaming ID: Renaming Identifier



Attribute: name — Type: String

Name: Review

□ Abstract Entity

Parent Entity: No Parent Entity

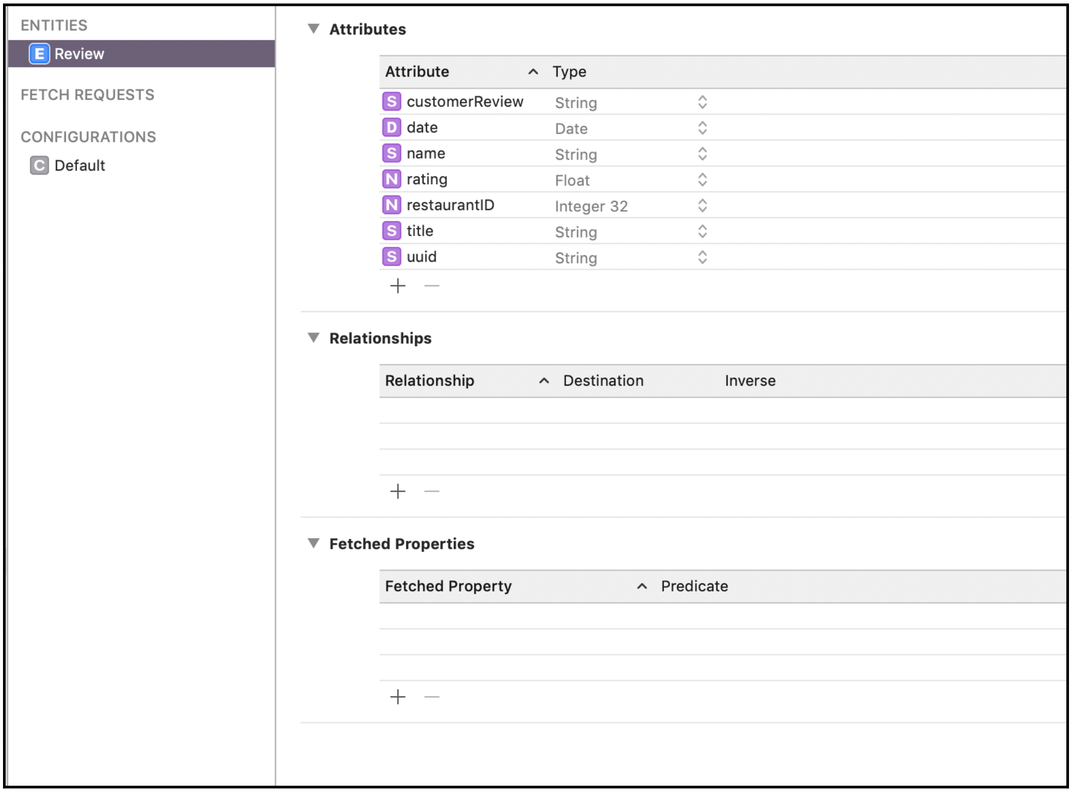
Name: NSManagedObject

Module: Global namespace

Codegen: Manual/None

Hash Modifier: Version Hash Modifier

Renaming ID: Renaming Identifier



ENTITIES

E Review

FETCH REQUESTS

CONFIGURATIONS

C Default

▼ Attributes

Attribute	Type
S customerReview	String
D date	Date
S name	String
N rating	Float
N restaurantID	Integer 32
S title	String
S uuid	String

+ —

▼ Relationships

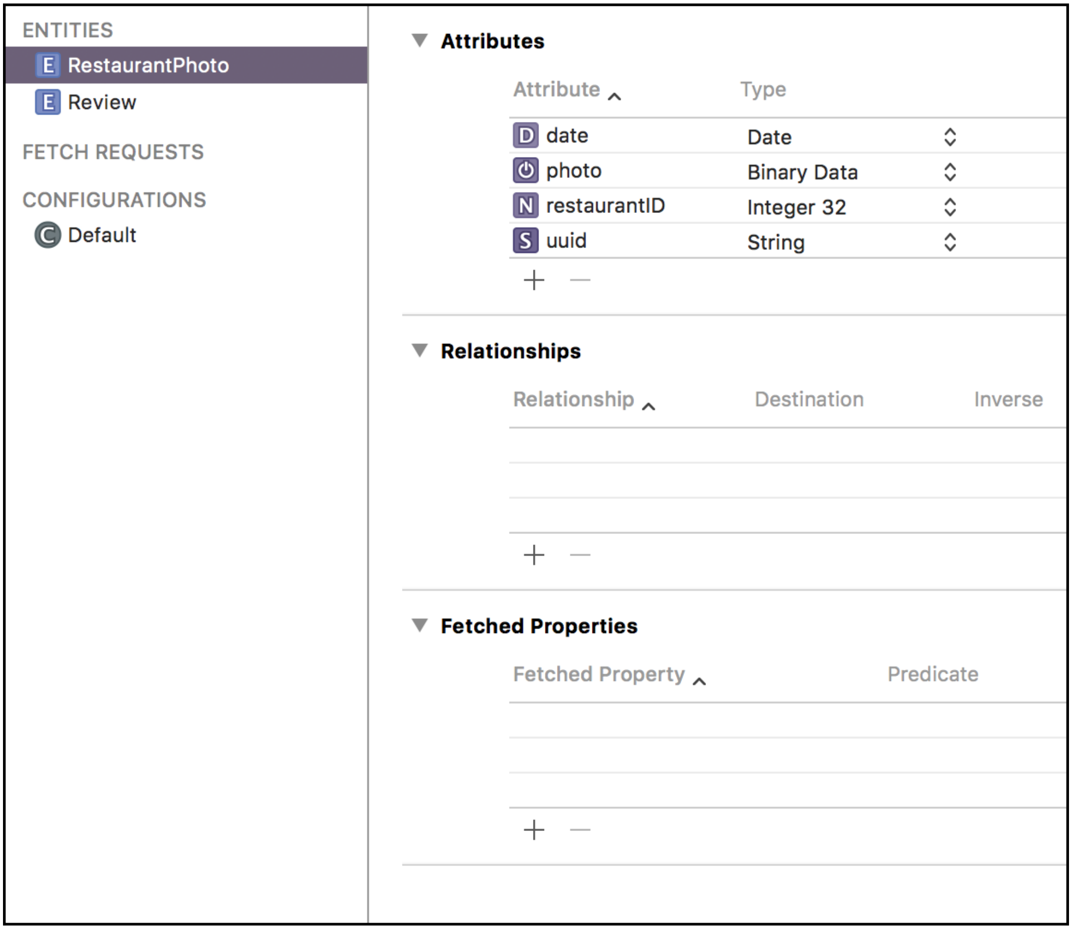
Relationship	Destination	Inverse

+ —

▼ Fetched Properties

Fetched Property	Predicate

+ —



Attributes

Attribute	Type
D date	Date
photo	Binary Data
N restaurantID	Integer 32
S uuid	String

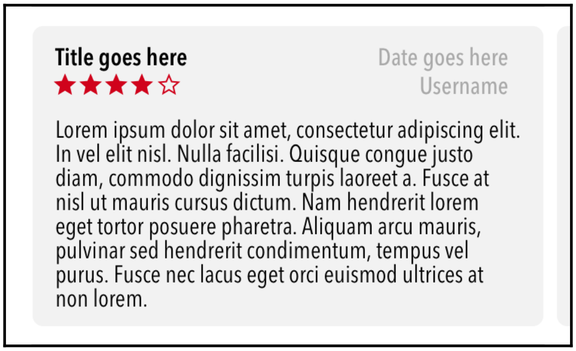
Relationships

Relationship	Destination	Inverse

Fetched Properties

Fetched Property	Predicate

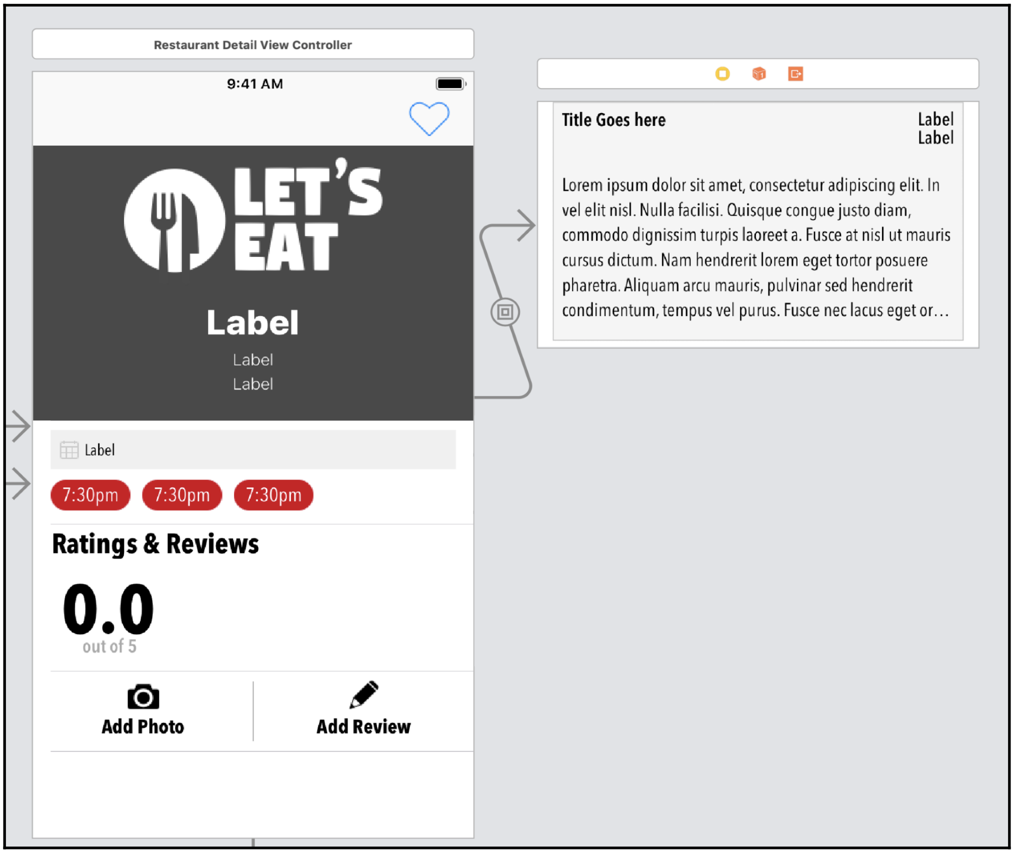
Chapter 20: Saving Reviews



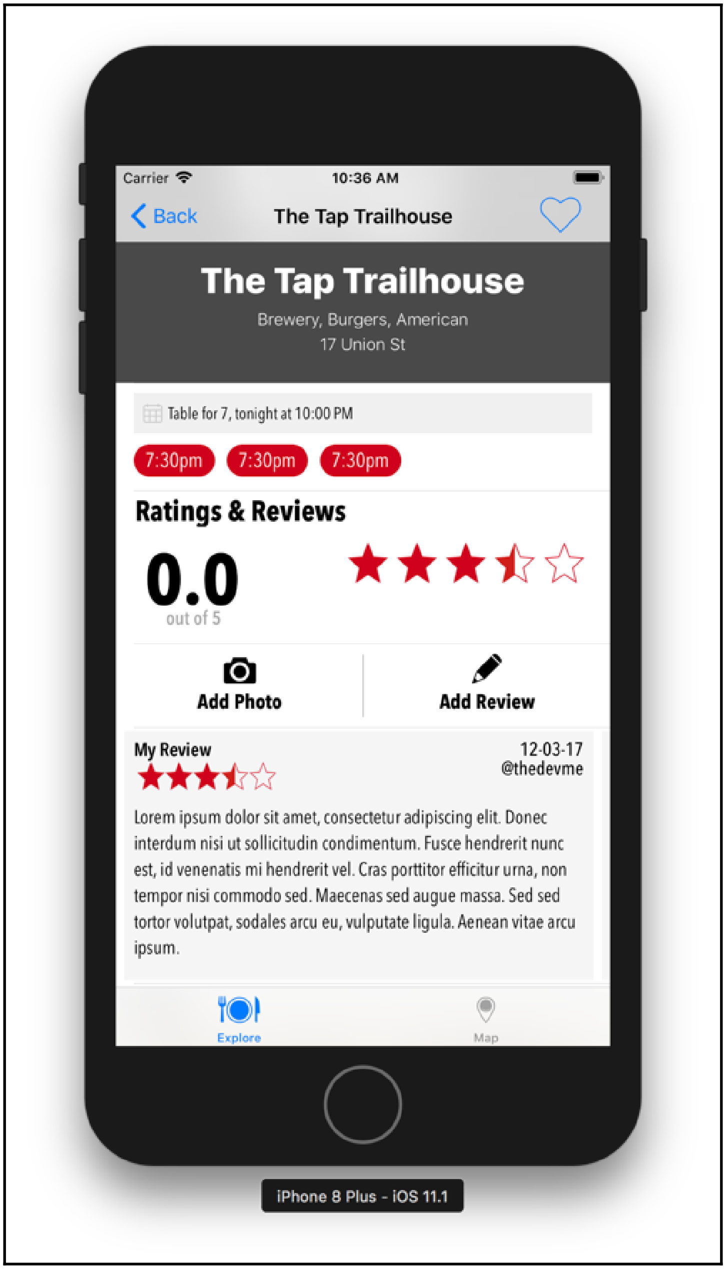
Title goes here
★★★★☆

Date goes here
Username

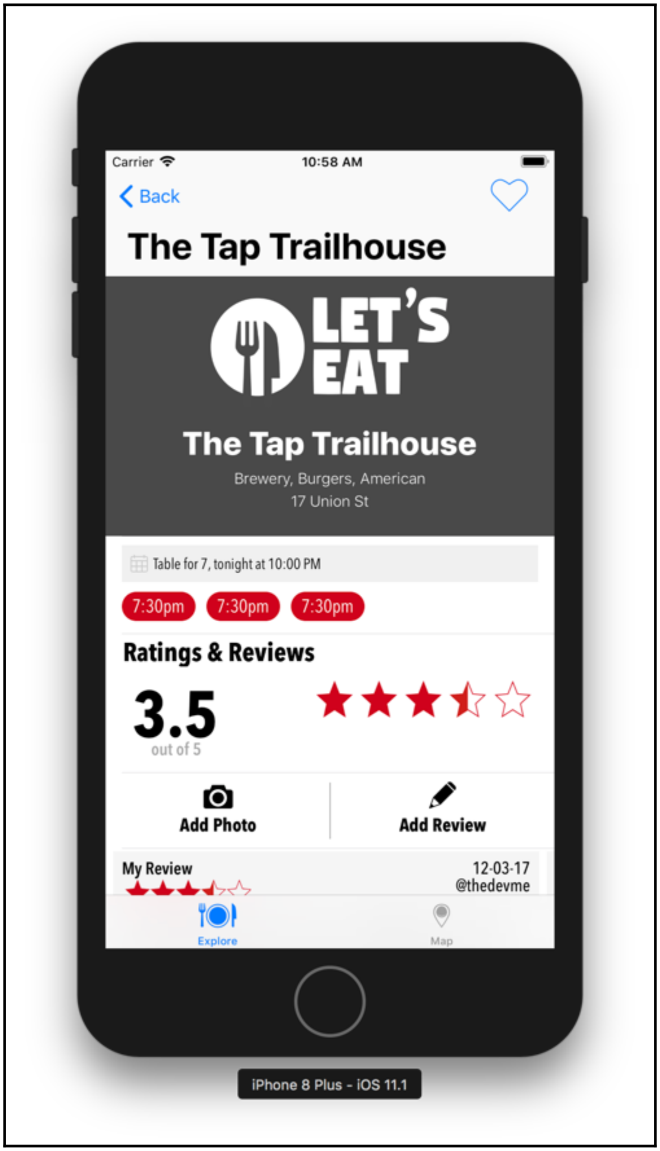
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vel elit nisl. Nulla facilisi. Quisque congue justo diam, commodo dignissim turpis laoreet a. Fusce at nisl ut mauris cursus dictum. Nam hendrerit lorem eget tortor posuere pharetra. Aliquam arcu mauris, pulvinar sed hendrerit condimentum, tempus vel purus. Fusce nec lacus eget orci euismod ultrices at non lorem.



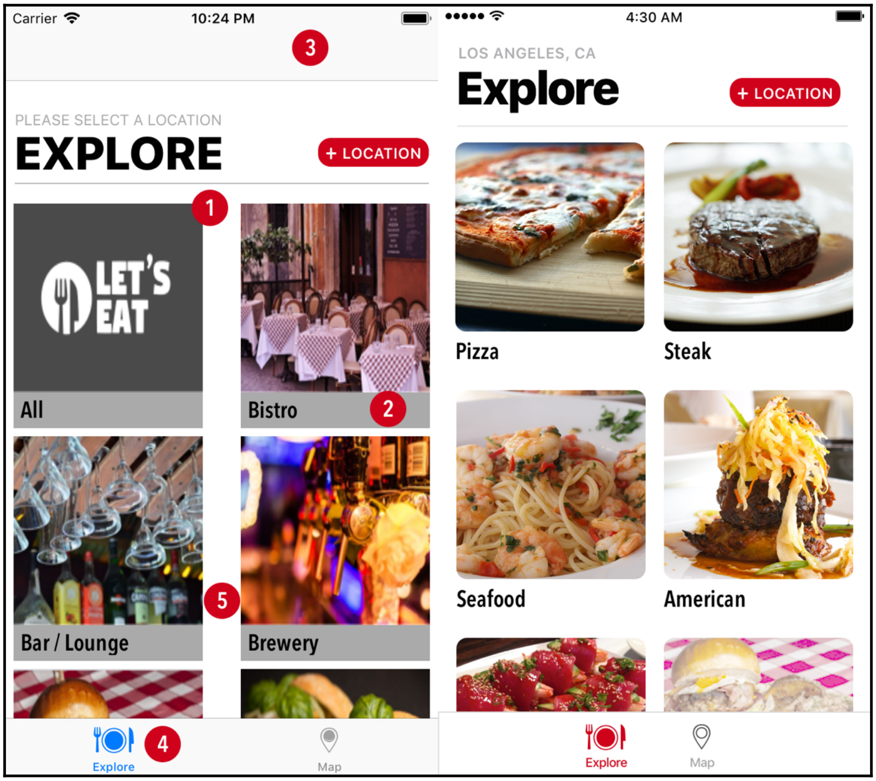
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vel elit nisl. Nulla facilisi. Quisque congue justo diam, commodo dignissim turpis laoreet a. Fusce at nisl ut mauris cursus dictum. Nam hendrerit lorem eget tortor posuere pharetra. Aliquam arcu mauris, pulvinar sed hendrerit condimentum, tempus vel purus. Fusce nec lacus eget or…

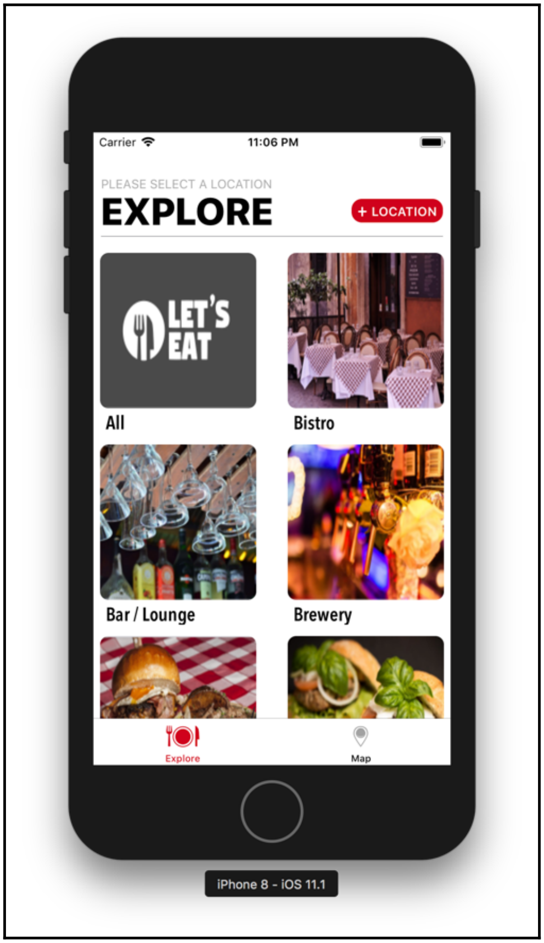


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec interdum nisi ut sollicitudin condimentum. Fusce hendrerit nunc est, id venenatis mi hendrerit vel. Cras porttitor efficitur urna, non tempor nisi commodo sed. Maecenas sed augue massa. Sed sed tortor volutpat, sodales arcu eu, vulputate ligula. Aenean vitae arcu ipsum.


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 21: 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

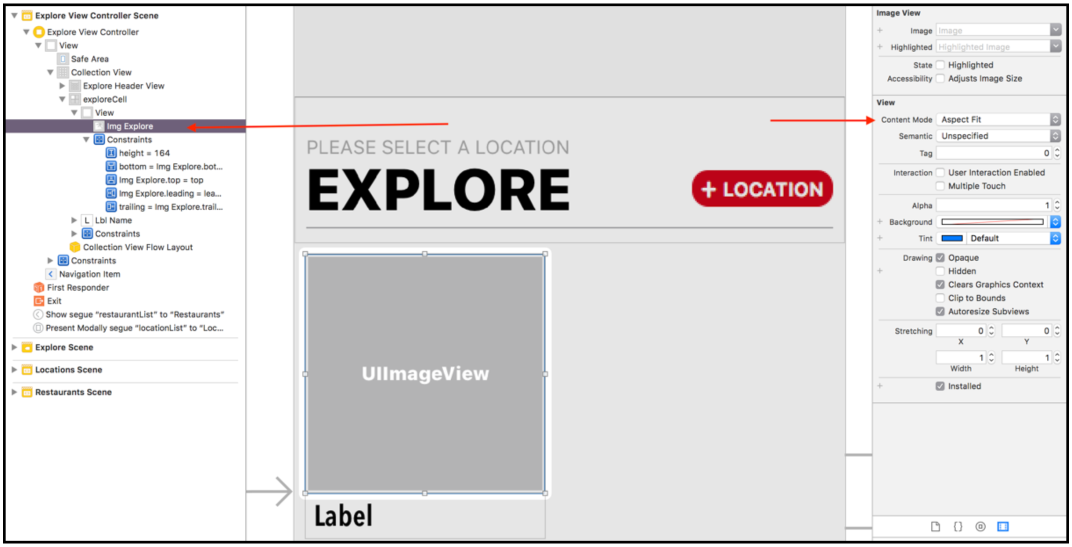


Image View

Image: Image

Highlighted: Highlighted Image

State: □ Highlighted

Accessibility: □ Adjusts Image Size

View

Content Mode: Aspect Fit

Semantic: Unspecified

Tag: 0

Interaction: □ User Interaction Enabled
□ Multiple Touch

Alpha: 1

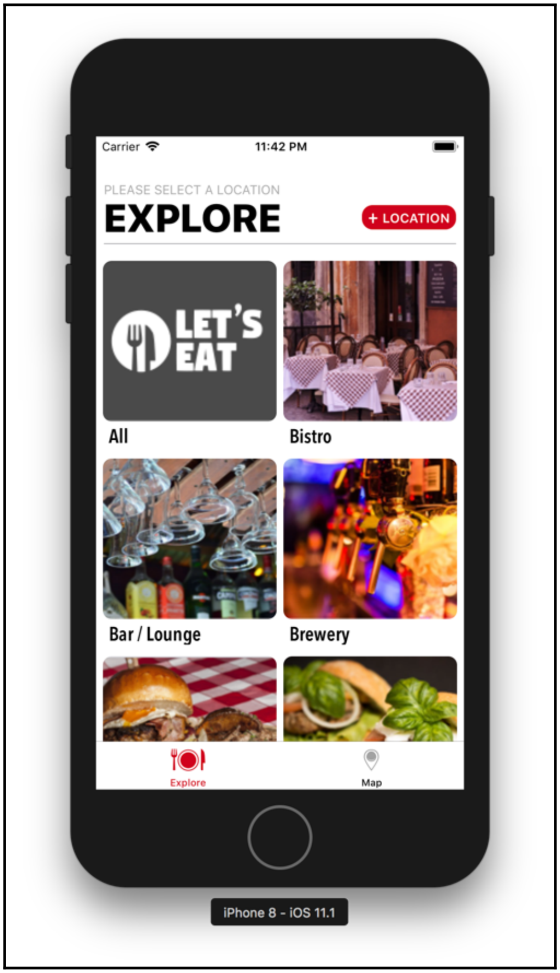
Background

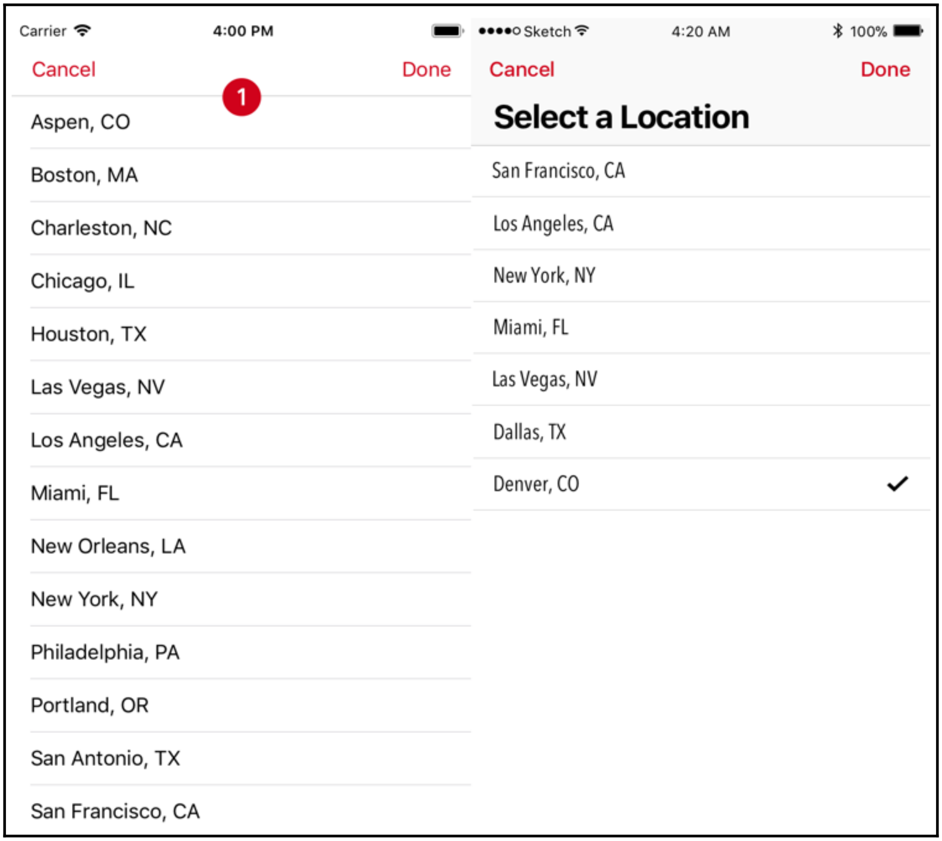
Tint: Default

Drawing: ☑ Opaque
□ Hidden
☑ Clears Graphics Context
□ Clip to Bounds
☑ Autoresize Subviews

Stretching: X 0, Y 0
Width 1, Height 1

☑ 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



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

Select a Location

San Francisco, CA

Los Angeles, CA

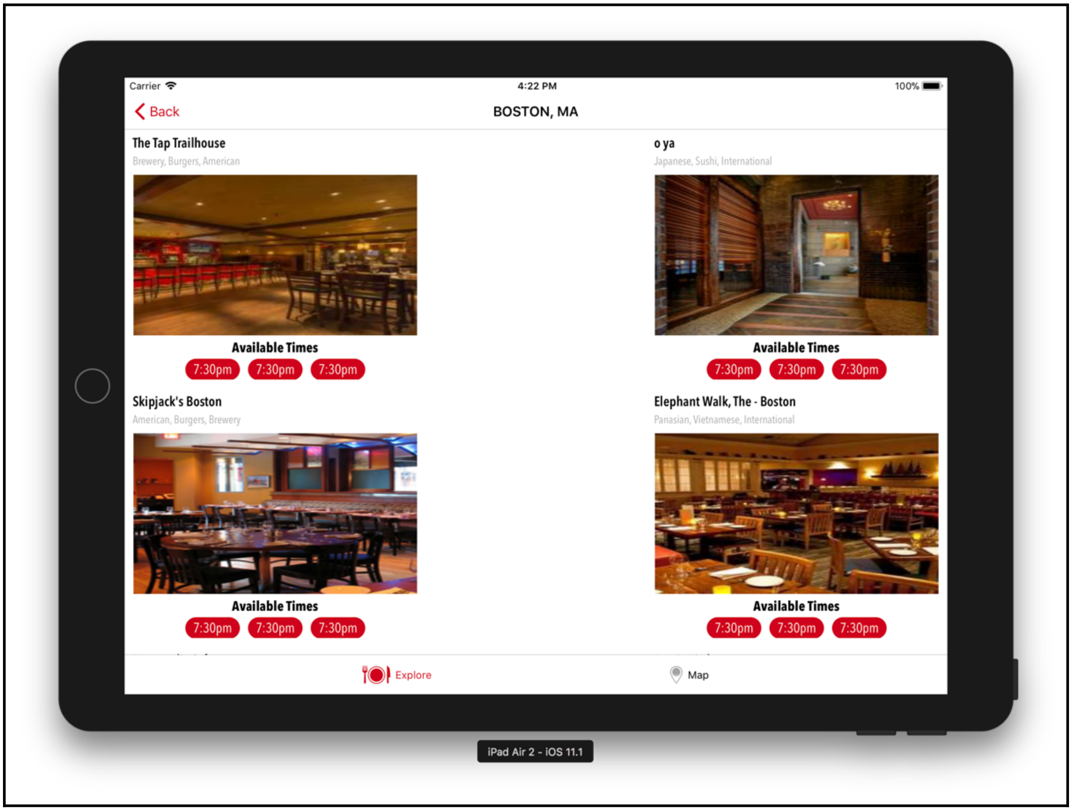
New York, NY

Miami, FL

Las Vegas, NV

Dallas, TX

Denver, CO ✓



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

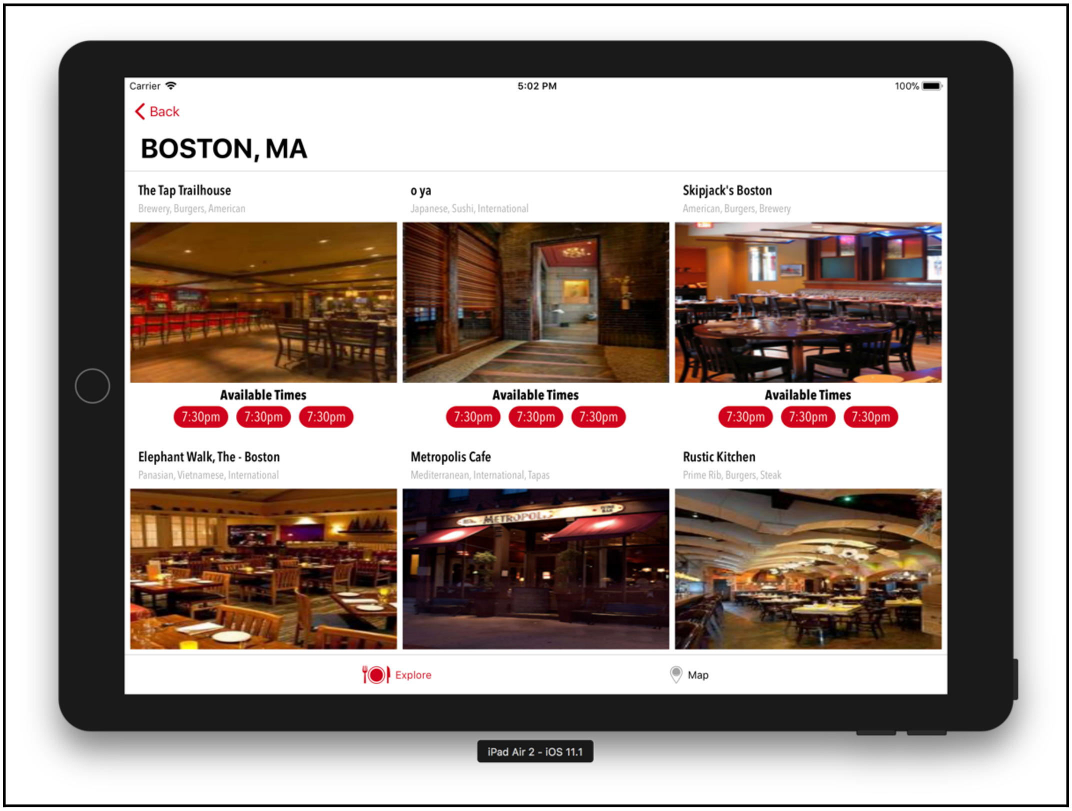
7:30pm 7:30pm 7:30pm

Elephant Walk, The - Boston

Panasian, Vietnamese, International

Available Times

7:30pm 7:30pm 7:30pm



Carrier 5:02 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

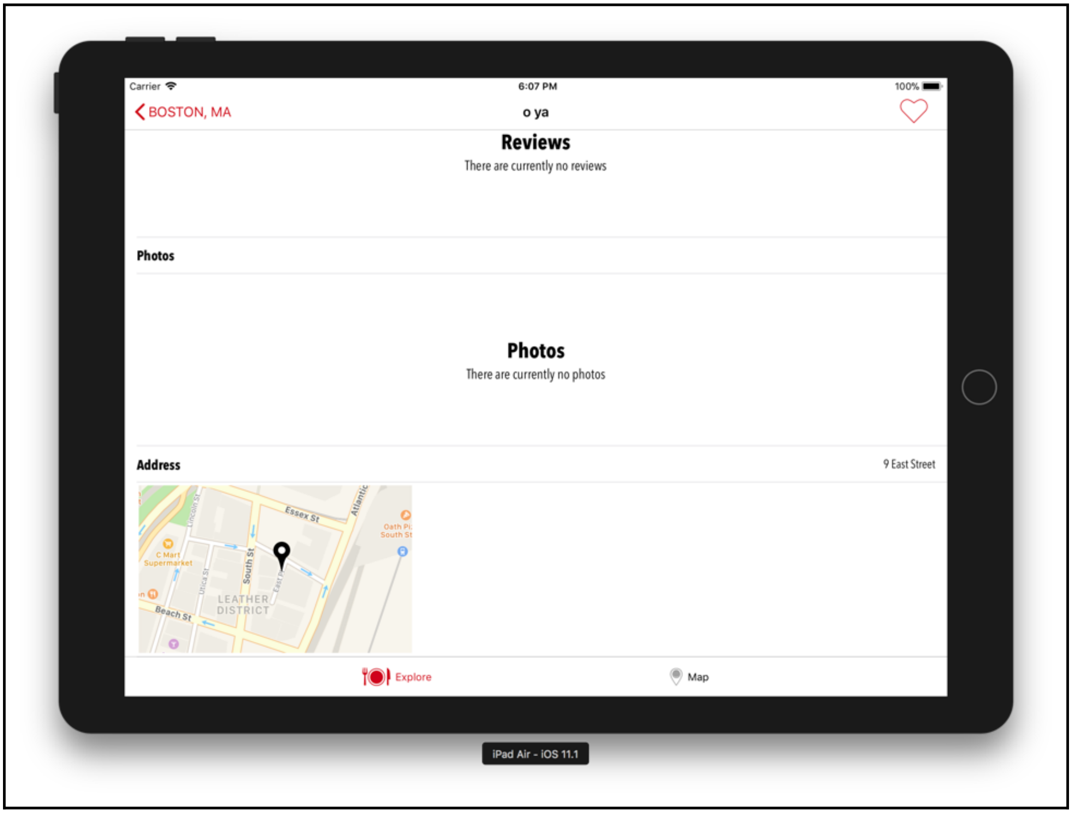
Available Times

7:30pm 7:30pm 7:30pm

Elephant Walk, The - Boston
Panasian, Vietnamese, International

Metropolis Cafe
Mediterranean, International, Tapas

Rustic Kitchen
Prime Rib, Burgers, Steak



o ya

Reviews

There are currently no reviews

Photos

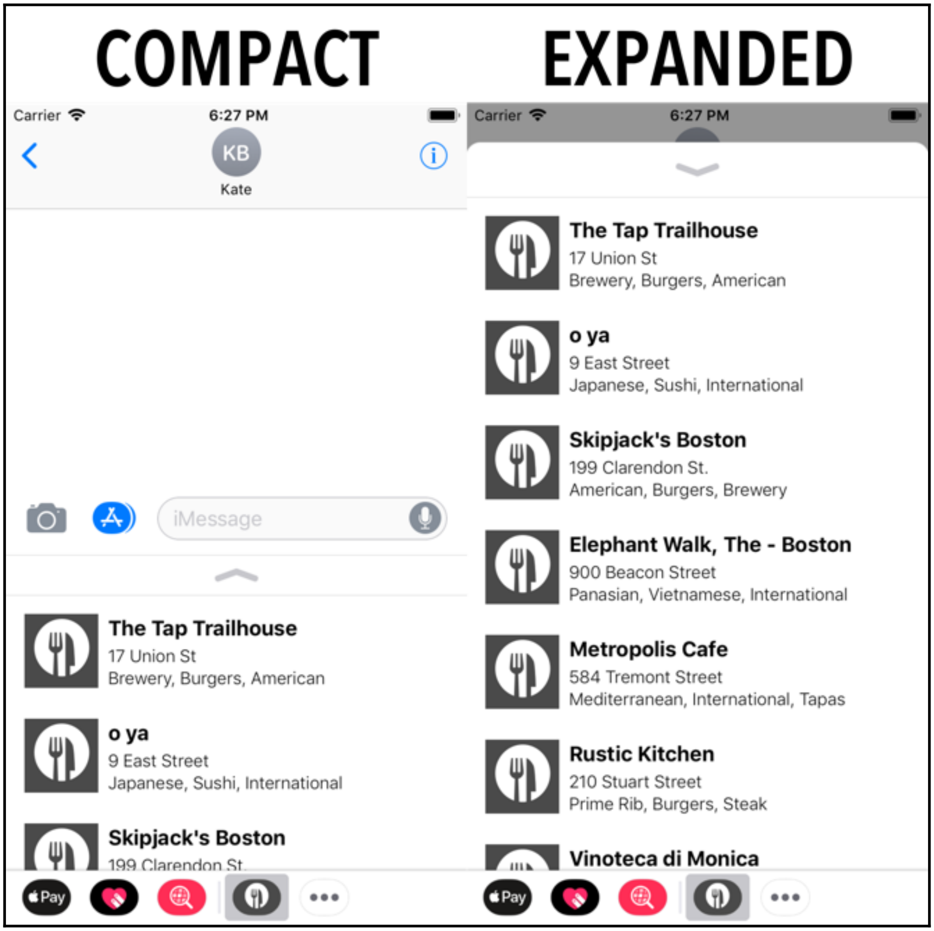
Photos

There are currently no photos

Address 9 East Street

iPad Air - iOS 11.1

Chapter 22: iMessages



COMPACT

EXPANDED

COMPACT:

The Tap Trailhouse
17 Union St
Brewery, Burgers, American

o ya
9 East Street
Japanese, Sushi, International

Skipjack's Boston
199 Clarendon St.

EXPANDED:

The Tap Trailhouse
17 Union St
Brewery, Burgers, American

o ya
9 East Street
Japanese, Sushi, International

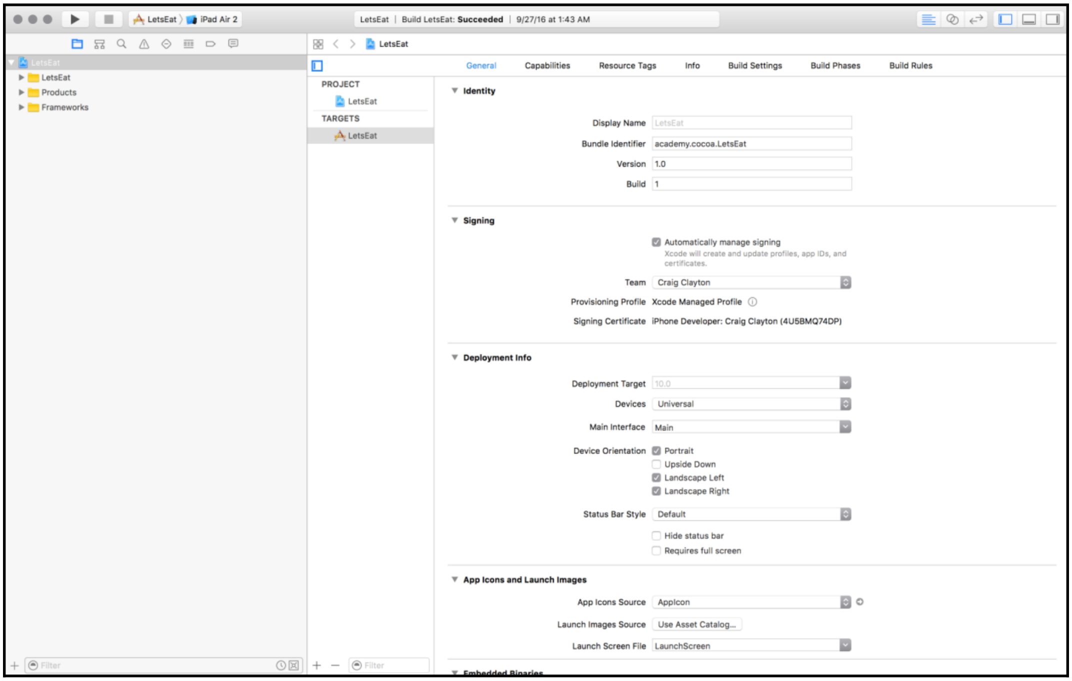
Skipjack's Boston
199 Clarendon St.
American, Burgers, Brewery

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



Identity

Display Name: LetsEat

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: 10.0

Devices: Universal

Main Interface: Main

Device Orientation:
☑ Portrait
□ Upside Down
☑ Landscape Left
☑ Landscape Right

Status Bar Style: Default

□ Hide status bar
□ Requires full screen

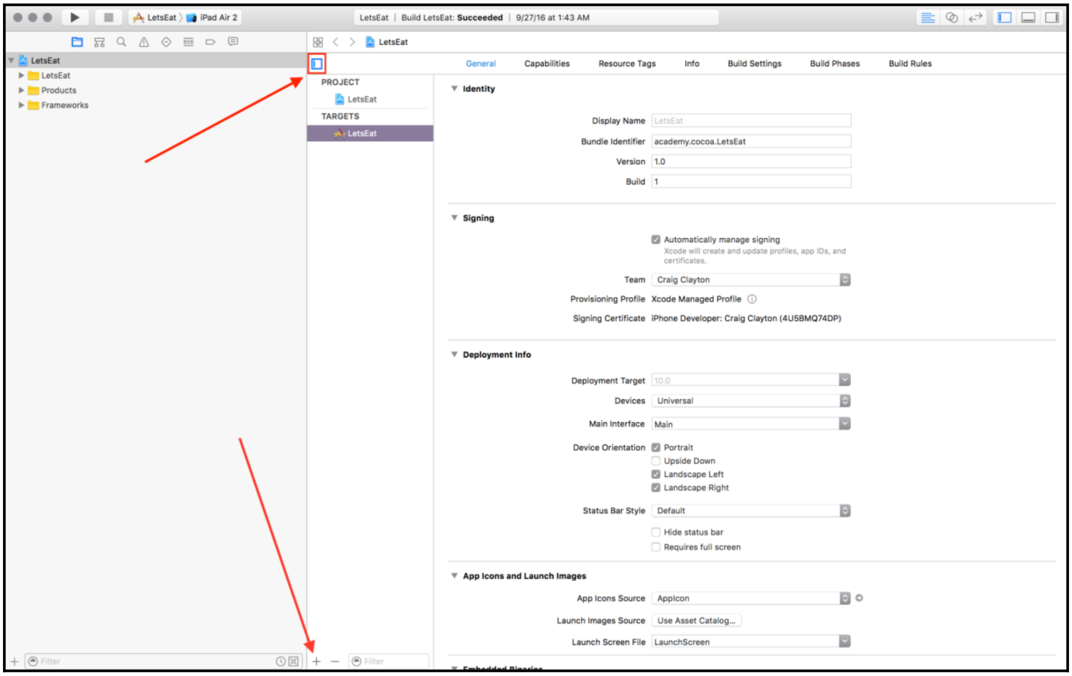
App Icons and Launch Images

App Icons Source: AppIcon

Launch Images Source: Use Asset Catalog...

Launch Screen File: LaunchScreen

Embedded Binaries



Identity

Display Name LetsEat
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)

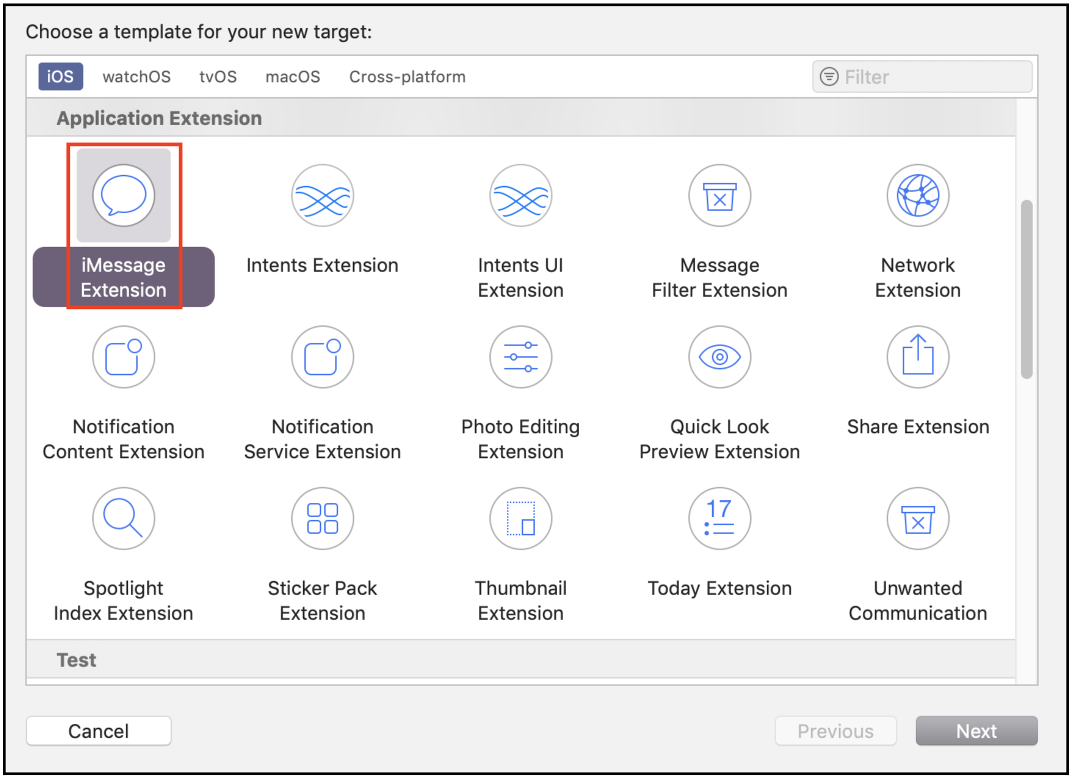
Deployment Info

Deployment Target 10.0
Devices Universal
Main Interface Main
Device Orientation ☑ Portrait
□ Upside Down
☑ Landscape Left
☑ Landscape Right
Status Bar Style Default
□ Hide status bar
□ Requires full screen

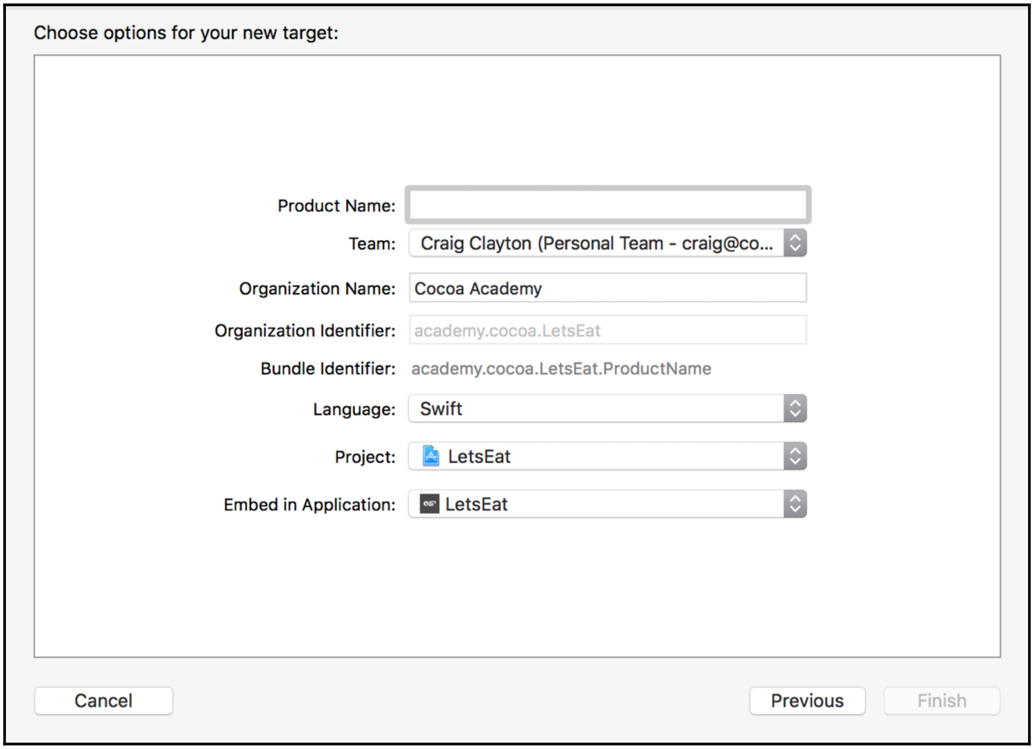
App Icons and Launch Images

App Icons Source AppIcon
Launch Images Source Use Asset Catalog...
Launch Screen File LaunchScreen

Embedded Binaries



Choose a template for your new target:



Choose options for your new target:

Product Name:

Team: Craig Clayton (Personal Team - craig@co...

Organization Name: Cocoa Academy

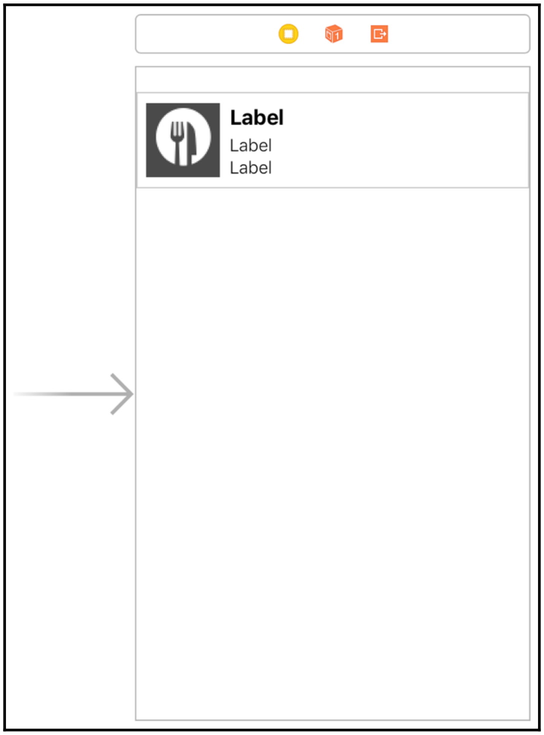
Organization Identifier: academy.cocoa.LetsEat

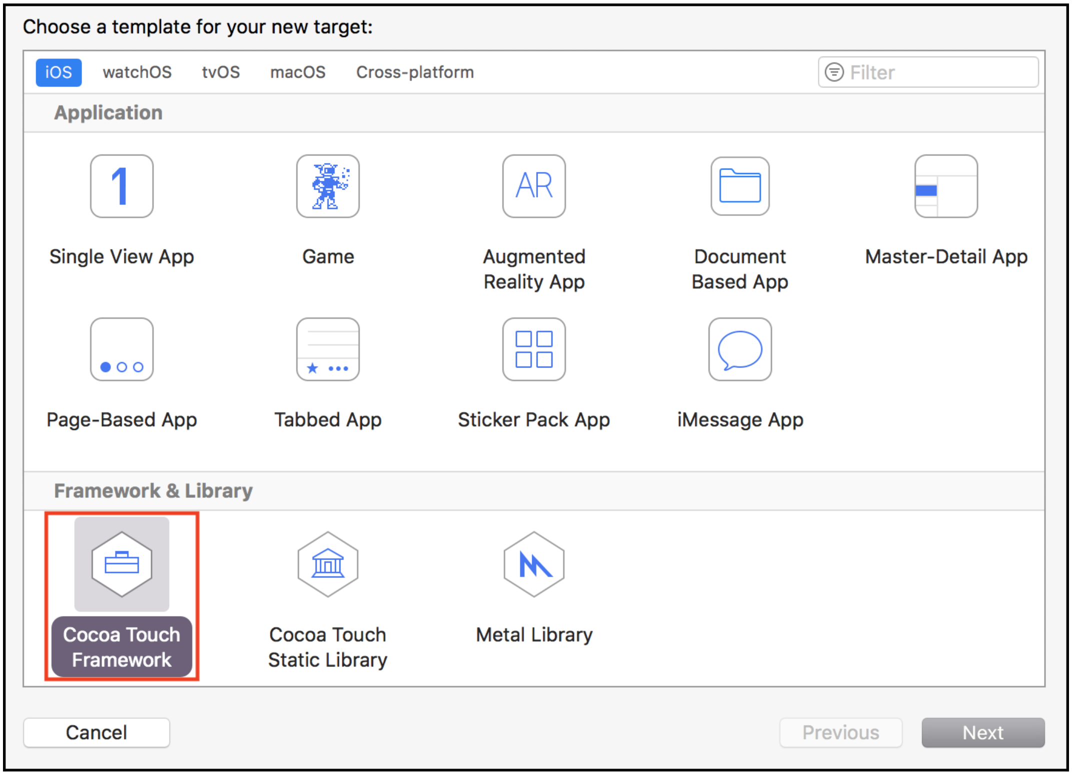
Bundle Identifier: academy.cocoa.LetsEat.ProductName

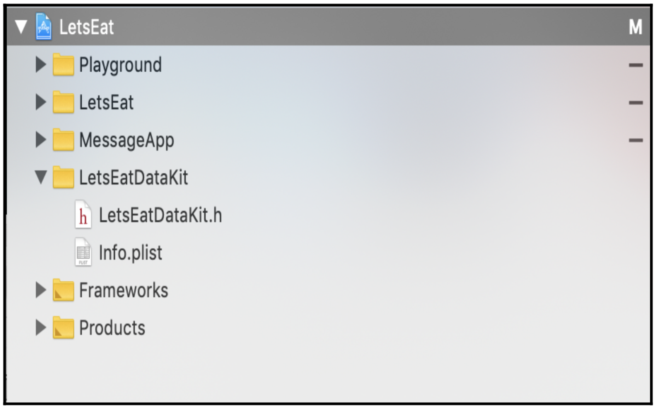
Language: Swift

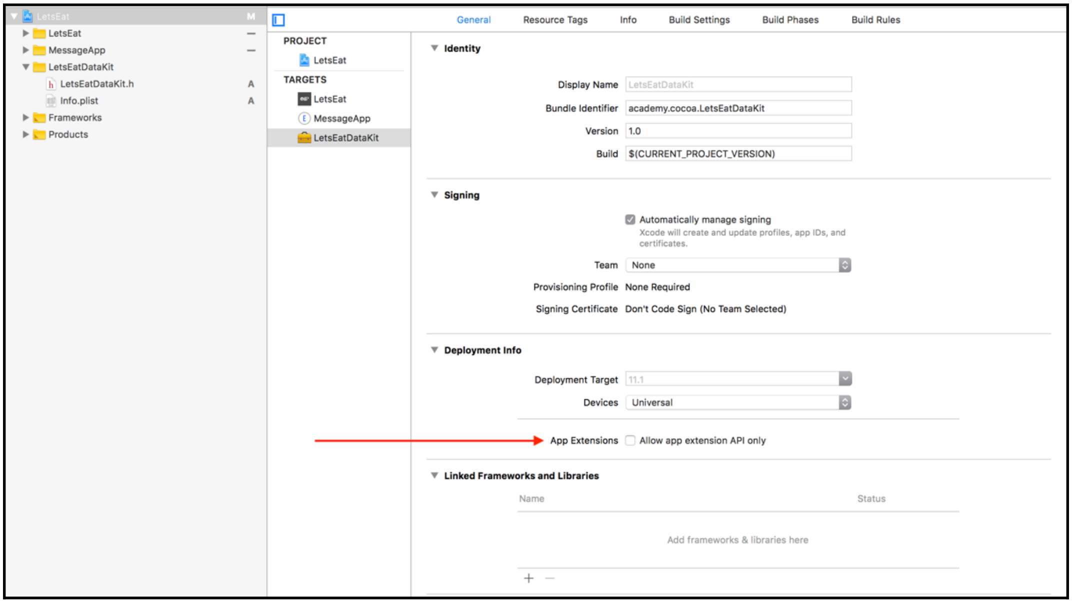
Project: LetsEat

Embed in Application: LetsEat


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
Playground
LetsEat
MessageApp
LetsEatDataKit
LetsEatDataKit.h
Info.plist
Frameworks
Products



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

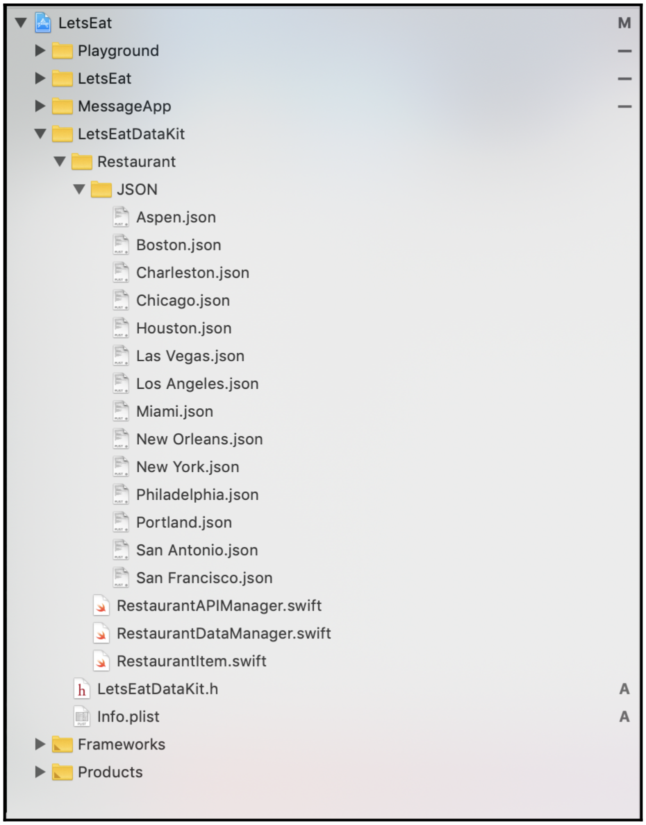
Deployment Target 11.1

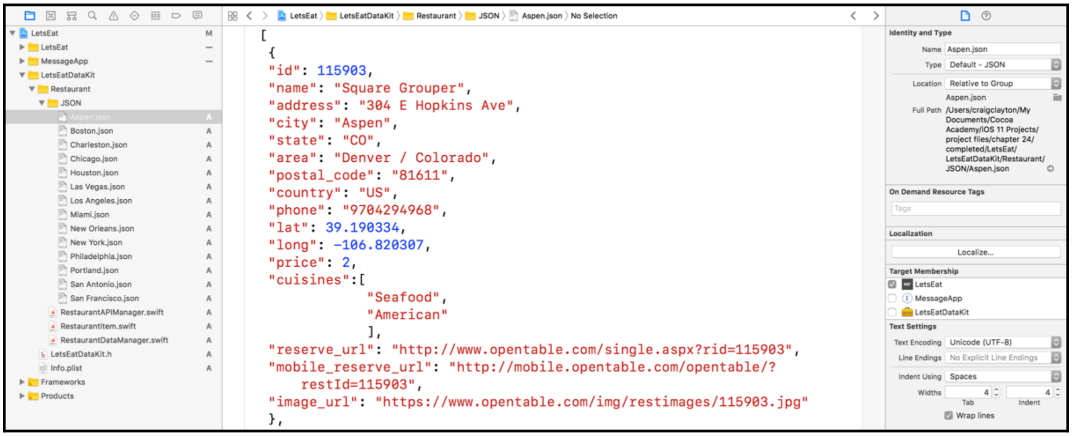
Devices Universal

App Extensions □ Allow app extension API only

Linked Frameworks and Libraries

Name	Status
Add frameworks & libraries here	





```
[
 {
 "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

Tags

Localization

Localize...

Target Membership

☑ LetsEat

□ MessageApp

□ LetsEatDataKit

Text Settings

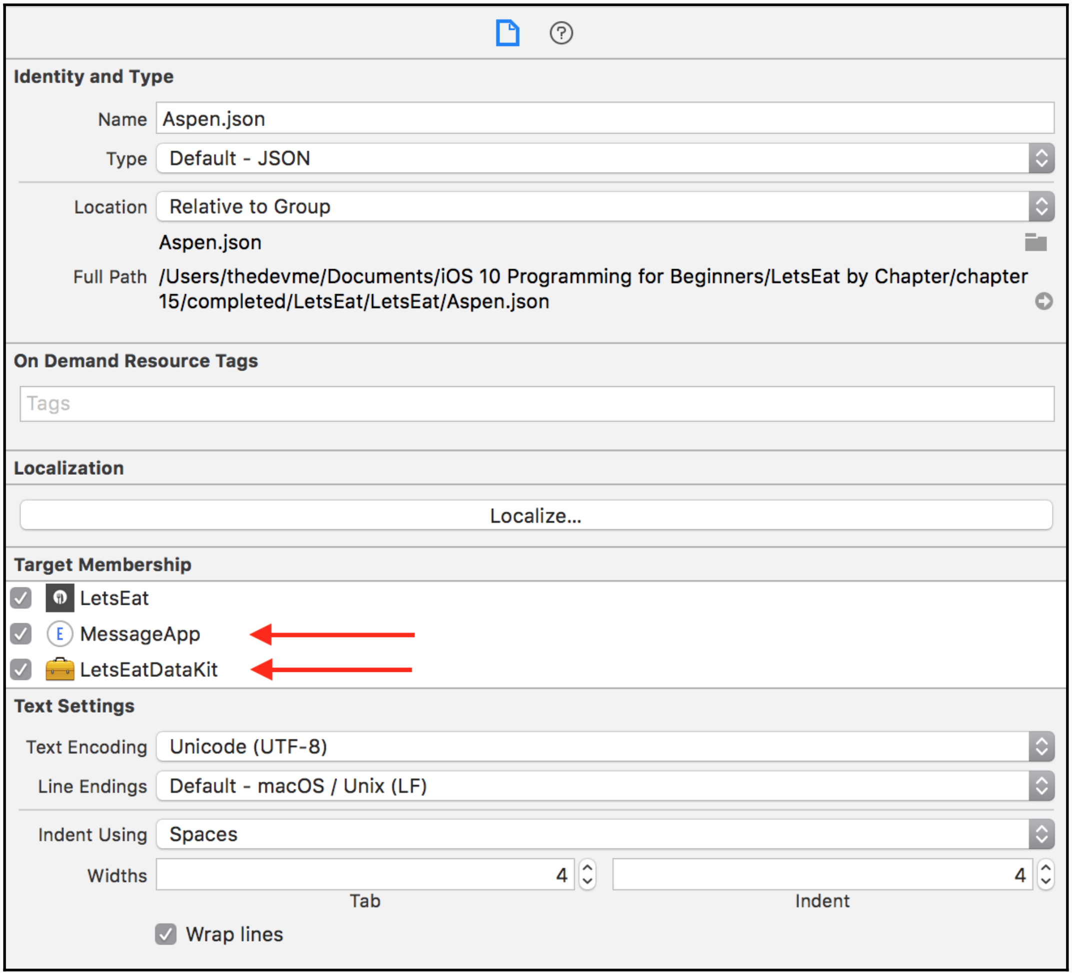
Text Encoding: Unicode (UTF-8)

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: 4 Tab, 4 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

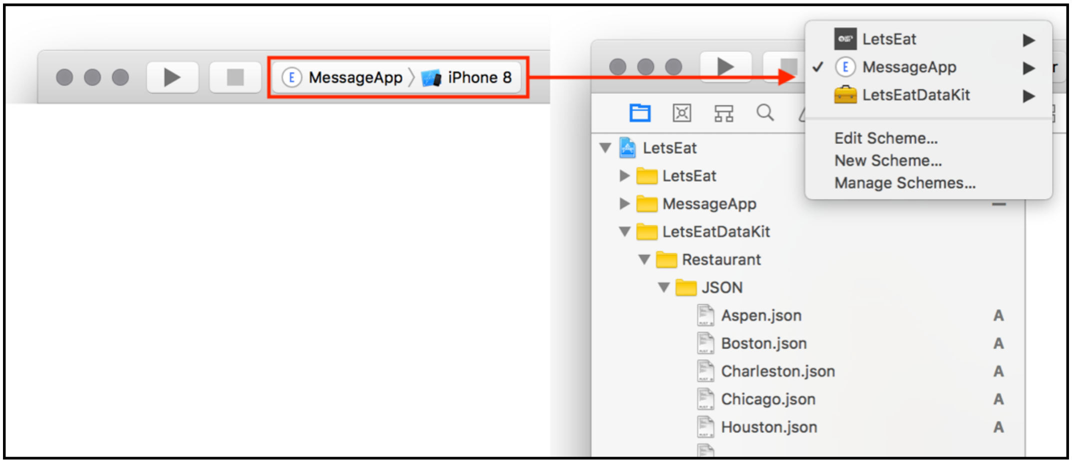
Text Encoding: Unicode (UTF-8)

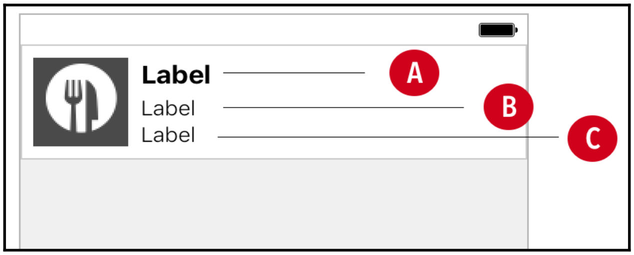
Line Endings: Default - macOS / Unix (LF)

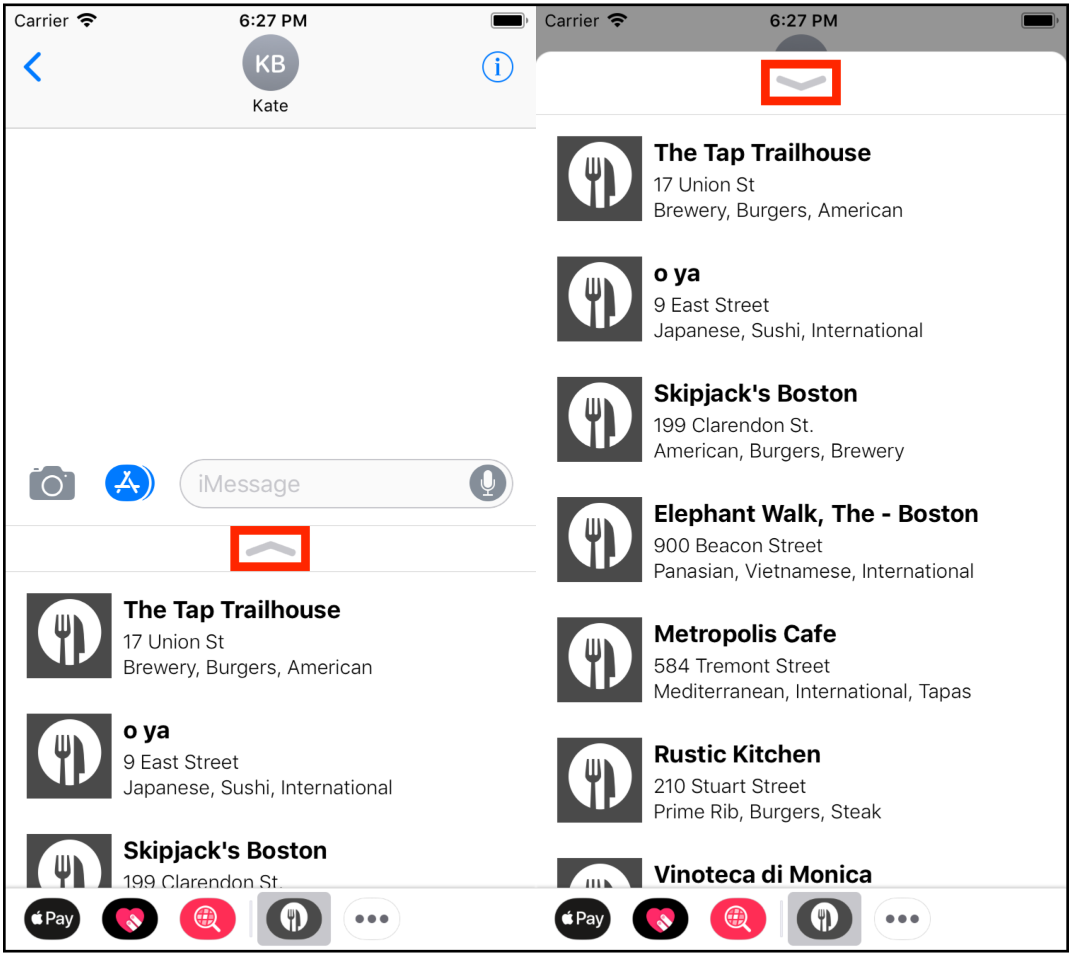
Indent Using: Spaces

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



The Tap Trailhouse
17 Union St
Brewery, Burgers, American

o ya
9 East Street
Japanese, Sushi, International

Skipjack's Boston
199 Clarendon St.

The Tap Trailhouse
17 Union St
Brewery, Burgers, American

o ya
9 East Street
Japanese, Sushi, International

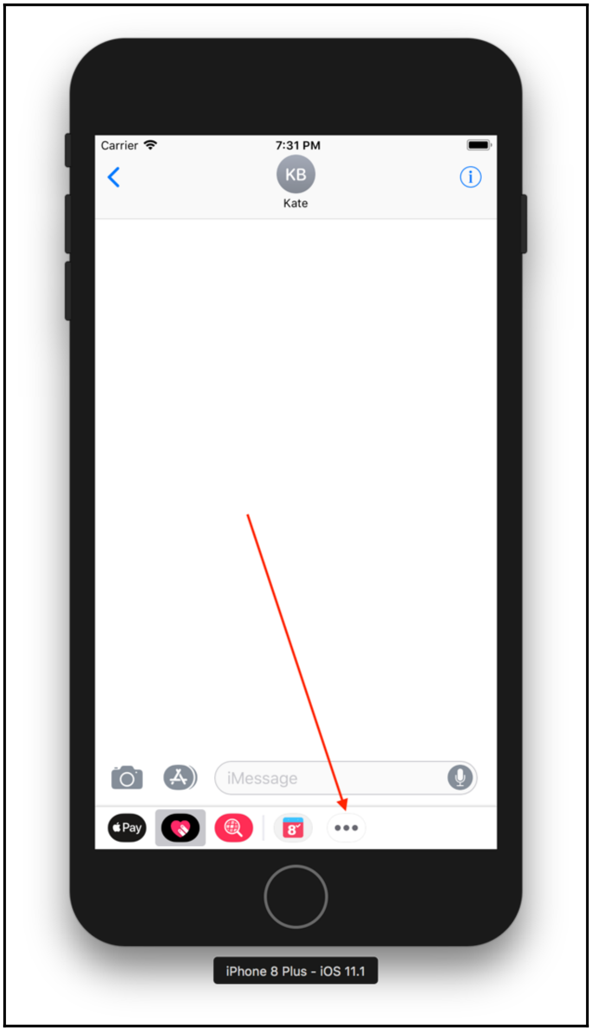
Skipjack's Boston
199 Clarendon St.
American, Burgers, Brewery

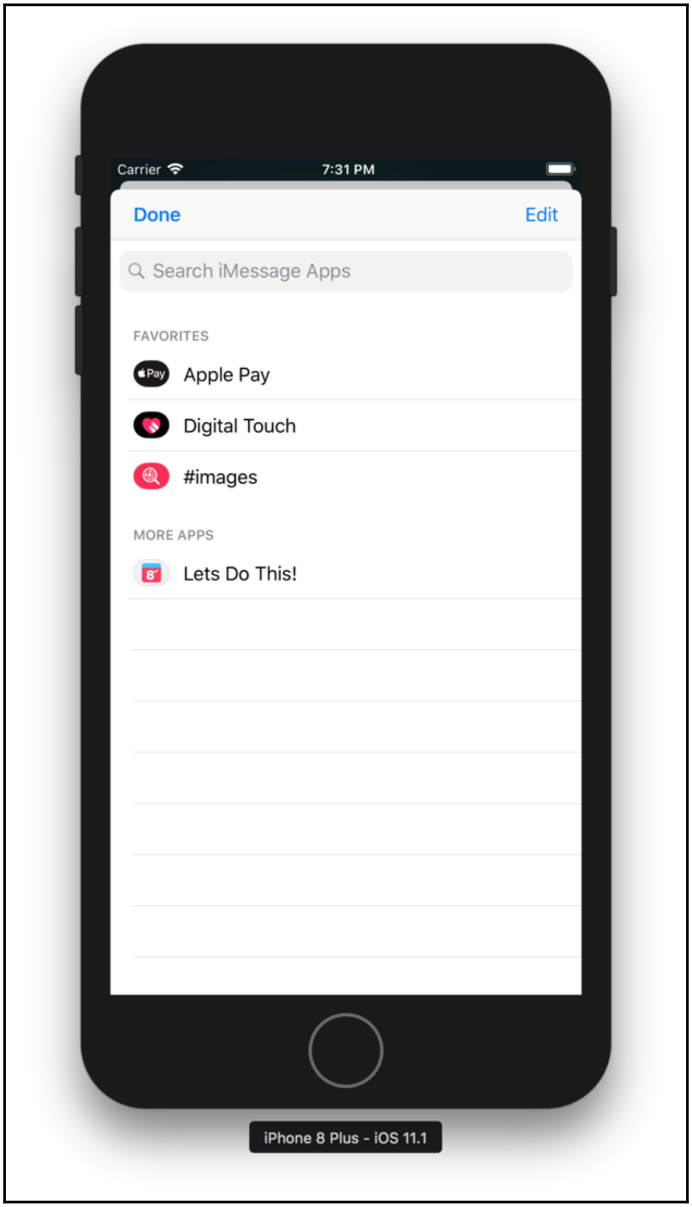
Elephant Walk, The - Boston
900 Beacon Street
Panasian, Vietnamese, International

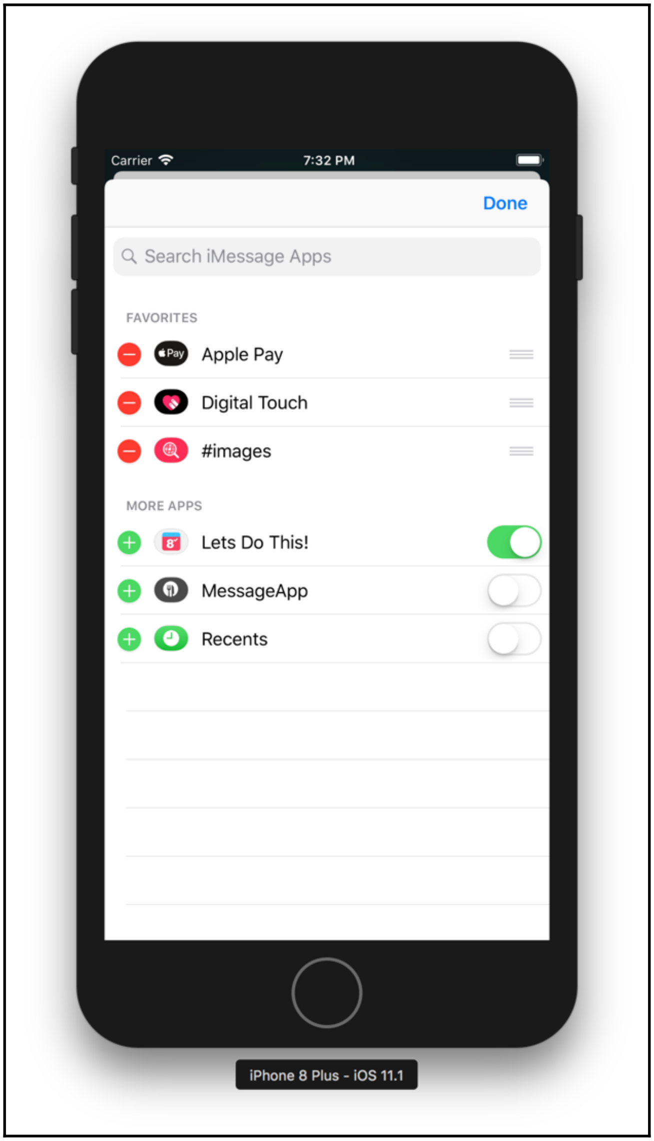
Metropolis Cafe
584 Tremont Street
Mediterranean, International, Tapas

Rustic Kitchen
210 Stuart Street
Prime Rib, Burgers, Steak

Vinoteca di Monica


Carrier
7:31 PM
KB
Kate
iMessage
Pay
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

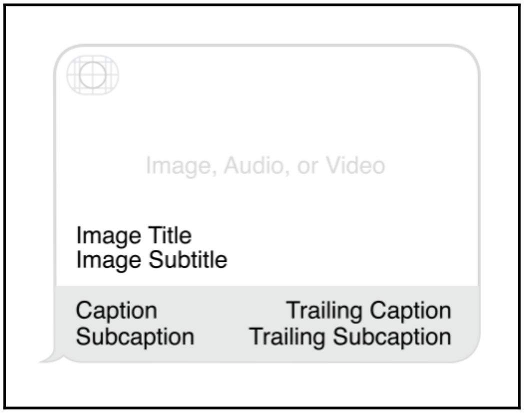
FAVORITES

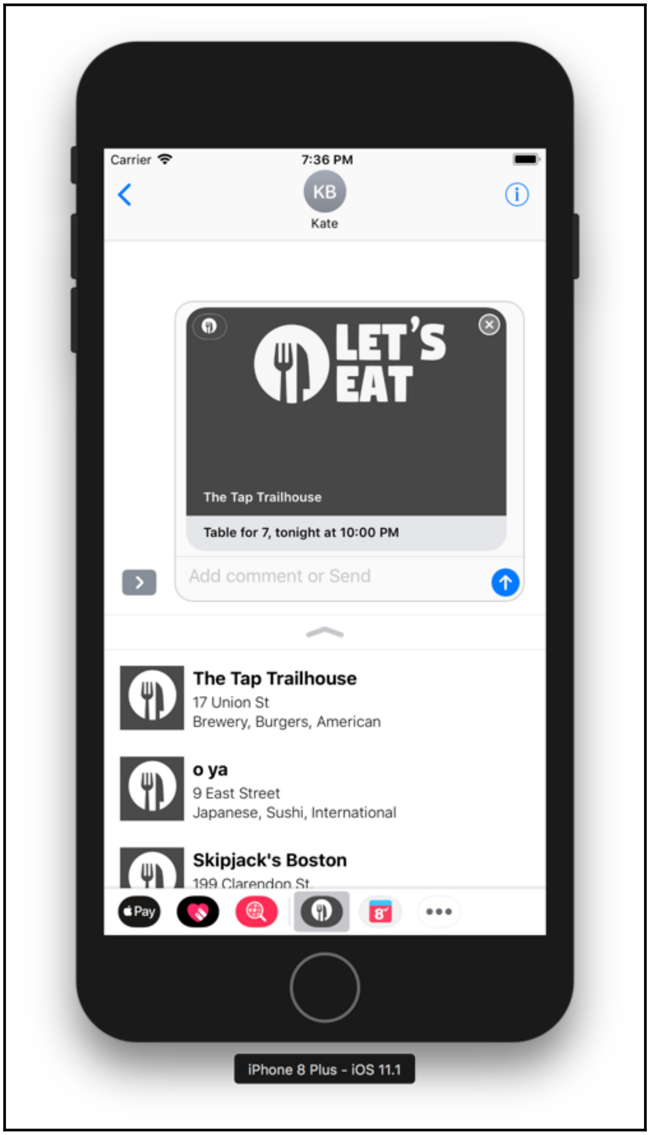
- Apple Pay
- Digital Touch
- #images

MORE APPS

- Lets Do This! (on)
- MessageApp (off)
- Recents (off)

iPhone 8 Plus - iOS 11.1


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



The Tap Trailhouse

Table for 7, tonight at 10:00 PM

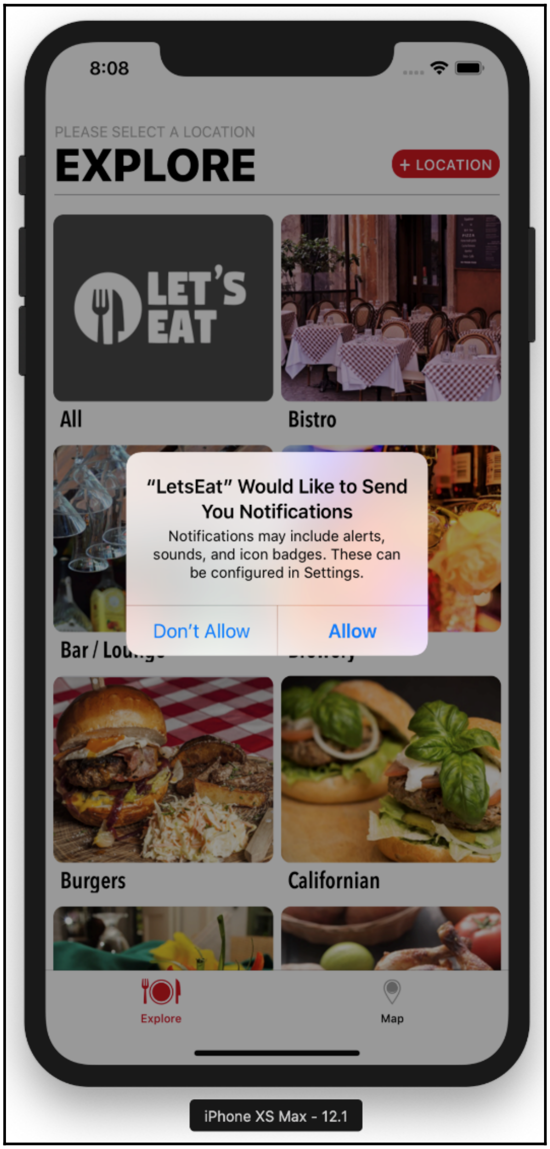
Add comment or Send

The Tap Trailhouse
17 Union St
Brewery, Burgers, American

o ya
9 East Street
Japanese, Sushi, International

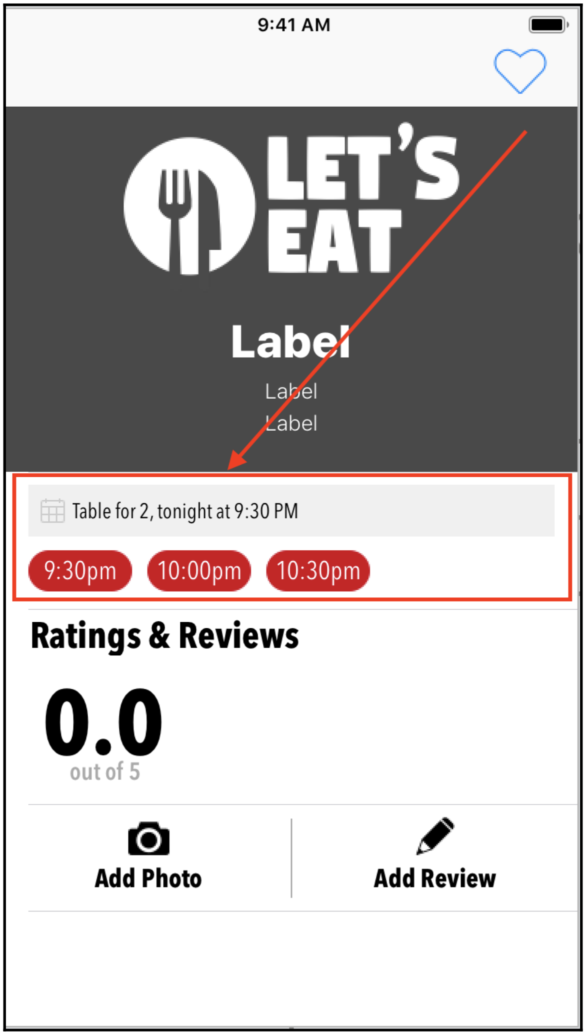
Skipjack's Boston
199 Clarendon St

Chapter 23: Notifications

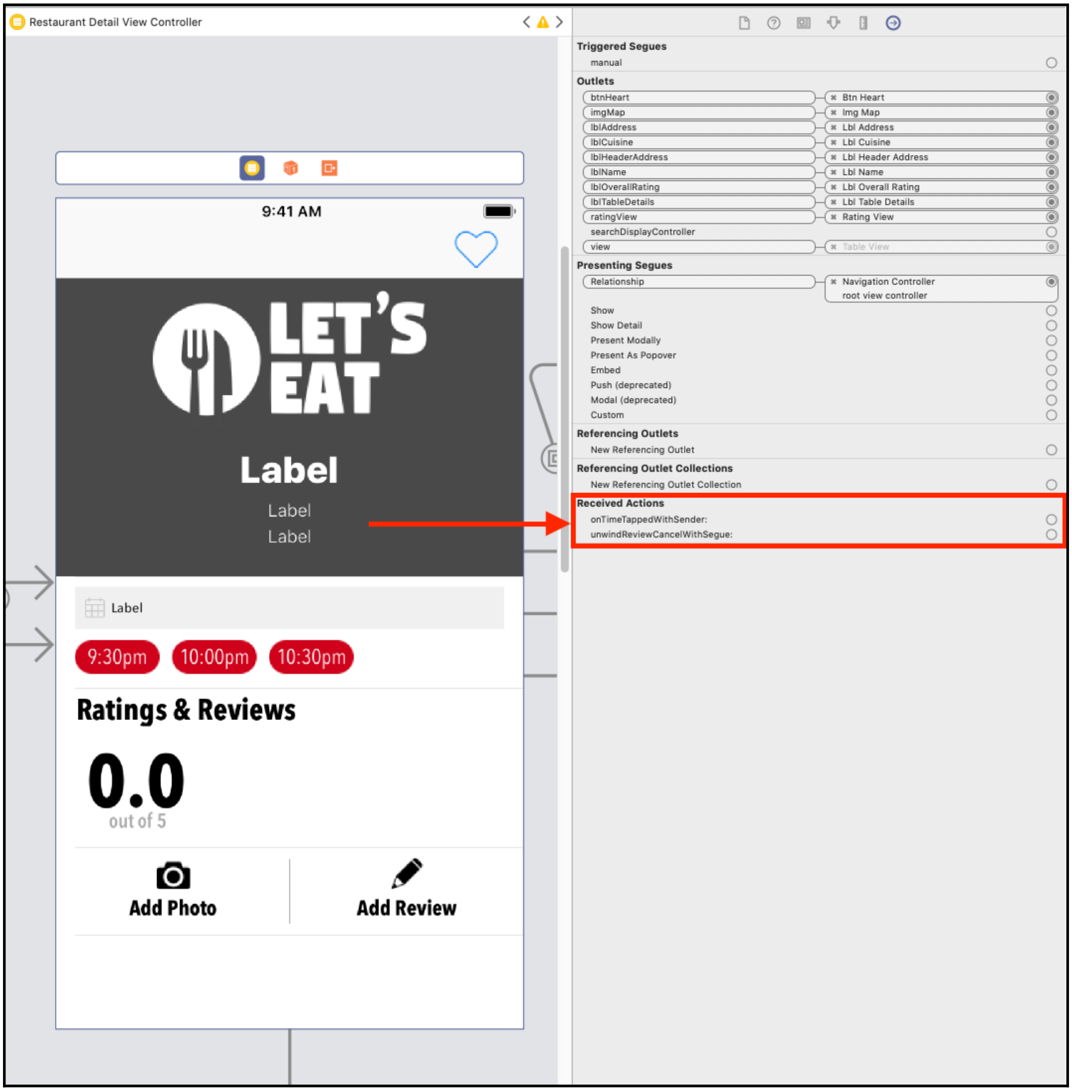


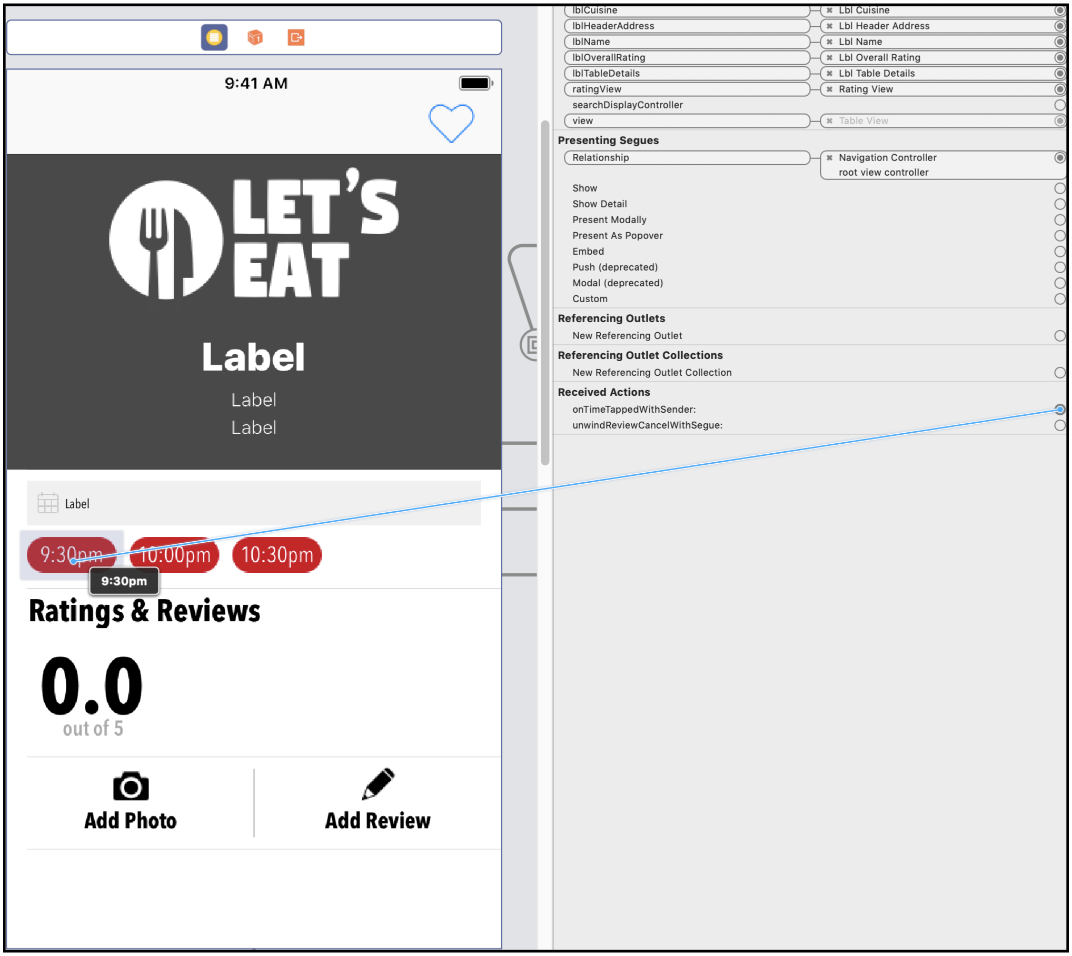
"LetsEat" Would Like to Send You Notifications

Notifications may include alerts, sounds, and icon badges. These can be configured in Settings.



9:41 AM
LET'S EAT
Label
Label
Label
Table for 2, tonight at 9:30 PM
9:30pm
10:00pm
10:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review


Restaurant Detail View Controller
9:41 AM
LET'S EAT
Label
Label
Label
Label
9:30pm
10:00pm
10:30pm
Ratings & Reviews
0.0
out of 5
Add Photo
Add Review
Triggered Segues
manual
Outlets
btnHeart — Btn Heart
imgMap — Img Map
lblAddress — Lbl Address
lblCuisine — Lbl Cuisine
lblHeaderAddress — Lbl Header Address
lblName — Lbl Name
lblOverallRating — Lbl Overall Rating
lblTableDetails — Lbl Table Details
ratingView — Rating View
searchDisplayController
view — Table 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
Received Actions
onTimeTappedWithSender:
unwindReviewCancelWithSegue:



lblCuisine — Lbl Cuisine
lblHeaderAddress — Lbl Header Address
lblName — Lbl Name
lblOverallRating — Lbl Overall Rating
lblTableDetails — Lbl Table Details
ratingView — Rating View
searchDisplayController
view — Table 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

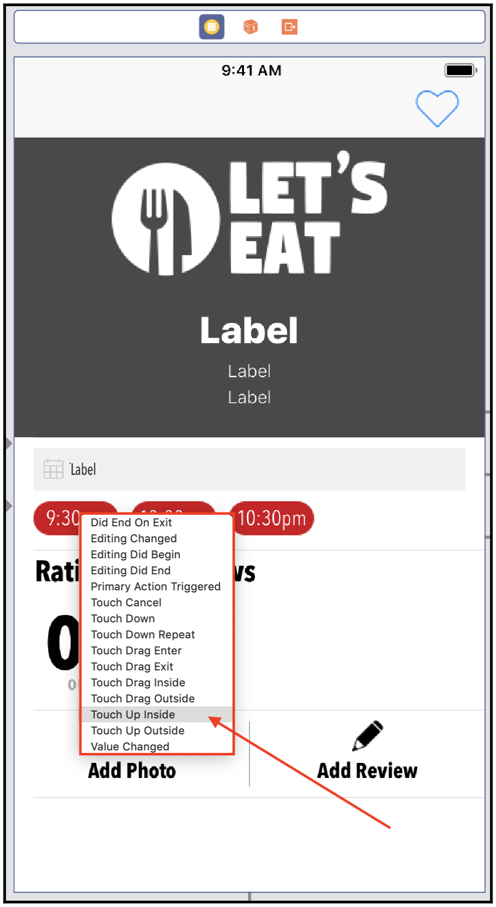
New Referencing Outlet

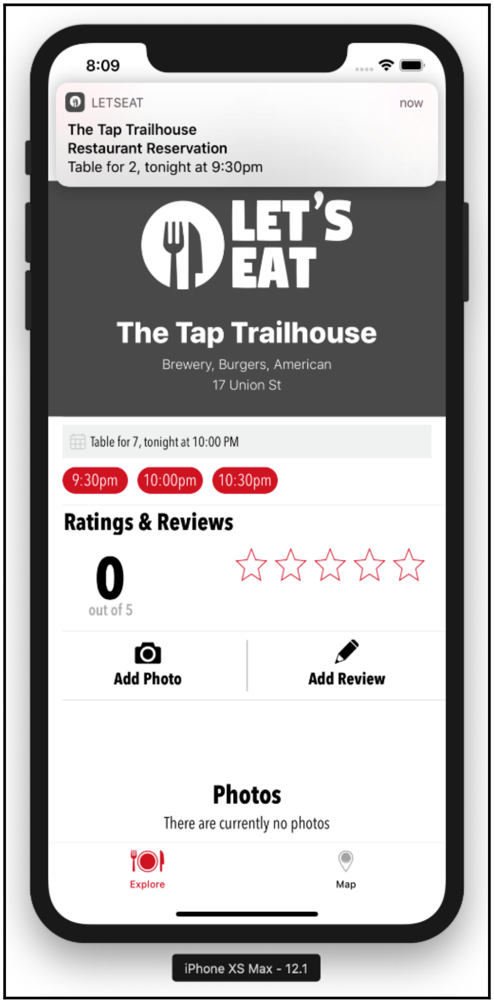
Referencing Outlet Collections

New Referencing Outlet Collection

Received Actions

onTimeTappedWithSender:
unwindReviewCancelWithSegue:


9:41 AM
LET'S EAT
Label
Label
Label
Label
9:30
10:30pm
Rati
vs
0
0
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 Photo
Add Review



8:09

LETSEAT now

The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm

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

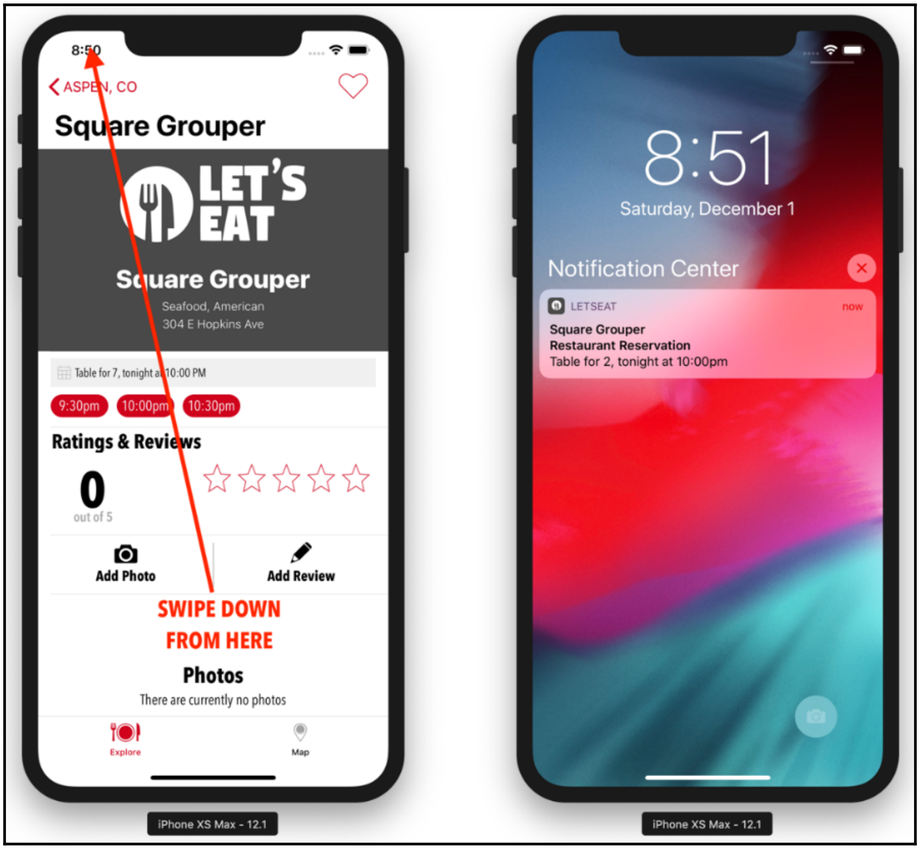
0
out of 5

☆ ☆ ☆ ☆ ☆

Add Photo **Add Review**

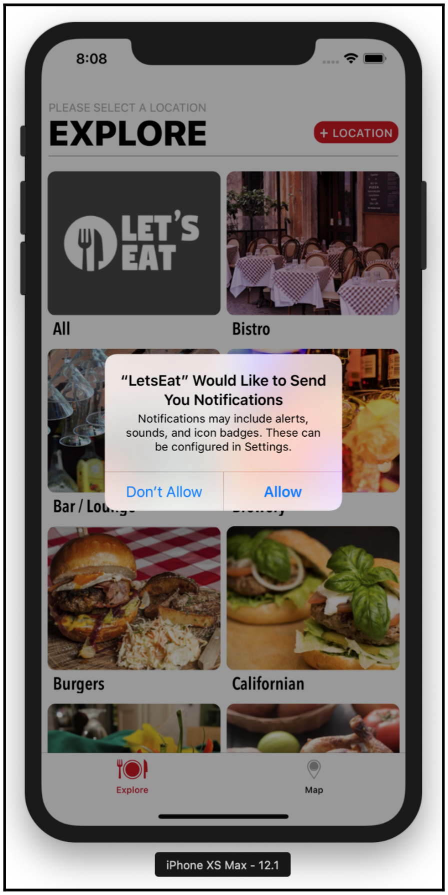
Photos

There are currently no photos



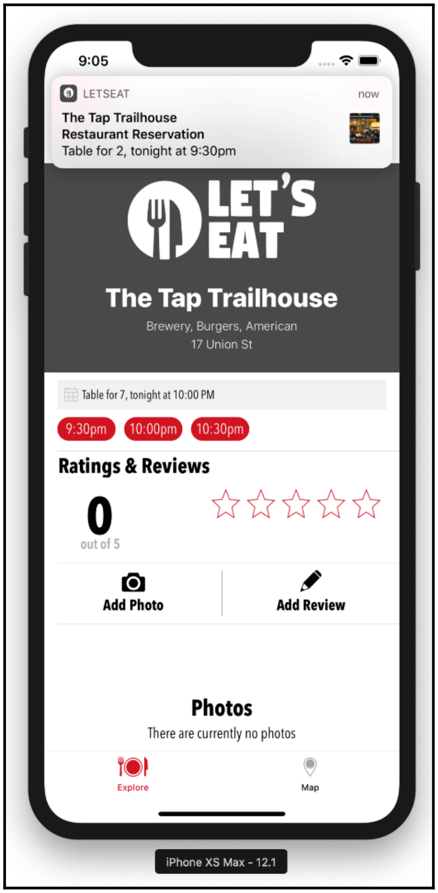
SWIPE DOWN
FROM HERE

Square Grouper
Restaurant Reservation
Table for 2, tonight at 10:00pm



“LetsEat” Would Like to Send You Notifications

Notifications may include alerts, sounds, and icon badges. These can be configured in Settings.

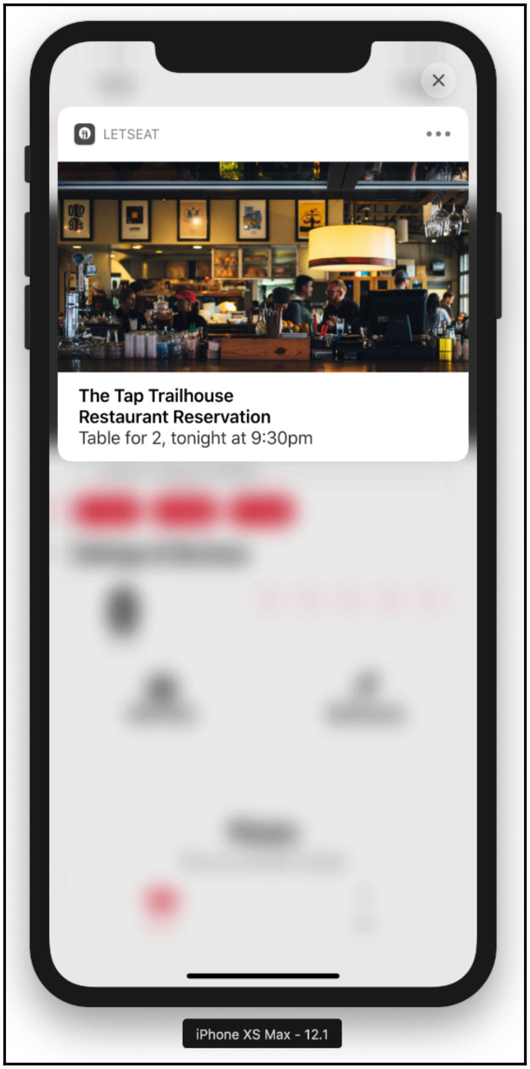


The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm

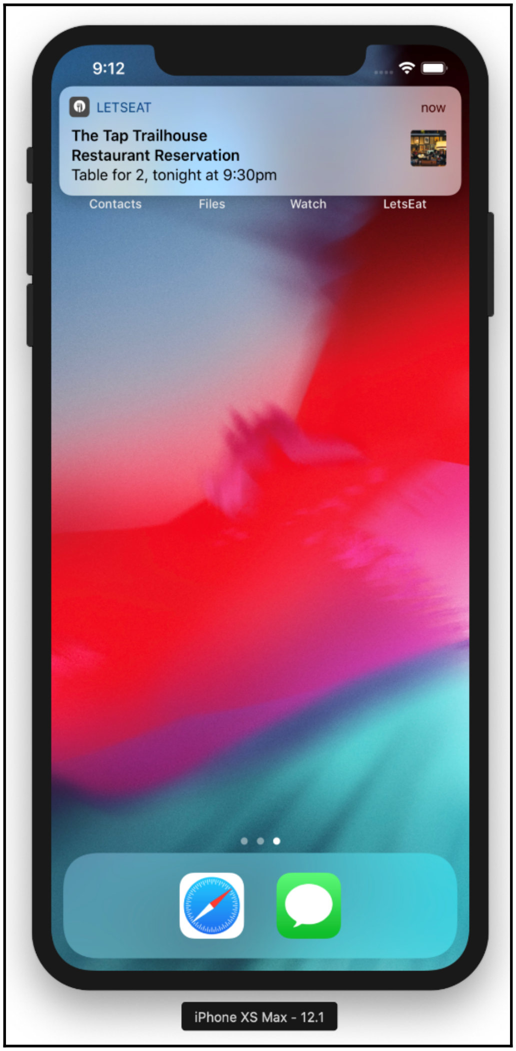
The Tap Trailhouse
Brewery, Burgers, American
17 Union St

Table for 7, tonight at 10:00 PM

There are currently no photos

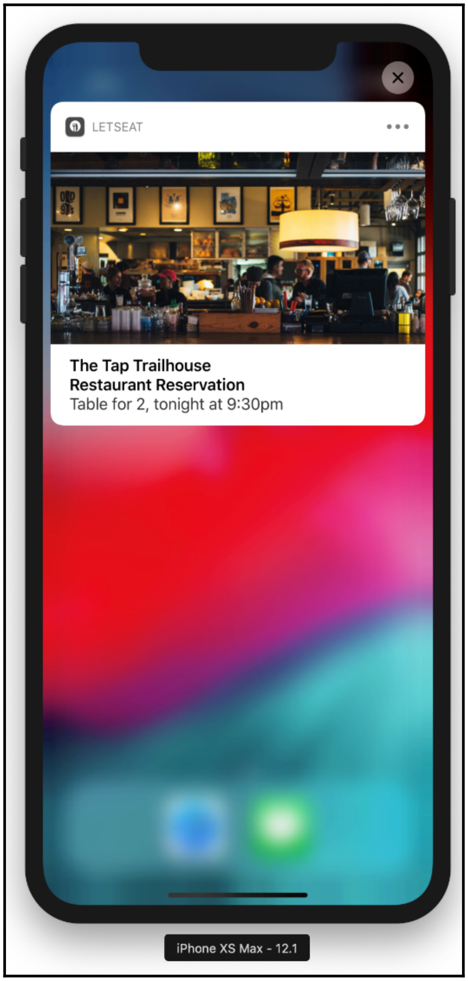


The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm

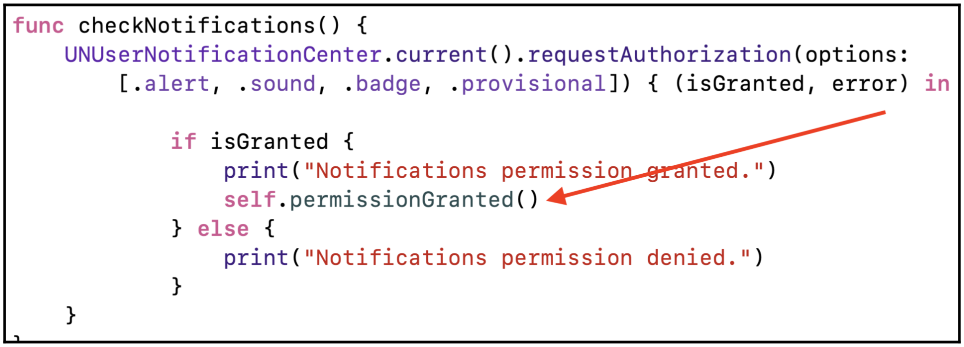


LETSEAT now

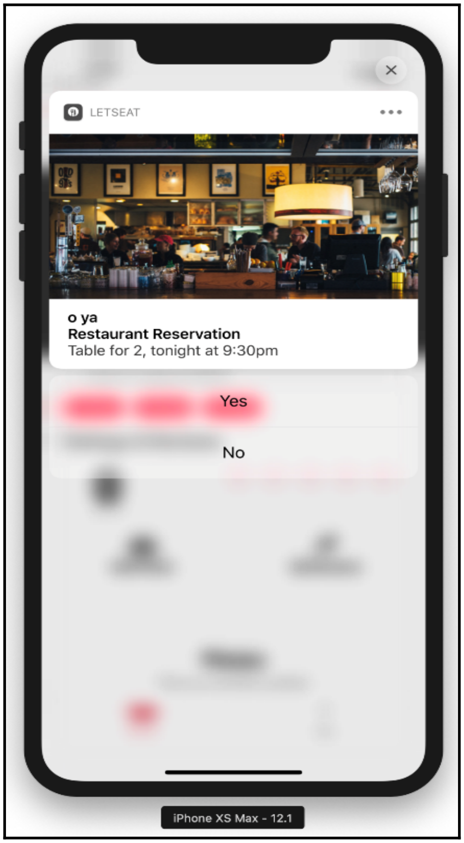
The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm



The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm



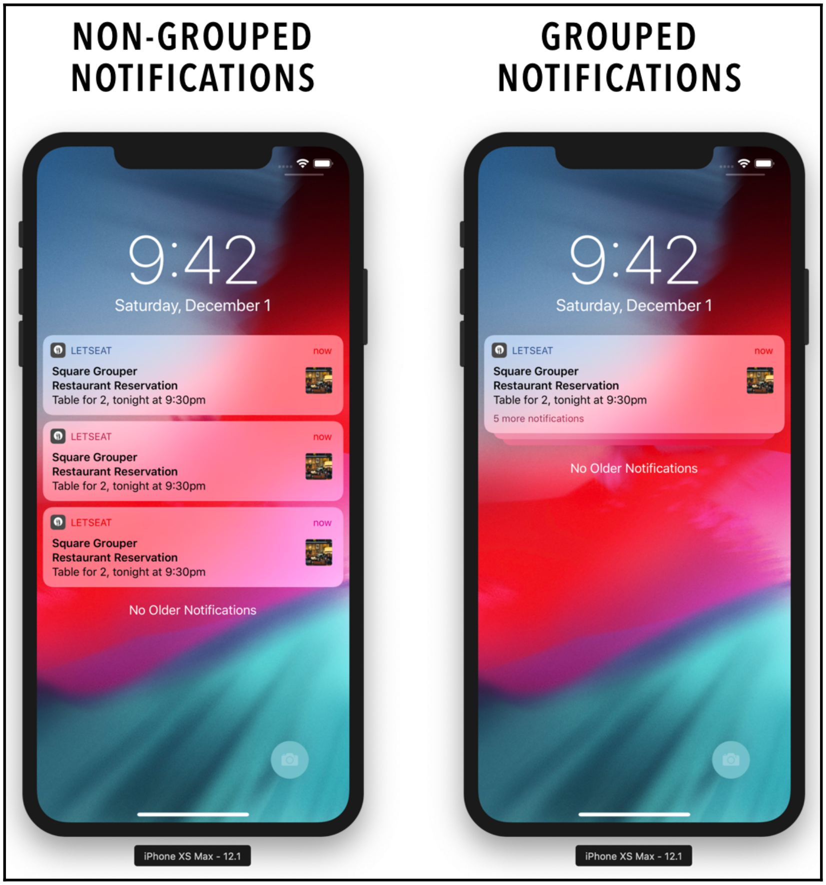
```
func checkNotifications() {
    UNUserNotificationCenter.current().requestAuthorization(options:
        [.alert, .sound, .badge, .provisional]) { (isGranted, error) in

            if isGranted {
                print("Notifications permission granted.")
                self.permissionGranted()
            } else {
                print("Notifications permission denied.")
            }
    }
}
```



o ya

Restaurant Reservation

Table for 2, tonight at 9:30pm



NON-GROUPED NOTIFICATIONS

LETSEAT now
Square Grouper
Restaurant Reservation
Table for 2, tonight at 9:30pm

LETSEAT now
Square Grouper
Restaurant Reservation
Table for 2, tonight at 9:30pm

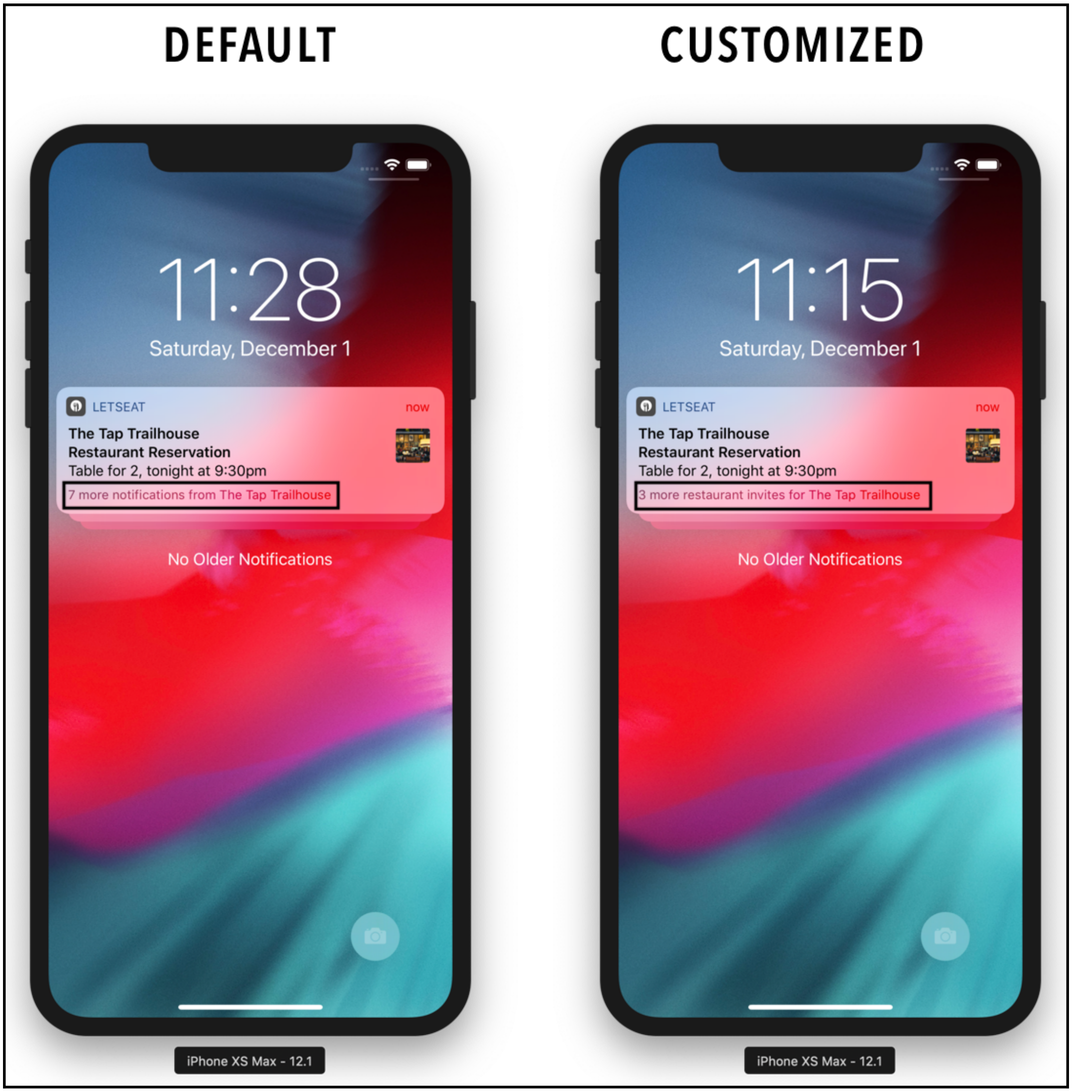
LETSEAT now
Square Grouper
Restaurant Reservation
Table for 2, tonight at 9:30pm

No Older Notifications

GROUPED NOTIFICATIONS

LETSEAT now
Square Grouper
Restaurant Reservation
Table for 2, tonight at 9:30pm
5 more notifications

No Older Notifications



DEFAULT

LETSEAT now

The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm
7 more notifications from The Tap Trailhouse

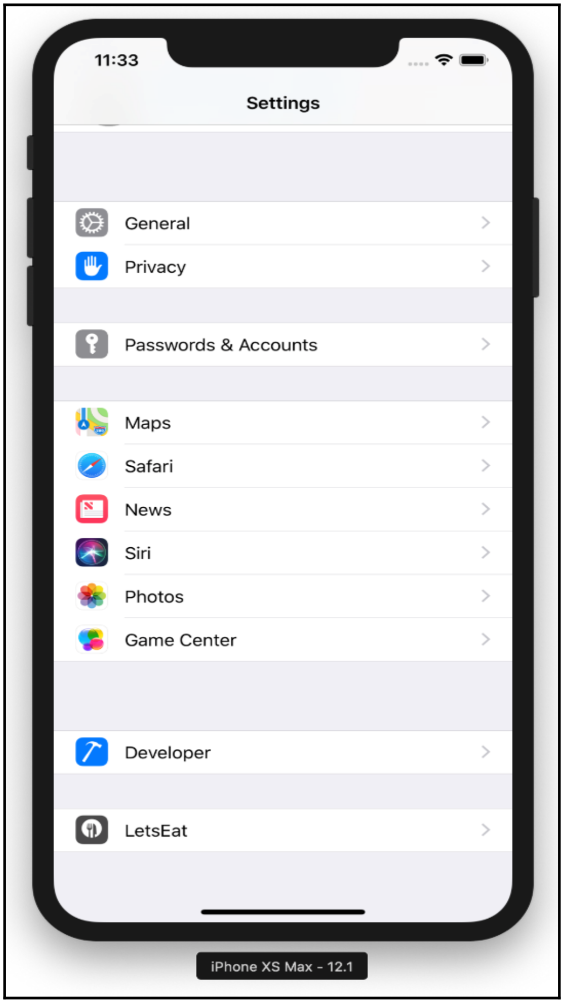
No Older Notifications

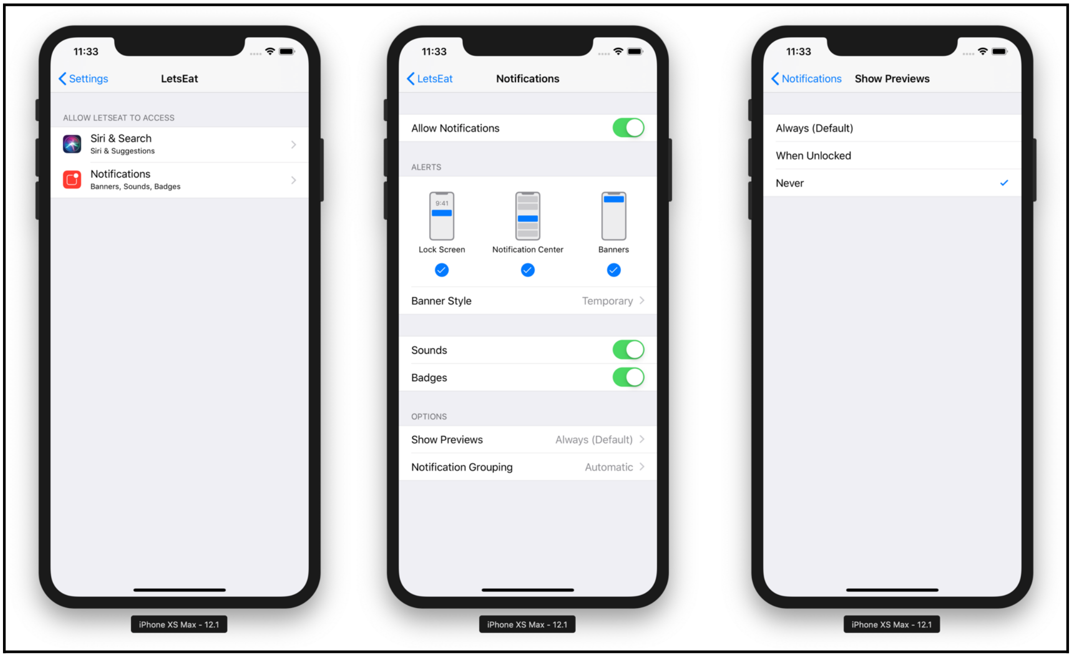
CUSTOMIZED

LETSEAT now

The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm
3 more restaurant invites for The Tap Trailhouse

No Older Notifications


11:33
Settings
General
Privacy
Passwords & Accounts
Maps
Safari
News
Siri
Photos
Game Center
Developer
LetsEat
iPhone XS Max - 12.1



11:33

Settings LetsEat

ALLOW LETSEAT TO ACCESS

Siri & Search
Siri & Suggestions

Notifications
Banners, Sounds, Badges

iPhone XS Max - 12.1

11:33

LetsEat Notifications

Allow Notifications

ALERTS

Lock Screen ✓
Notification Center ✓
Banners ✓

Banner Style Temporary

Sounds

Badges

OPTIONS

Show Previews Always (Default)

Notification Grouping Automatic

iPhone XS Max - 12.1

11:33

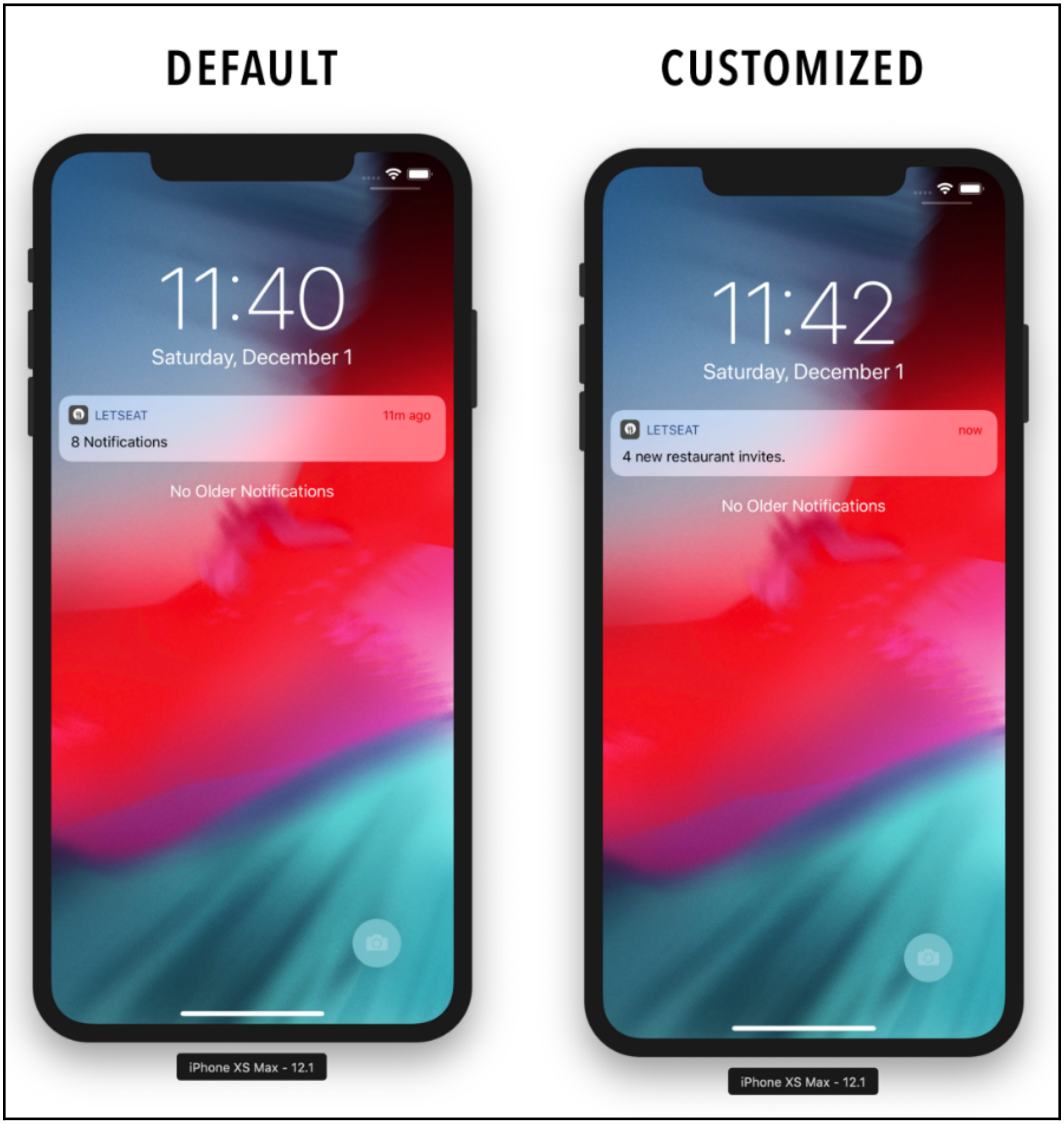
Notifications Show Previews

Always (Default)

When Unlocked

Never ✓

iPhone XS Max - 12.1



DEFAULT

11:40

Saturday, December 1

LETSEAT 11m ago

8 Notifications

No Older Notifications

CUSTOMIZED

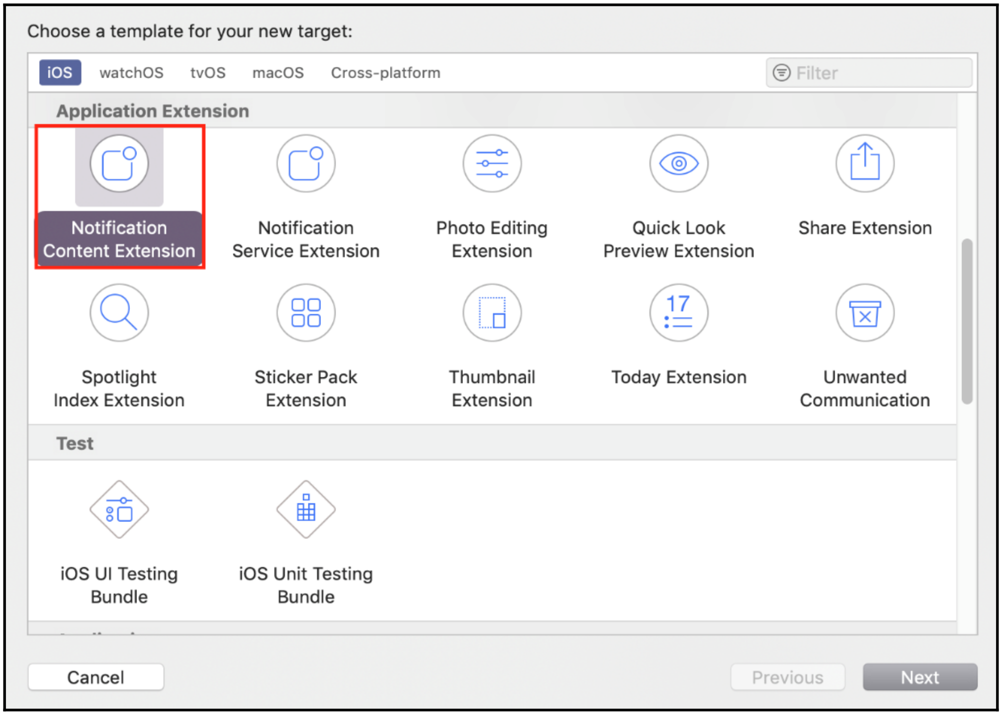
11:42

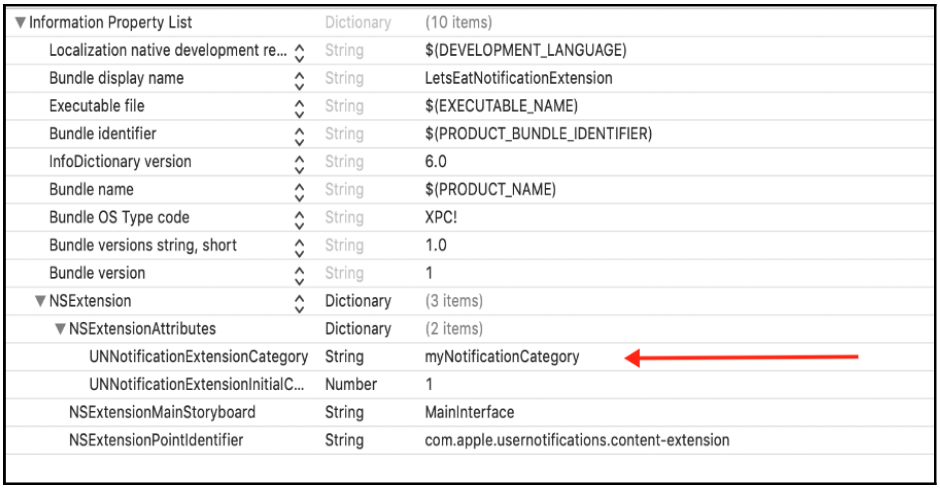
Saturday, December 1

LETSEAT now

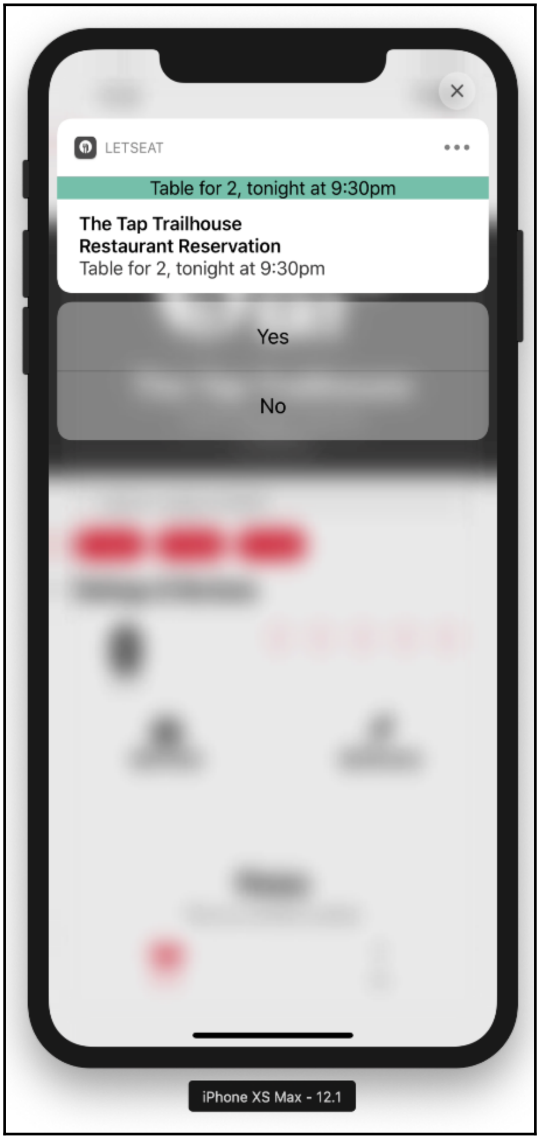
4 new restaurant invites.

No Older Notifications





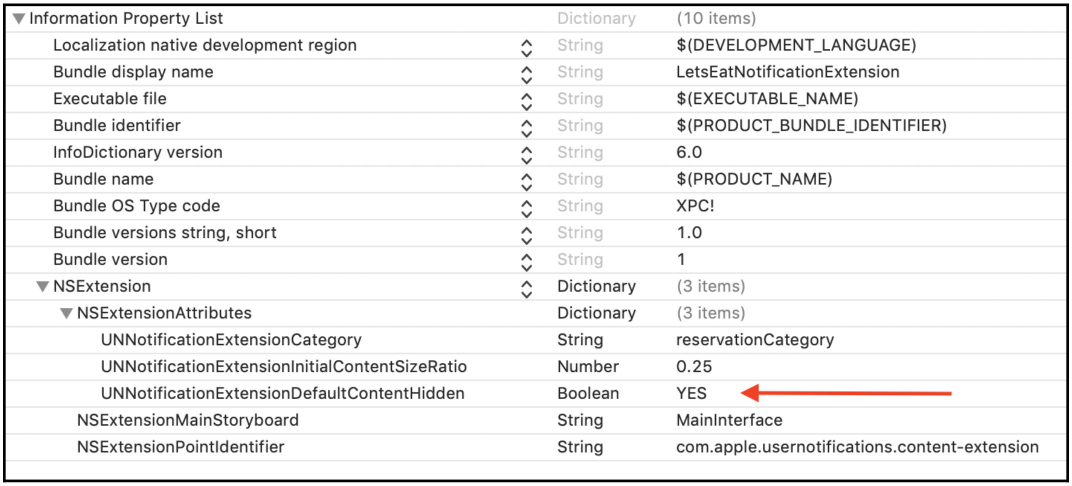
▼Information Property List	Dictionary	(10 items)
Localization native development re...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	LetsEatNotificationExtension
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
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



LETSEAT

Table for 2, tonight at 9:30pm

The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm



▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	LetsEatNotificationExtension
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

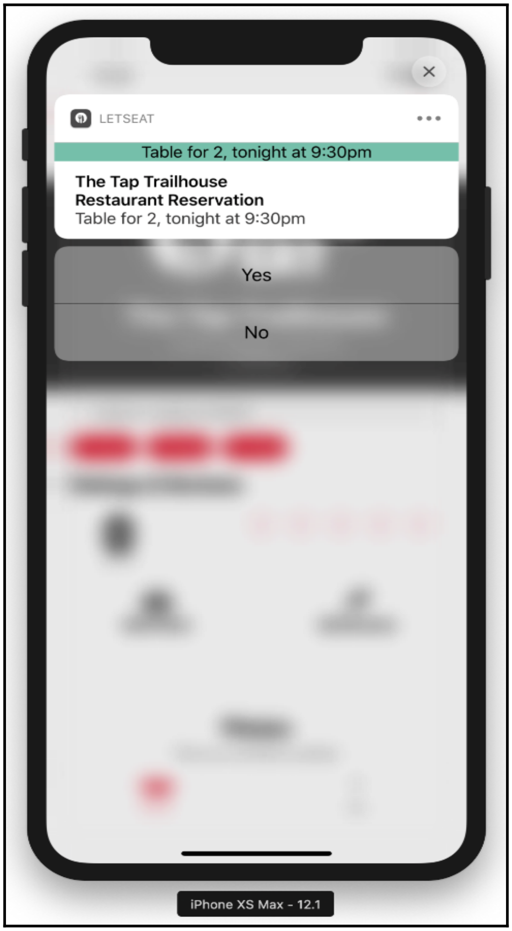
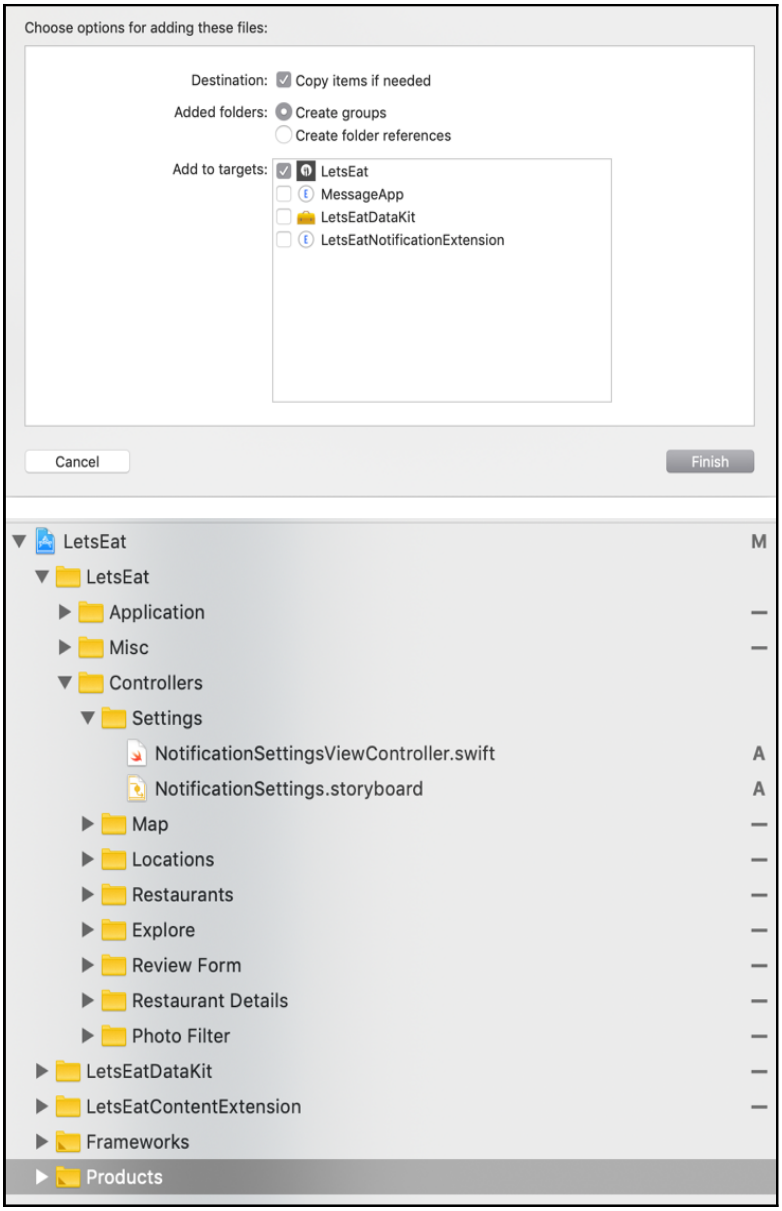


Table for 2, tonight at 9:30pm

The Tap Trailhouse
Restaurant Reservation
Table for 2, tonight at 9:30pm



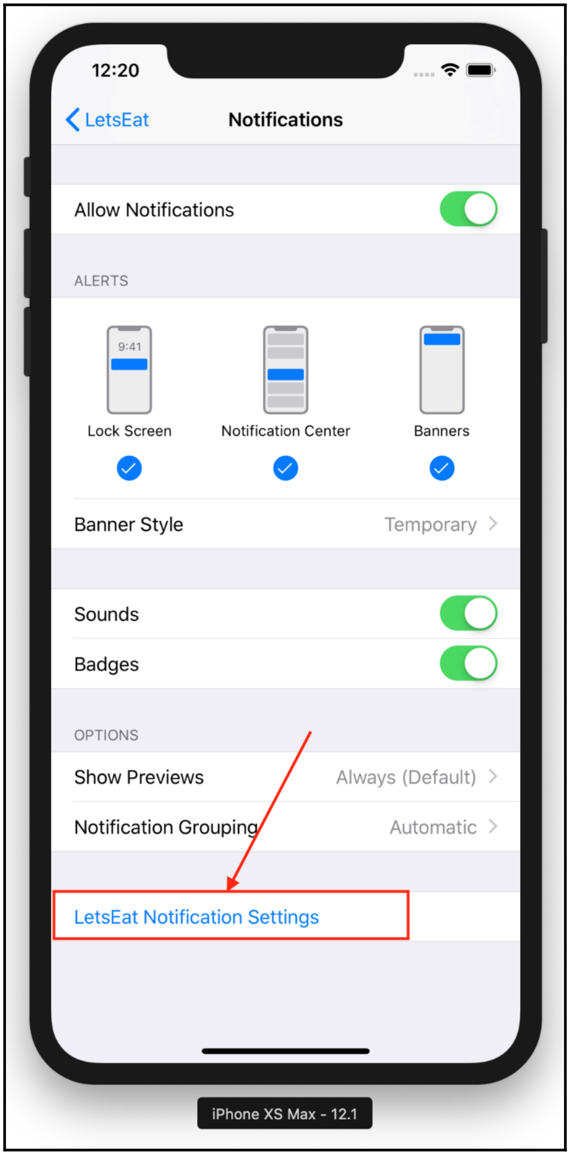
Choose options for adding these files:

Destination: ☑ Copy items if needed

Added folders: ● Create groups
○ Create folder references

Add to targets: ☑ LetsEat
□ MessageApp
□ LetsEatDataKit
□ LetsEatNotificationExtension

```
▼ LetsEat                                              M
  ▼ LetsEat
    ▶ Application                                      —
    ▶ Misc                                             —
    ▼ Controllers
      ▼ Settings
          NotificationSettingsViewController.swift     A
          NotificationSettings.storyboard              A
      ▶ Map                                            —
      ▶ Locations                                      —
      ▶ Restaurants                                    —
      ▶ Explore                                        —
      ▶ Review Form                                    —
      ▶ Restaurant Details                             —
      ▶ Photo Filter                                   —
  ▶ LetsEatDataKit                                     —
  ▶ LetsEatContentExtension                            —
  ▶ Frameworks
  ▶ Products
```



Allow Notifications ●

ALERTS

Lock Screen ✓

Notification Center ✓

Banners ✓

Banner Style Temporary >

Sounds ●

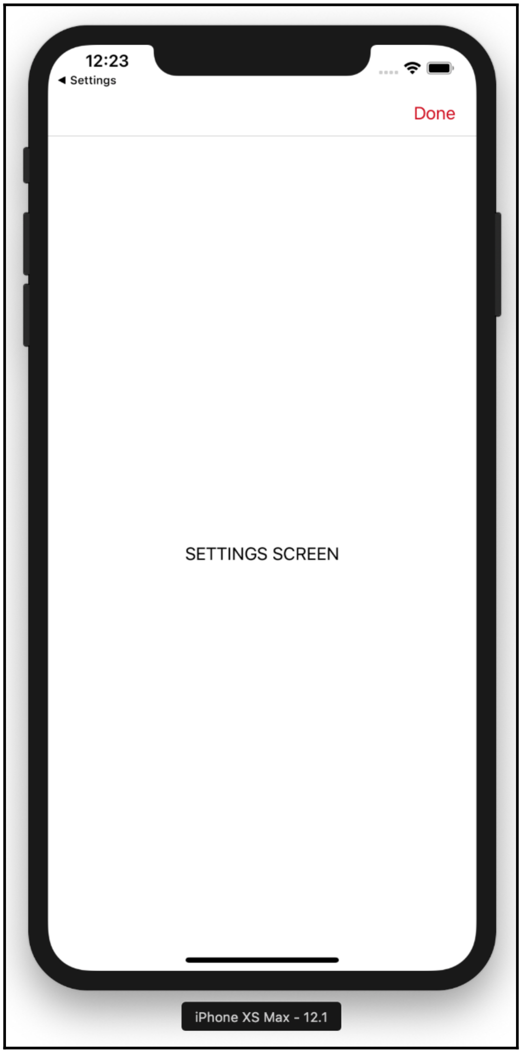
Badges ●

OPTIONS

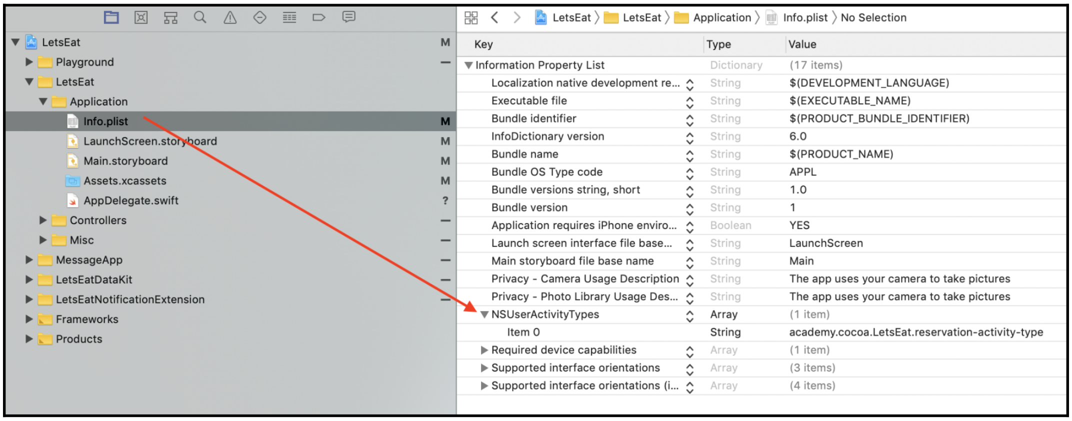
Show Previews Always (Default) >

Notification Grouping Automatic >

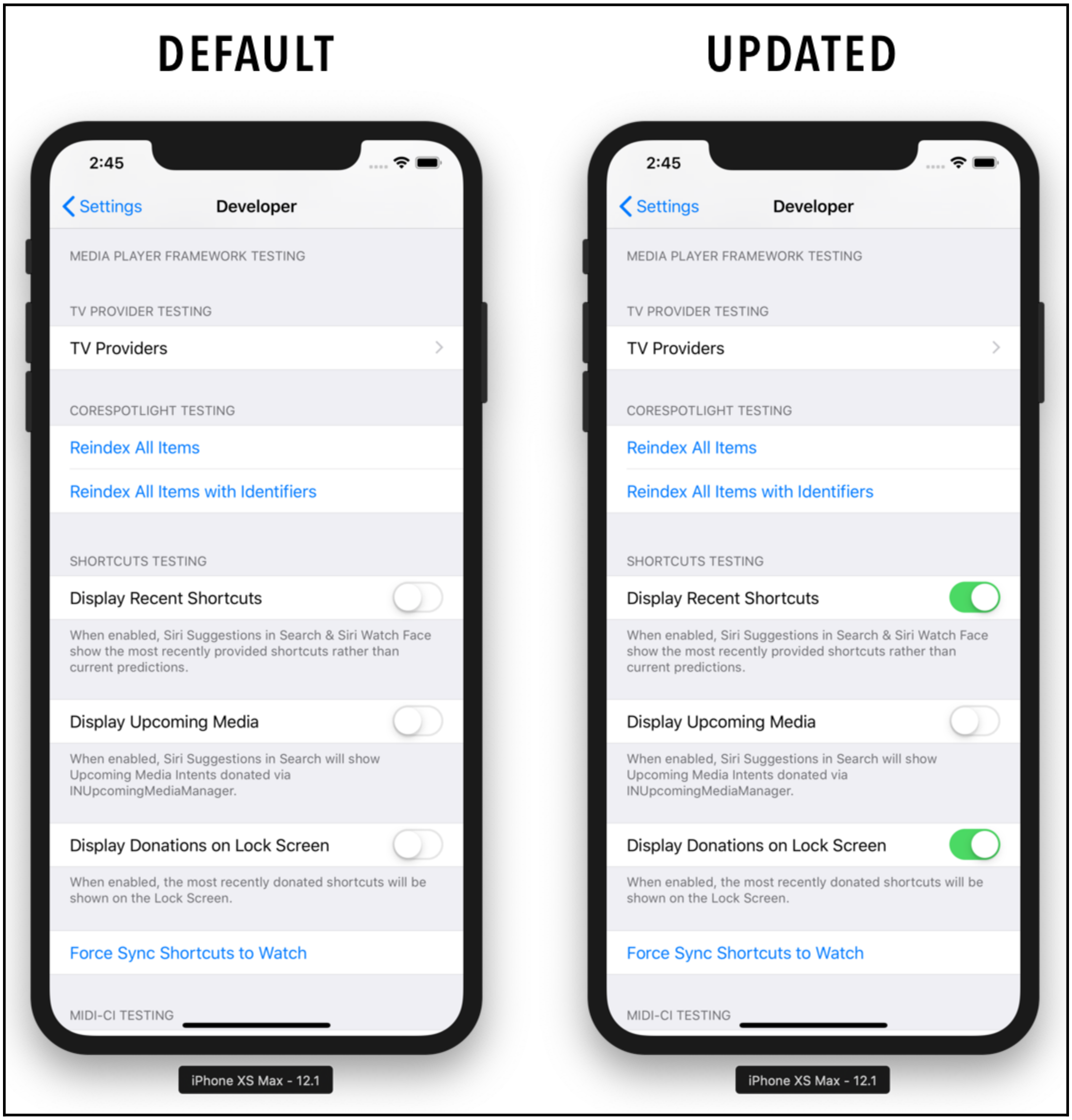
LetsEat Notification Settings


12:23
◀ Settings
Done
SETTINGS SCREEN
iPhone XS Max - 12.1

Chapter 24: SiriKit



Key	Type	Value
▼ Information Property List	Dictionary	(17 items)
Localization native development re...	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
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
▼ NSUserActivityTypes	Array	(1 item)
Item 0	String	academy.cocoa.LetsEat.reservation-activity-type
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (i...	Array	(4 items)



DEFAULT

2:45

Settings Developer

MEDIA PLAYER FRAMEWORK TESTING

TV PROVIDER TESTING

TV Providers >

CORESPOTLIGHT TESTING

Reindex All Items

Reindex All Items with Identifiers

SHORTCUTS TESTING

Display Recent Shortcuts ○

When enabled, Siri Suggestions in Search & Siri Watch Face show the most recently provided shortcuts rather than current predictions.

Display Upcoming Media ○

When enabled, Siri Suggestions in Search will show Upcoming Media Intents donated via INUpcomingMediaManager.

Display Donations on Lock Screen ○

When enabled, the most recently donated shortcuts will be shown on the Lock Screen.

Force Sync Shortcuts to Watch

MIDI-CI TESTING

iPhone XS Max - 12.1

UPDATED

2:45

Settings Developer

MEDIA PLAYER FRAMEWORK TESTING

TV PROVIDER TESTING

TV Providers >

CORESPOTLIGHT TESTING

Reindex All Items

Reindex All Items with Identifiers

SHORTCUTS TESTING

Display Recent Shortcuts ●

When enabled, Siri Suggestions in Search & Siri Watch Face show the most recently provided shortcuts rather than current predictions.

Display Upcoming Media ○

When enabled, Siri Suggestions in Search will show Upcoming Media Intents donated via INUpcomingMediaManager.

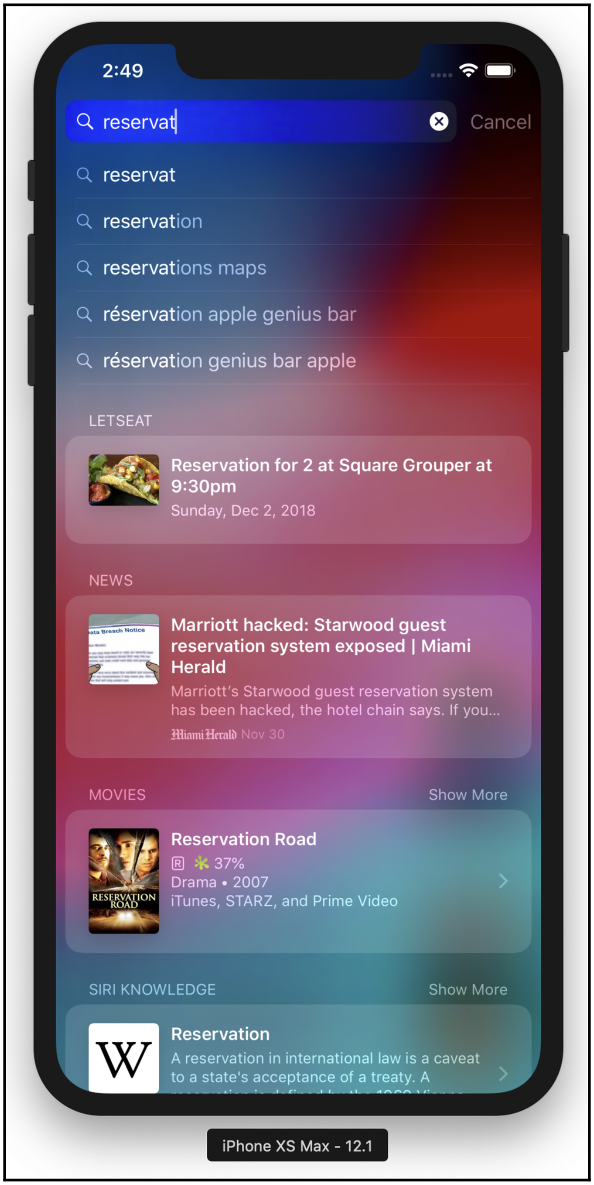
Display Donations on Lock Screen ●

When enabled, the most recently donated shortcuts will be shown on the Lock Screen.

Force Sync Shortcuts to Watch

MIDI-CI TESTING

iPhone XS Max - 12.1



2:49

reservat

Cancel

reservat

reservation

reservations maps

réservation apple genius bar

réservation genius bar apple

LETSEAT

Reservation for 2 at Square Grouper at 9:30pm
Sunday, Dec 2, 2018

NEWS

Marriott hacked: Starwood guest reservation system exposed | Miami Herald
Marriott's Starwood guest reservation system has been hacked, the hotel chain says. If you...
Miami Herald Nov 30

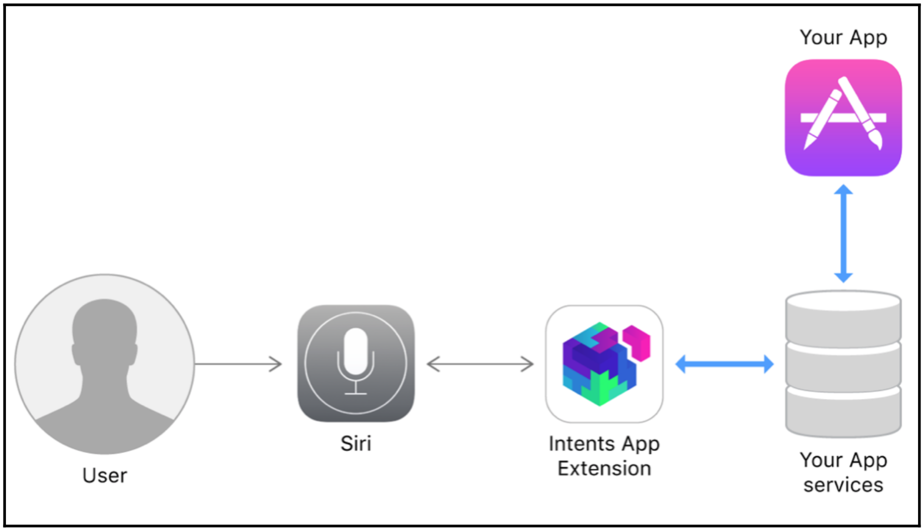
MOVIES Show More

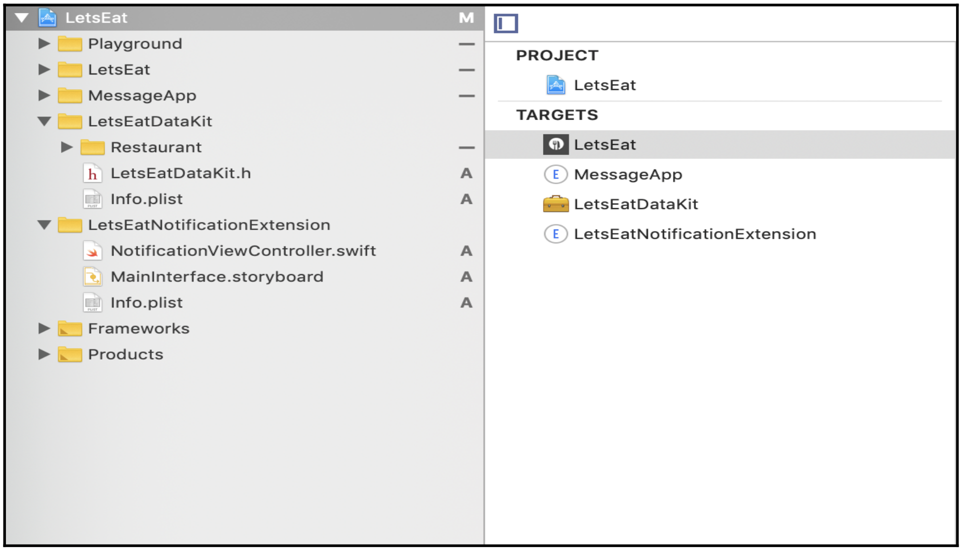
Reservation Road
R 37%
Drama • 2007
iTunes, STARZ, and Prime Video

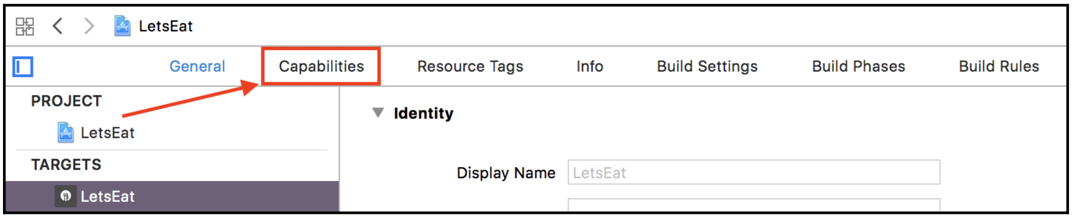
SIRI KNOWLEDGE Show More

Reservation
A reservation in international law is a caveat to a state's acceptance of a treaty. A
reservation is defined by the 1969 Vienna

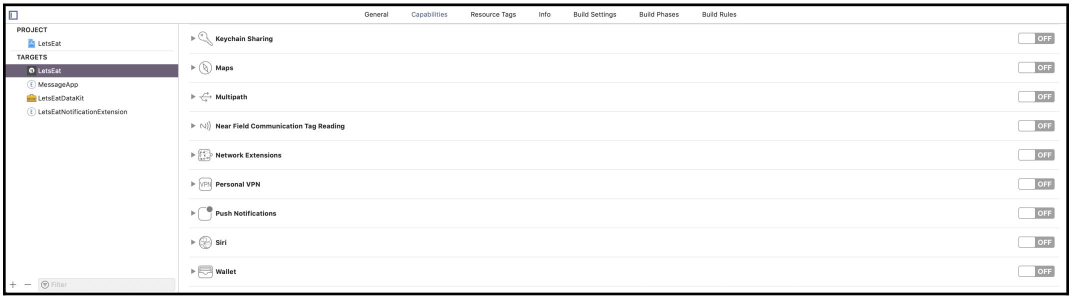
iPhone XS Max - 12.1



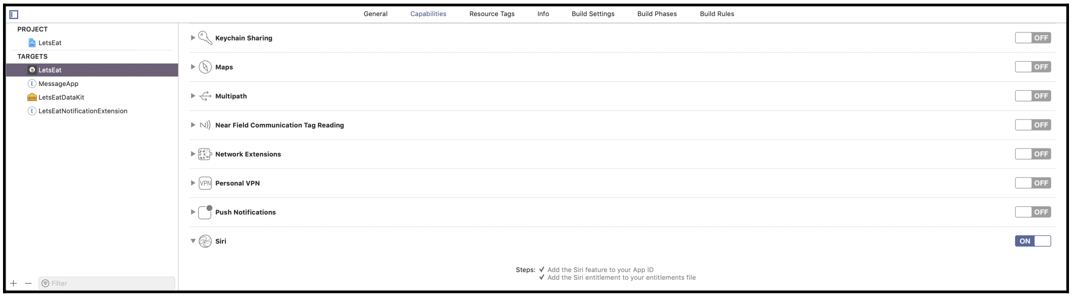




Display Name: LetsEat

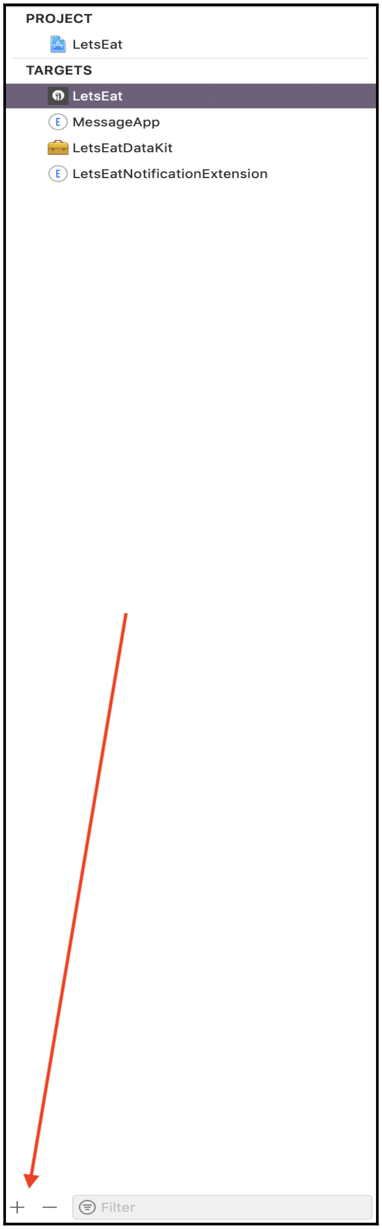


Keychain Sharing OFF
Maps OFF
Multipath OFF
Near Field Communication Tag Reading OFF
Network Extensions OFF
Personal VPN OFF
Push Notifications OFF
Siri OFF
Wallet OFF

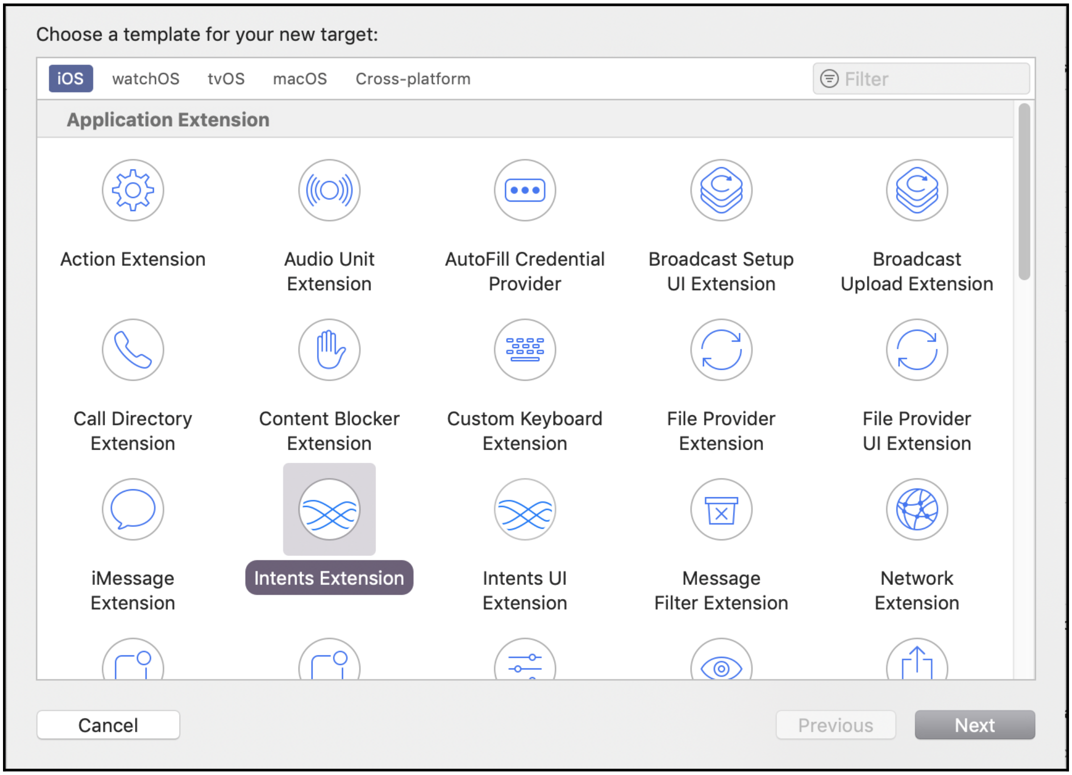


Keychain Sharing OFF
Maps OFF
Multipath OFF
Near Field Communication Tag Reading OFF
Network Extensions OFF
Personal VPN OFF
Push Notifications OFF
Siri ON

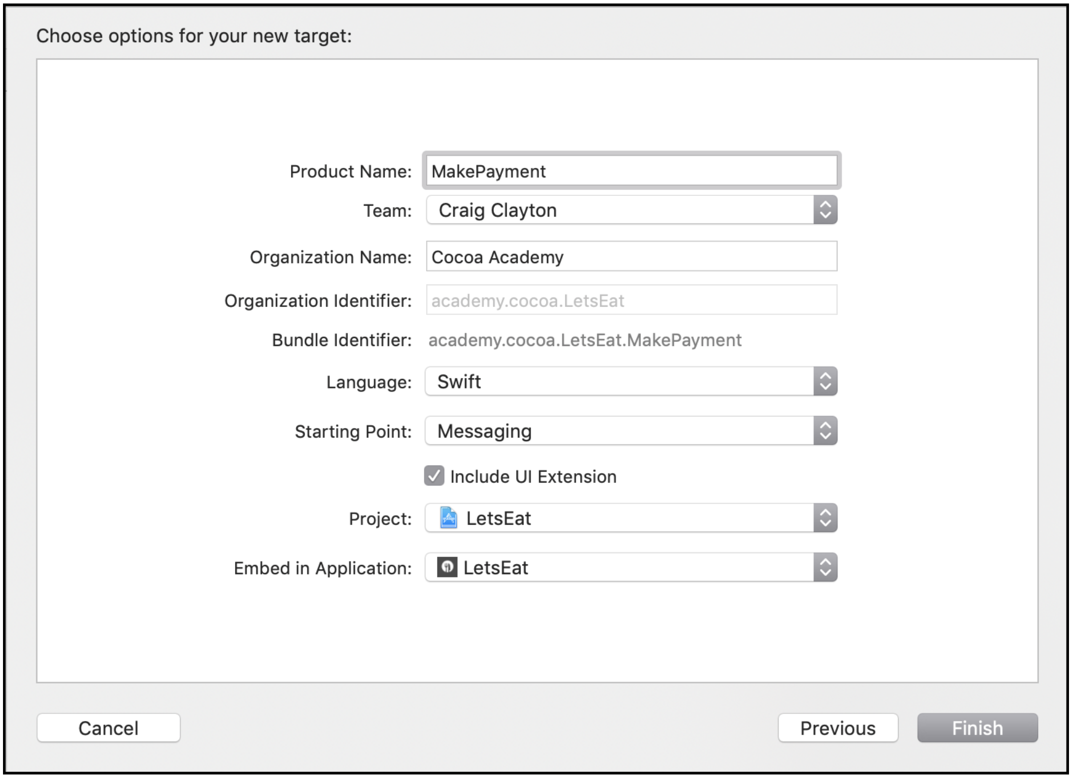
Steps: ✓ Add the Siri feature to your App ID
✓ Add the Siri entitlement to your entitlements file



PROJECT
LetsEat
TARGETS
LetsEat
MessageApp
LetsEatDataKit
LetsEatNotificationExtension
+
—
Filter

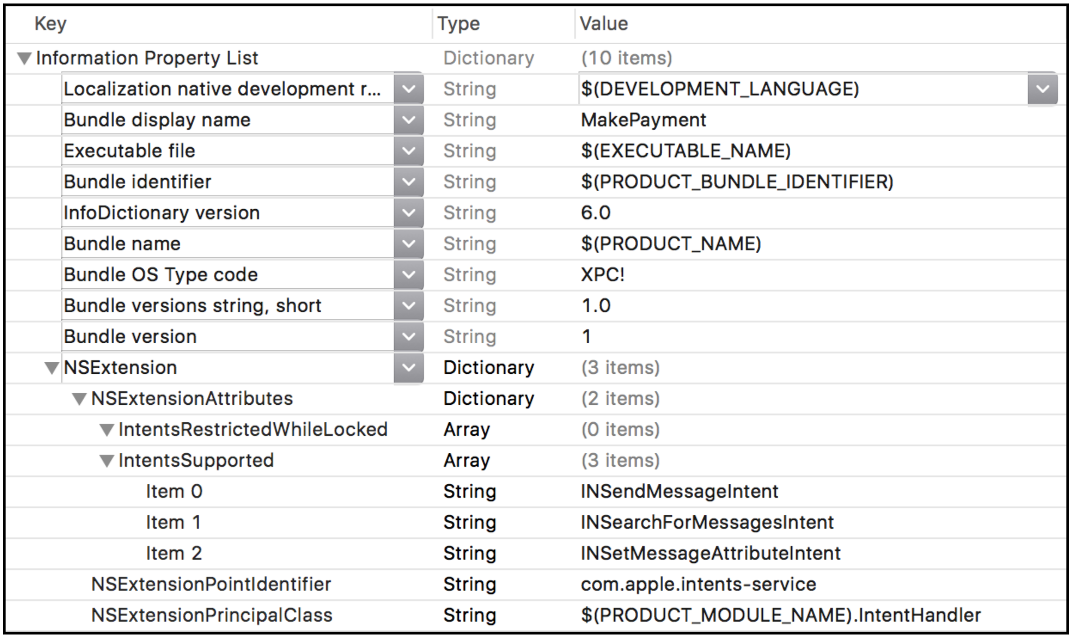


Choose a template for your new target:

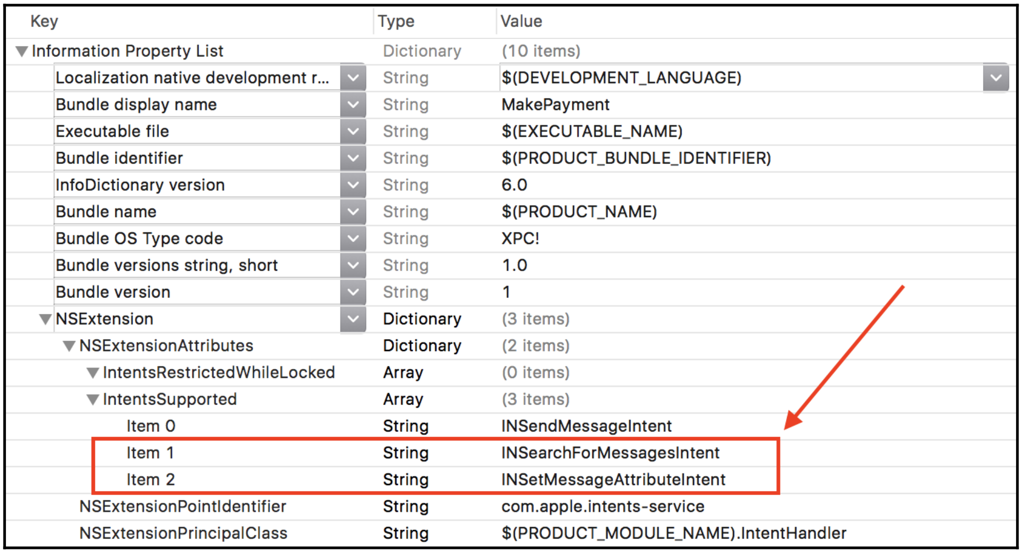


Choose options for your new target:

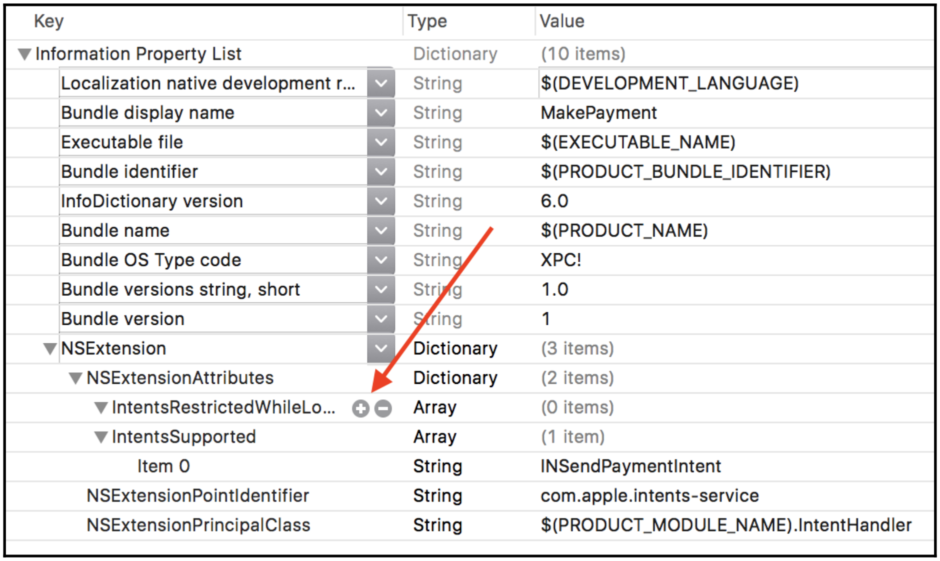
Product Name:	MakePayment
Team:	Craig Clayton
Organization Name:	Cocoa Academy
Organization Identifier:	academy.cocoa.LetsEat
Bundle Identifier:	academy.cocoa.LetsEat.MakePayment
Language:	Swift
Starting Point:	Messaging
	☑ Include UI Extension
Project:	LetsEat
Embed in Application:	LetsEat



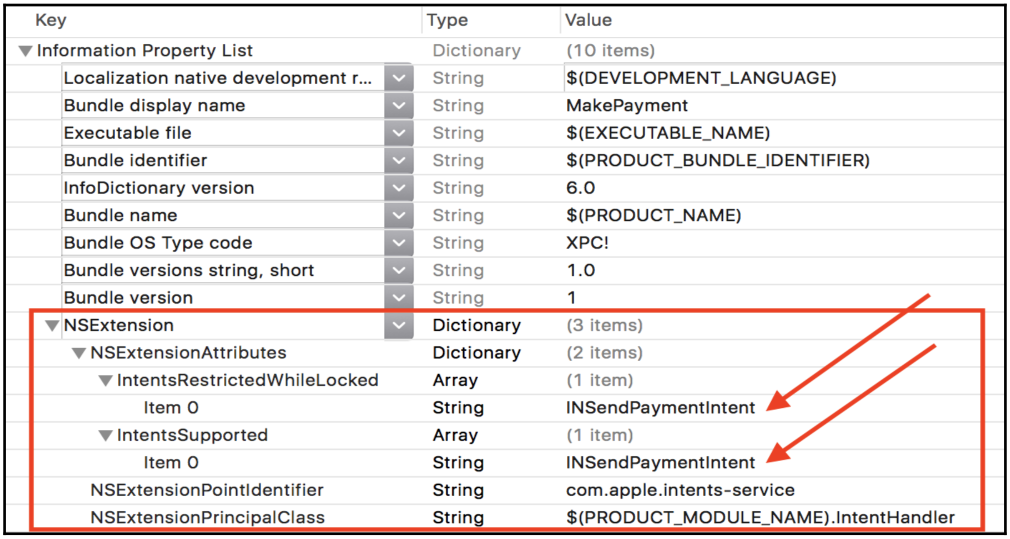
Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
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



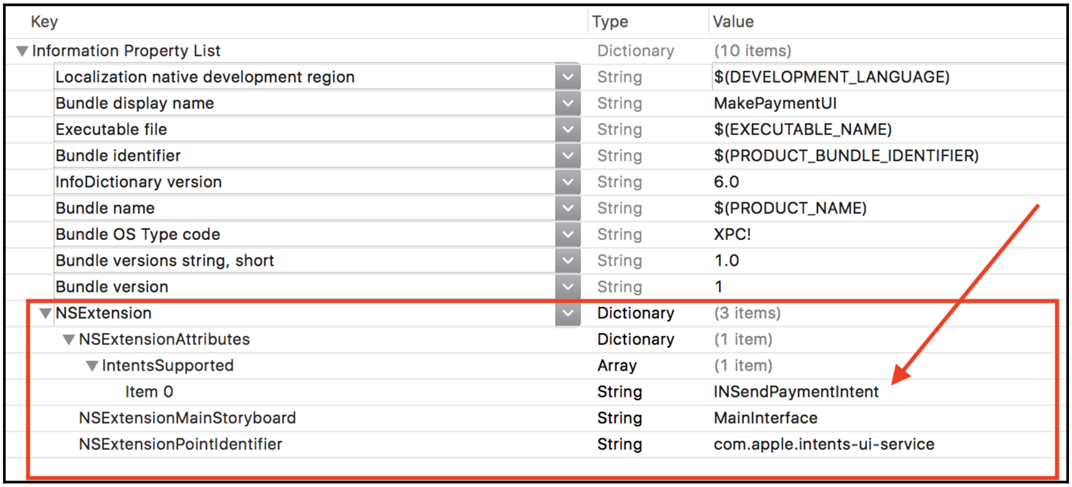
Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
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



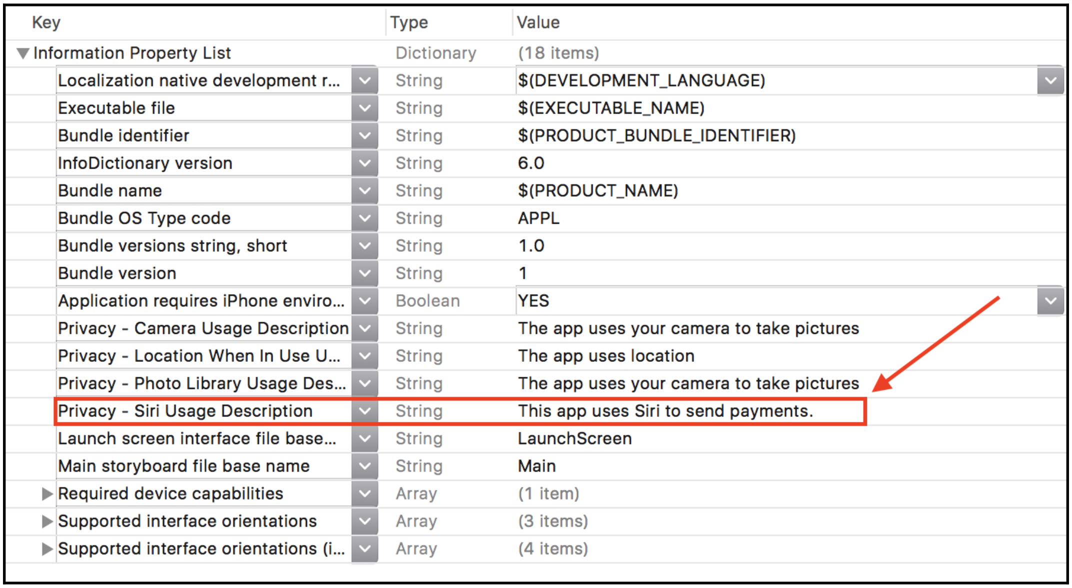
Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle 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



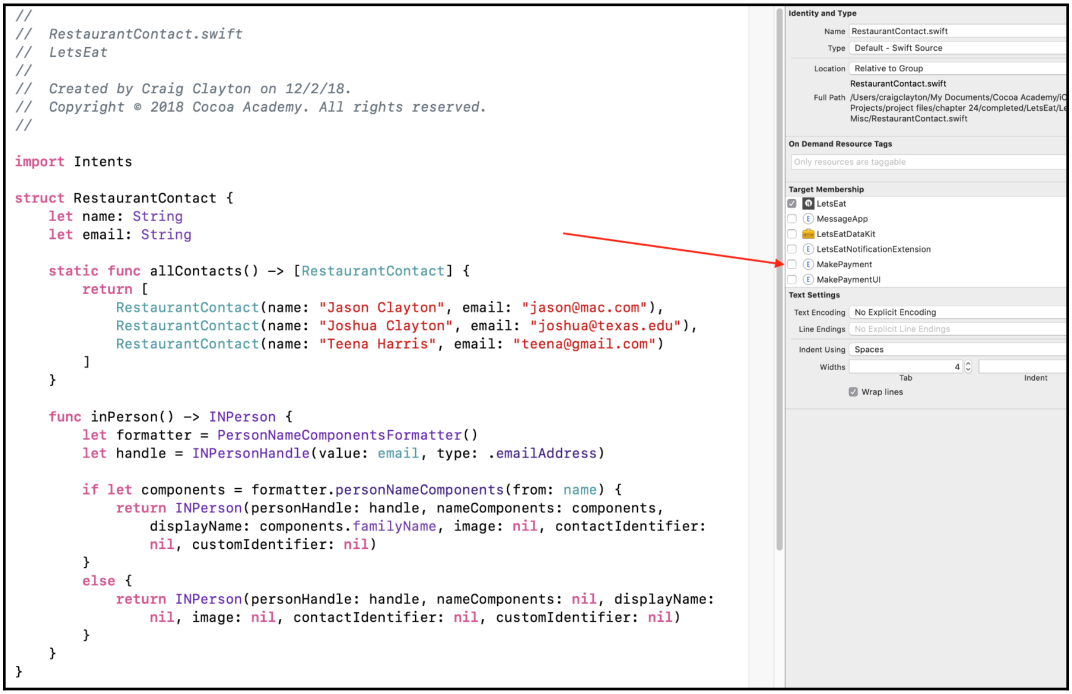
Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development r...	String	$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	MakePayment
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle 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)



```
//
//  RestaurantContact.swift
//  LetsEat
//
//  Created by Craig Clayton on 12/2/18.
//  Copyright © 2018 Cocoa Academy. All rights reserved.
//

import Intents

struct RestaurantContact {
    let name: String
    let email: String

    static func allContacts() -> [RestaurantContact] {
        return [
            RestaurantContact(name: "Jason Clayton", email: "jason@mac.com"),
            RestaurantContact(name: "Joshua Clayton", email: "joshua@texas.edu"),
            RestaurantContact(name: "Teena Harris", email: "teena@gmail.com")
        ]
    }

    func inPerson() -> INPerson {
        let formatter = PersonNameComponentsFormatter()
        let handle = INPersonHandle(value: email, type: .emailAddress)

        if let components = formatter.personNameComponents(from: name) {
            return INPerson(personHandle: handle, nameComponents: components,
                displayName: components.familyName, image: nil, contactIdentifier:
                nil, customIdentifier: nil)
        }
        else {
            return INPerson(personHandle: handle, nameComponents: nil, displayName:
                nil, image: nil, contactIdentifier: nil, customIdentifier: nil)
        }
    }
}
```

Identity and Type

Name: RestaurantContact.swift

Type: Default - Swift Source

Location: Relative to Group

RestaurantContact.swift

Full Path: /Users/craigclayton/My Documents/Cocoa Academy/iC
Projects/project files/chapter 24/completed/LetsEat/Le
Misc/RestaurantContact.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

☑ LetsEat
☐ MessageApp
☐ LetsEatDataKit
☐ LetsEatNotificationExtension
☐ MakePayment
☐ MakePaymentUI

Text Settings

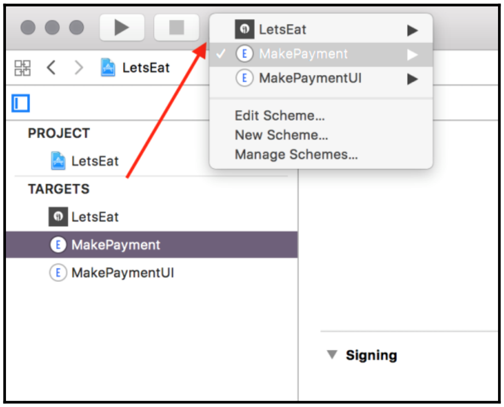
Text Encoding: No Explicit Encoding

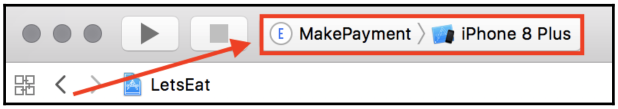
Line Endings: No Explicit Line Endings

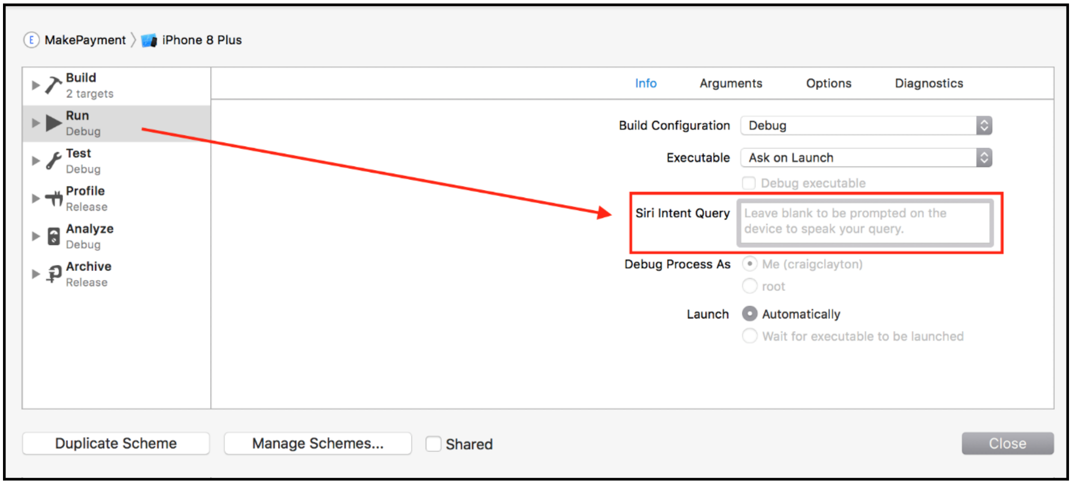
Indent Using: Spaces

Widths: 4 Tab, Indent

☑ Wrap lines







Build Configuration: Debug

Executable: Ask on Launch

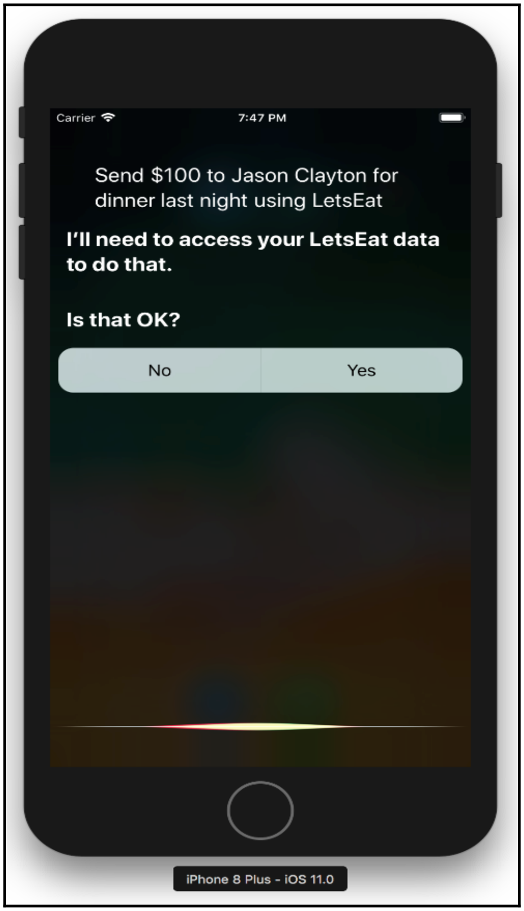
□ Debug executable

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

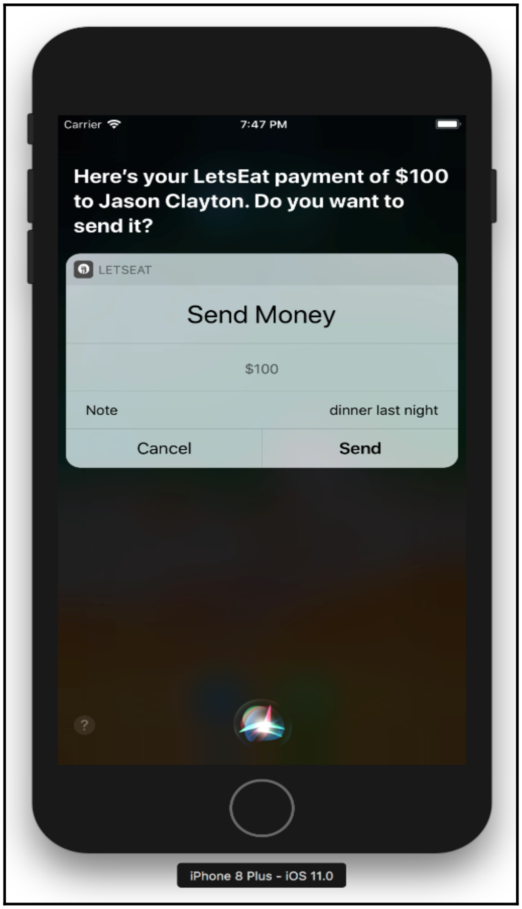
□ Shared



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?

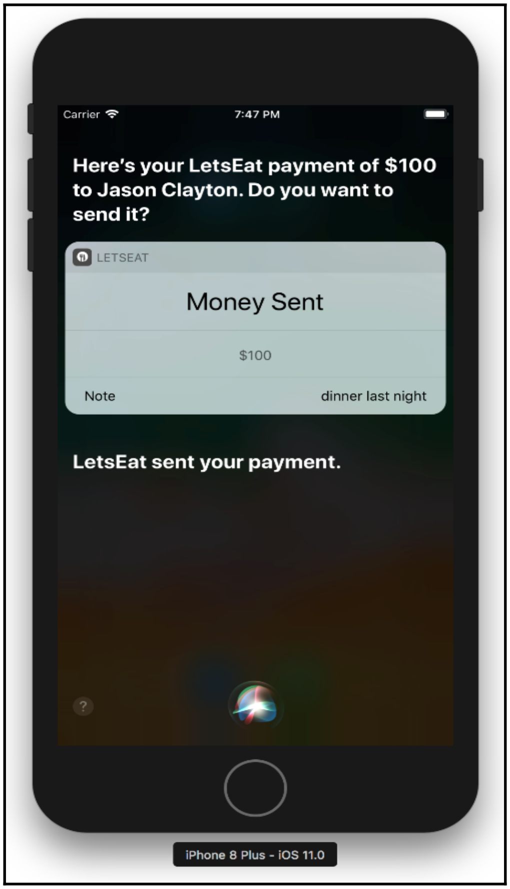


Here's your LetsEat payment of $100 to Jason Clayton. Do you want to send it?

Send Money

$100

Note dinner last night



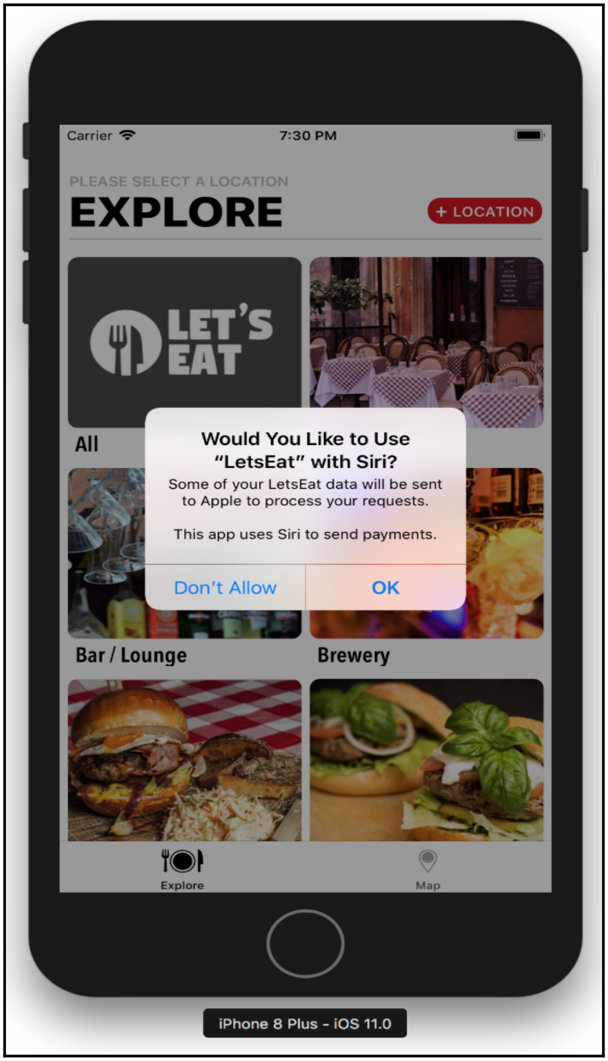
Here's your LetsEat payment of $100 to Jason Clayton. Do you want to send it?

Money Sent

$100

Note dinner last night

LetsEat sent your payment.

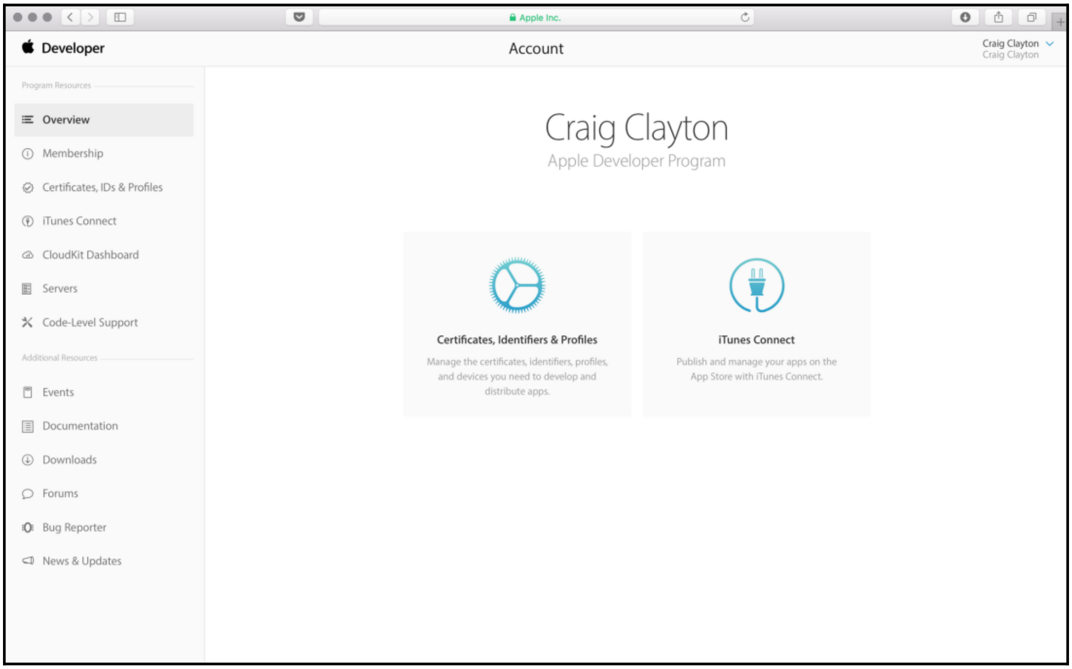


Would You Like to Use "LetsEat" with Siri?

Some of your LetsEat data will be sent to Apple to process your requests.

This app uses Siri to send payments.

Chapter 25: Beta and Store Submission



Craig Clayton

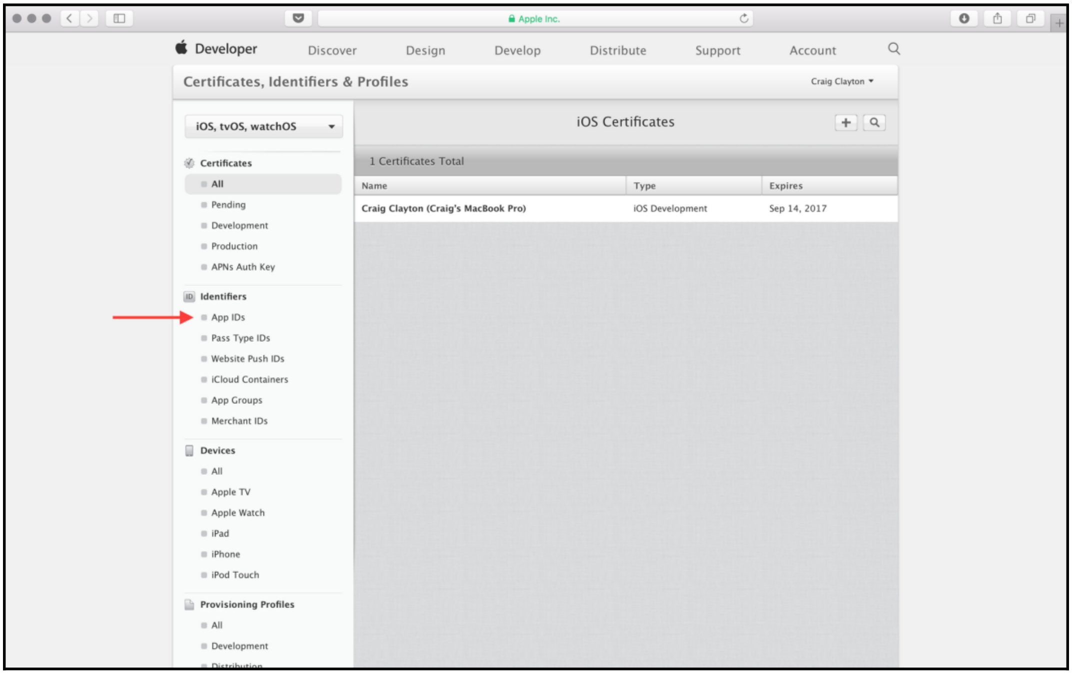
Apple Developer Program

Certificates, Identifiers & Profiles

Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.

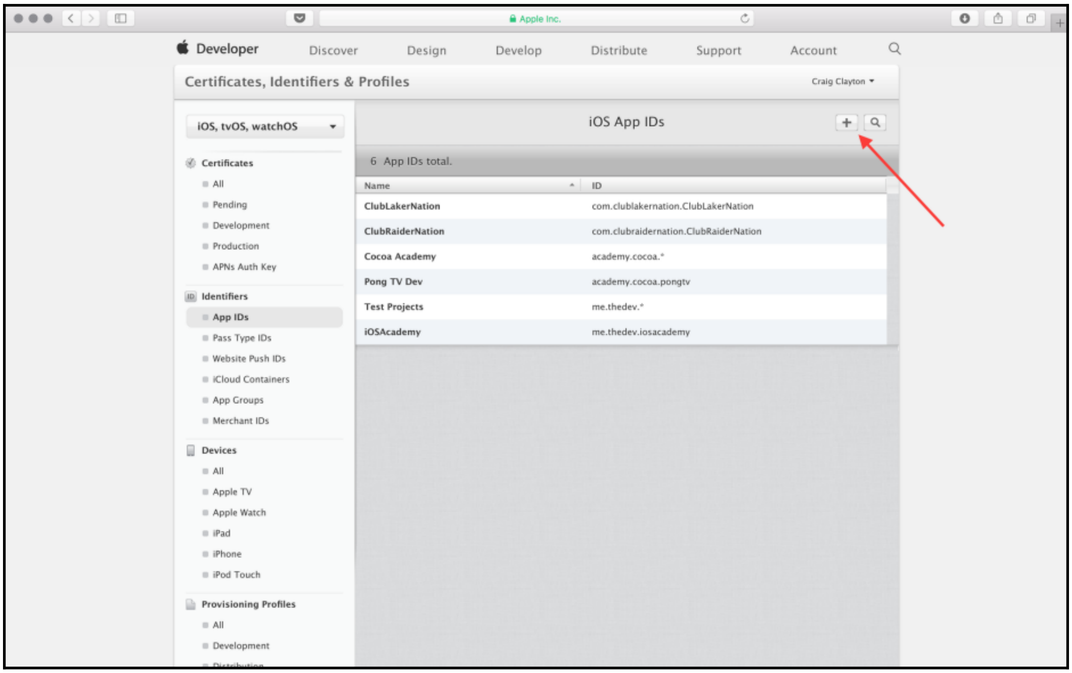
iTunes Connect

Publish and manage your apps on the App Store with iTunes Connect.



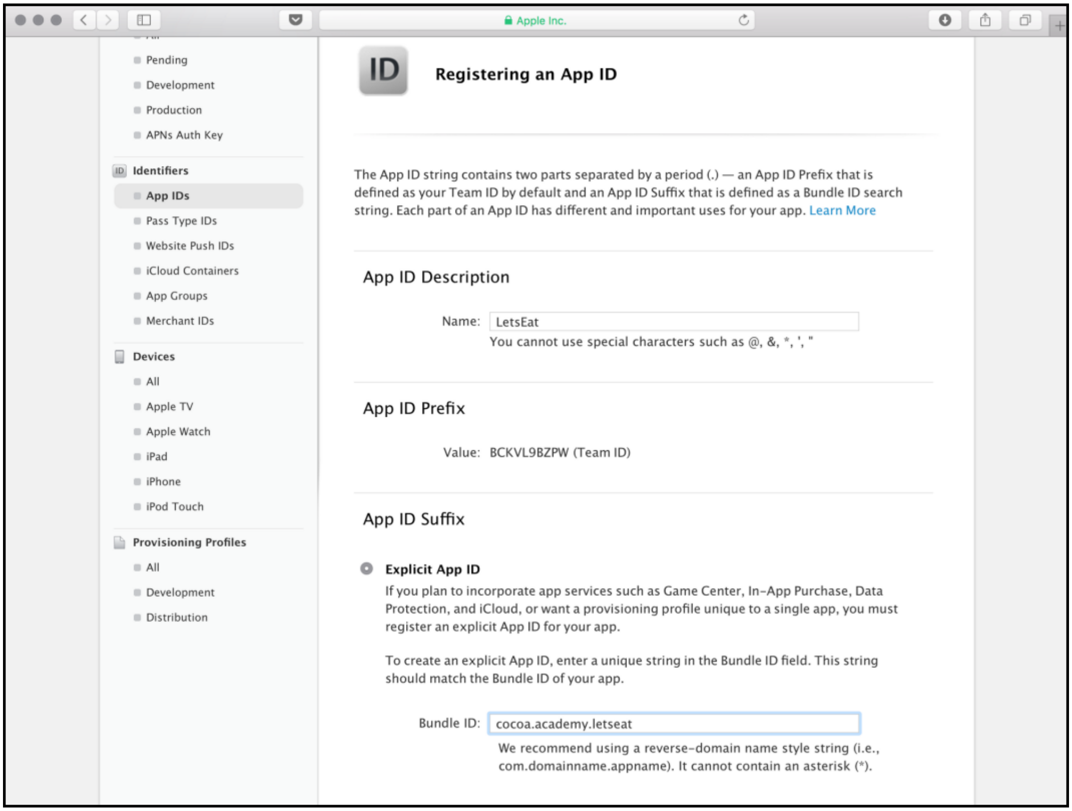
1 Certificates Total

Name	Type	Expires
Craig Clayton (Craig's MacBook Pro)	iOS Development	Sep 14, 2017



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



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

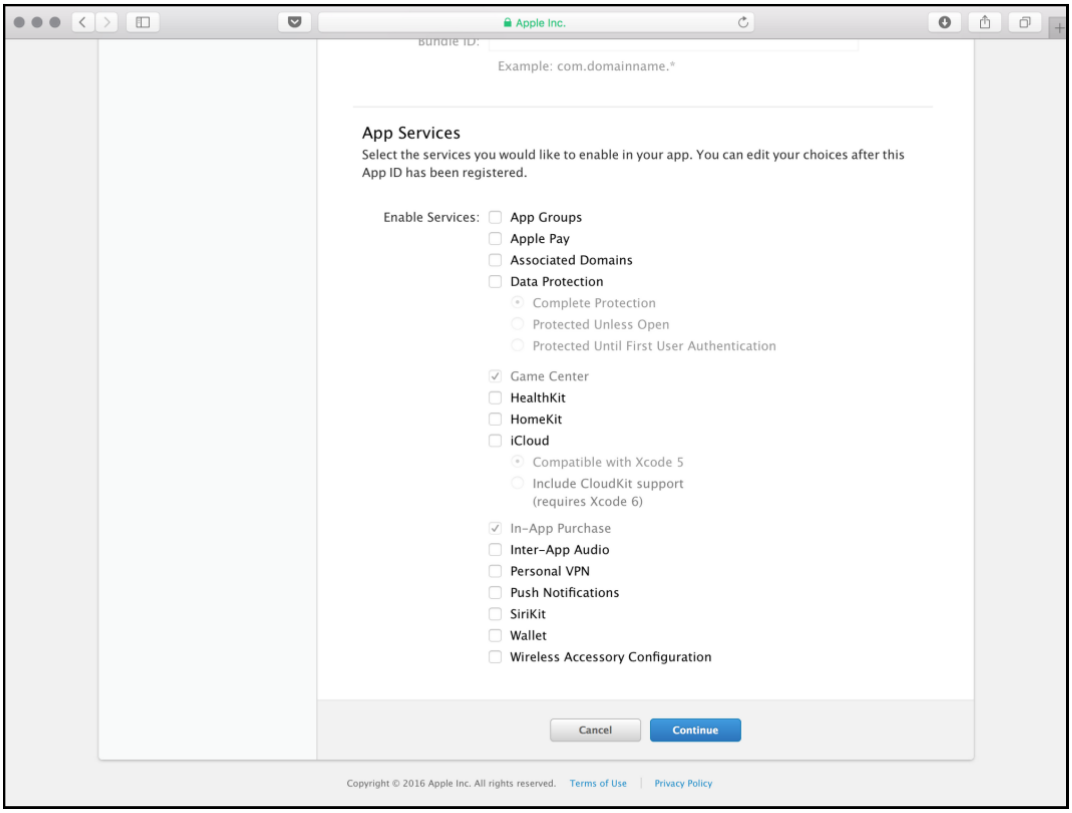
Value: BCKVL9BZPW (Team ID)

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 (*).



Bundle ID:

Example: com.domainname.*

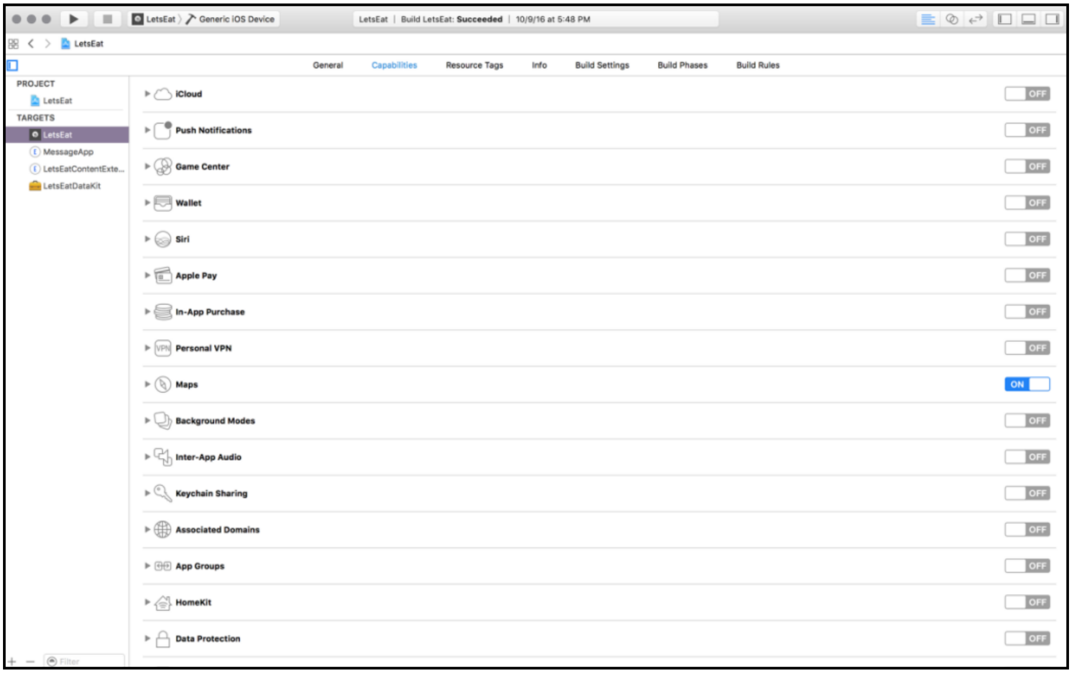
App Services

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

Copyright © 2016 Apple Inc. All rights reserved.



iCloud OFF

Push Notifications OFF

Game Center OFF

Wallet OFF

Siri OFF

Apple Pay OFF

In-App Purchase OFF

Personal VPN OFF

Maps ON

Background Modes OFF

Inter-App Audio OFF

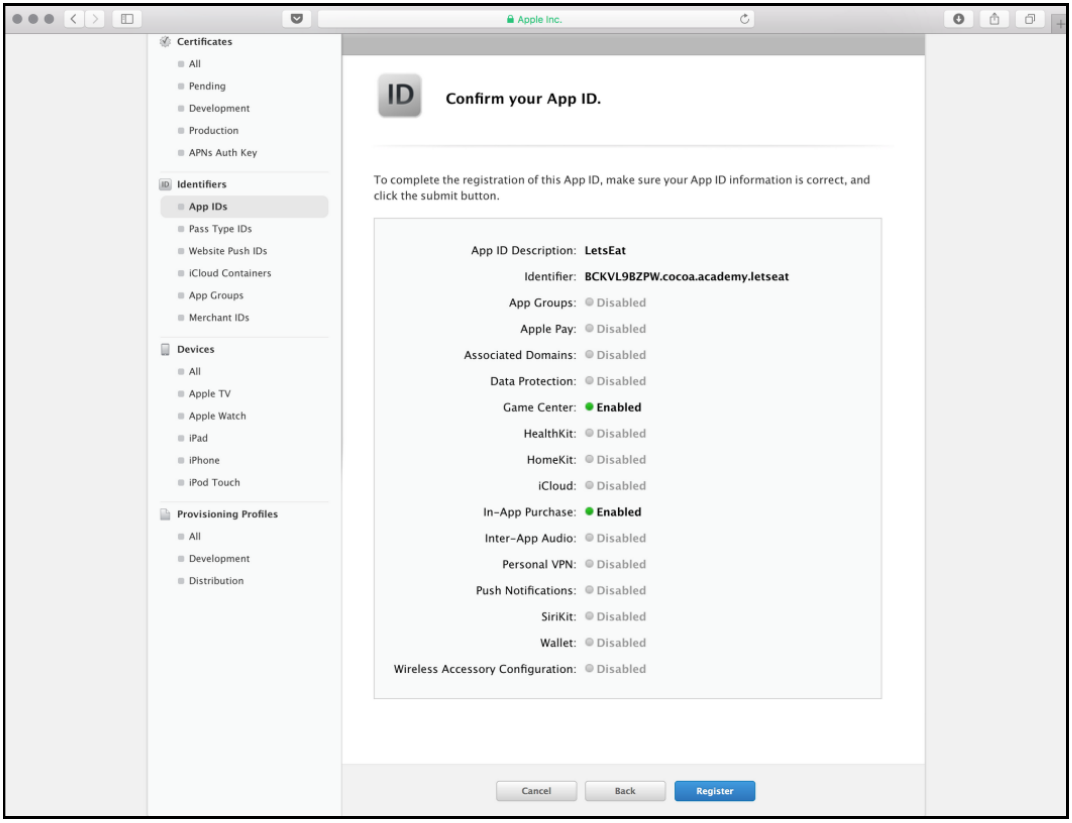
Keychain Sharing OFF

Associated Domains OFF

App Groups OFF

HomeKit OFF

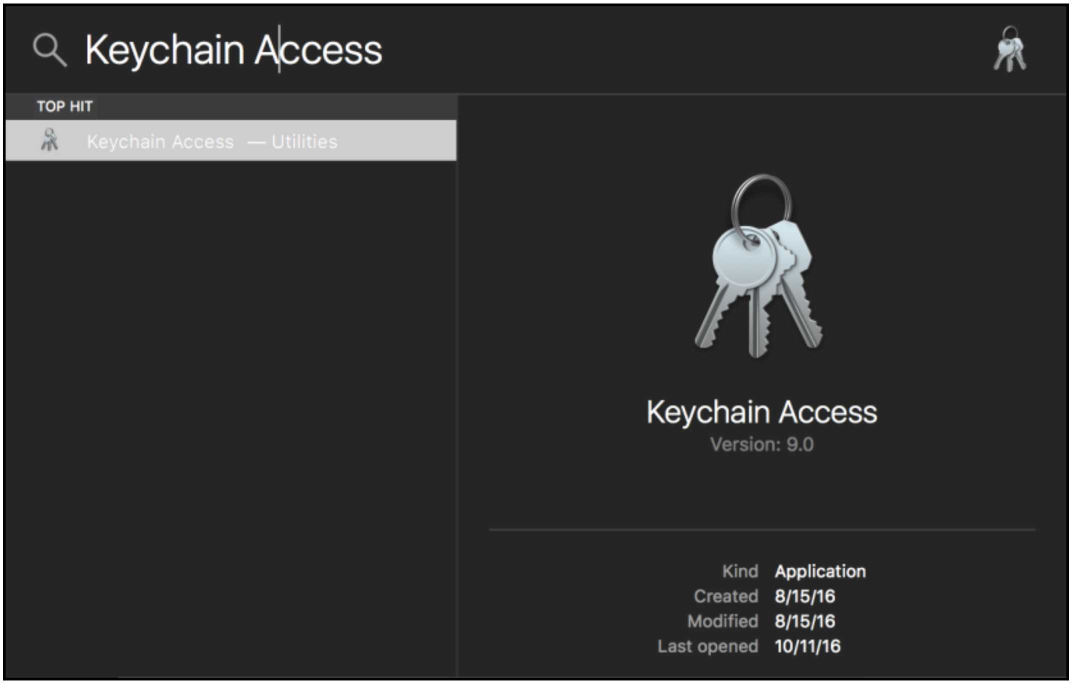
Data Protection OFF



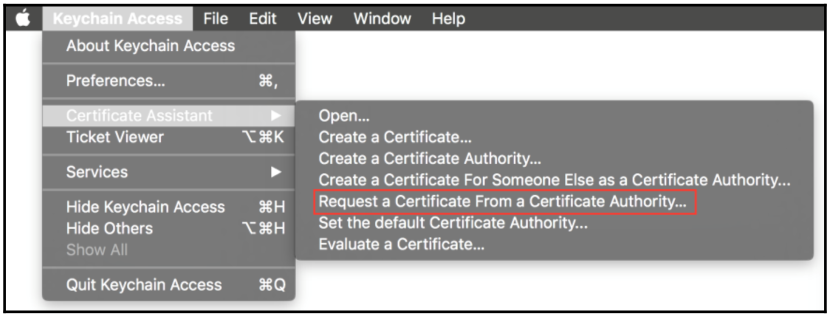
Confirm your App ID.

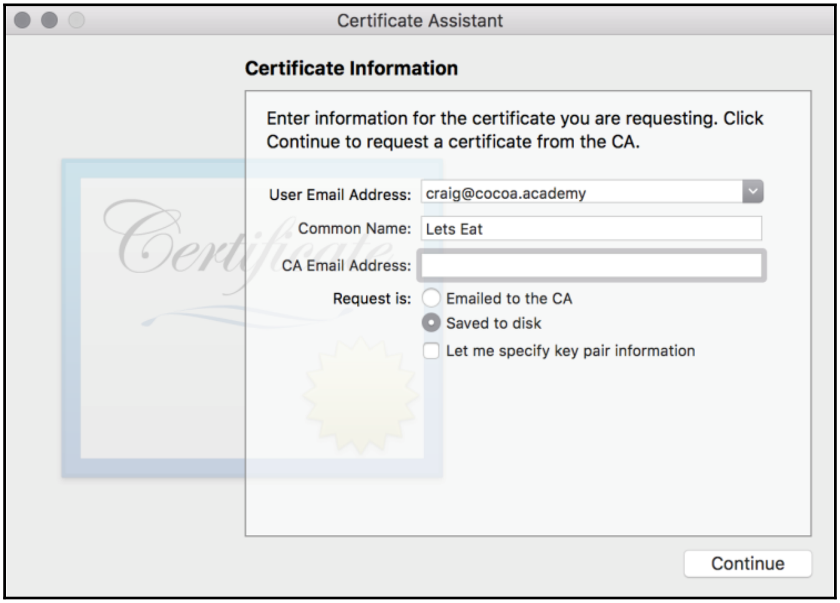
To complete the registration of this App ID, make sure your App ID information is correct, and click the submit button.

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



Kind Application
Created 8/15/16
Modified 8/15/16
Last opened 10/11/16





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

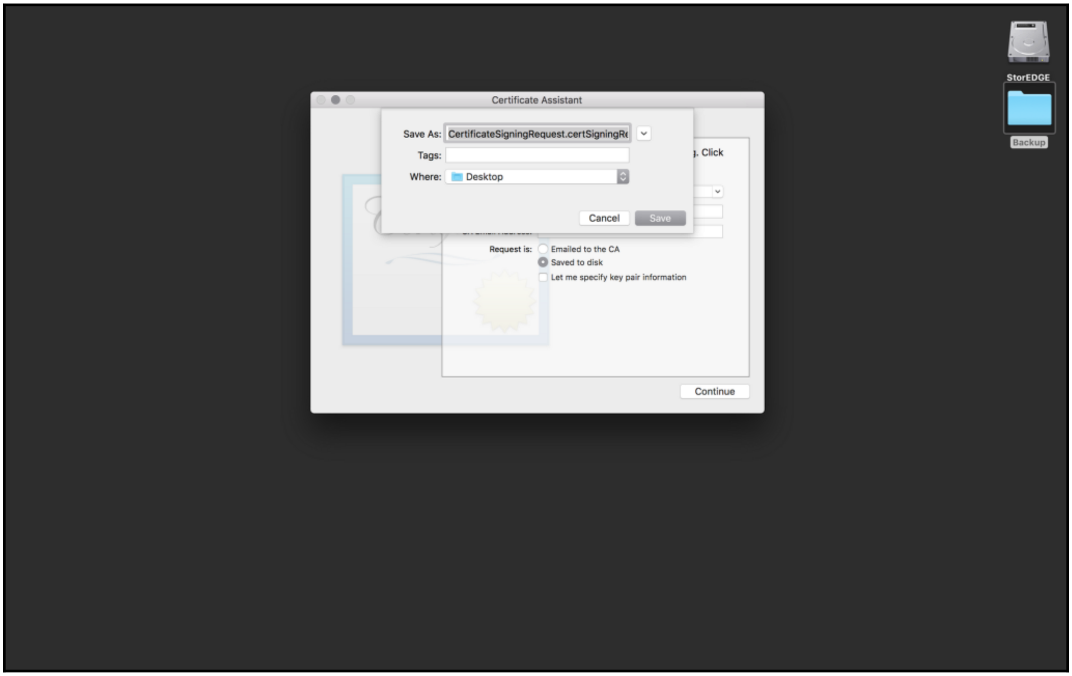
Common Name: Lets Eat

CA Email Address:

Request is: ○ Emailed to the CA

● Saved to disk

□ Let me specify key pair information



Save As: CertificateSigningRequest.certSigningRe

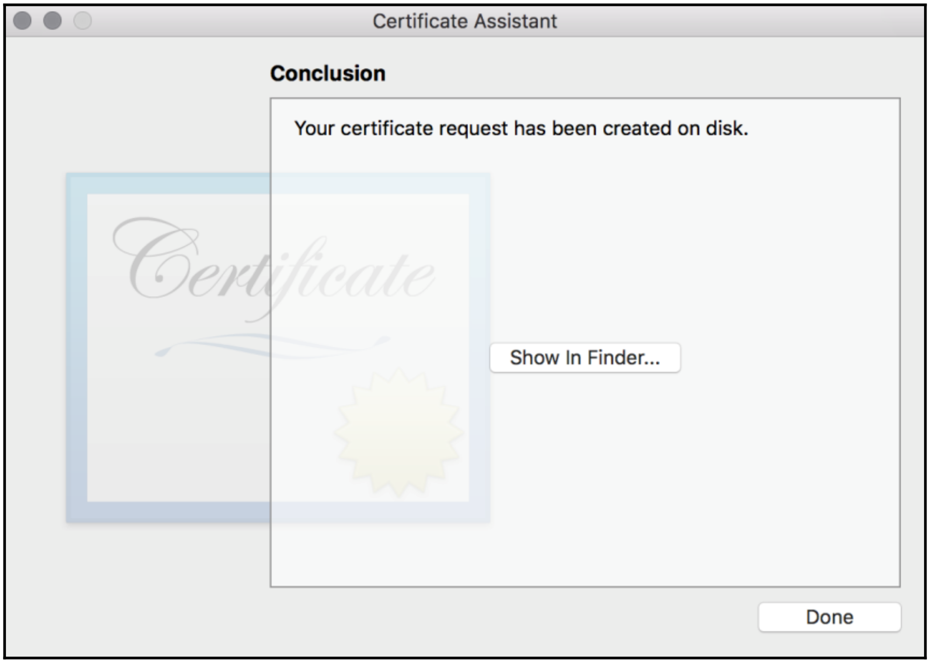
Tags:

Where: Desktop

Request is: ○ Emailed to the CA

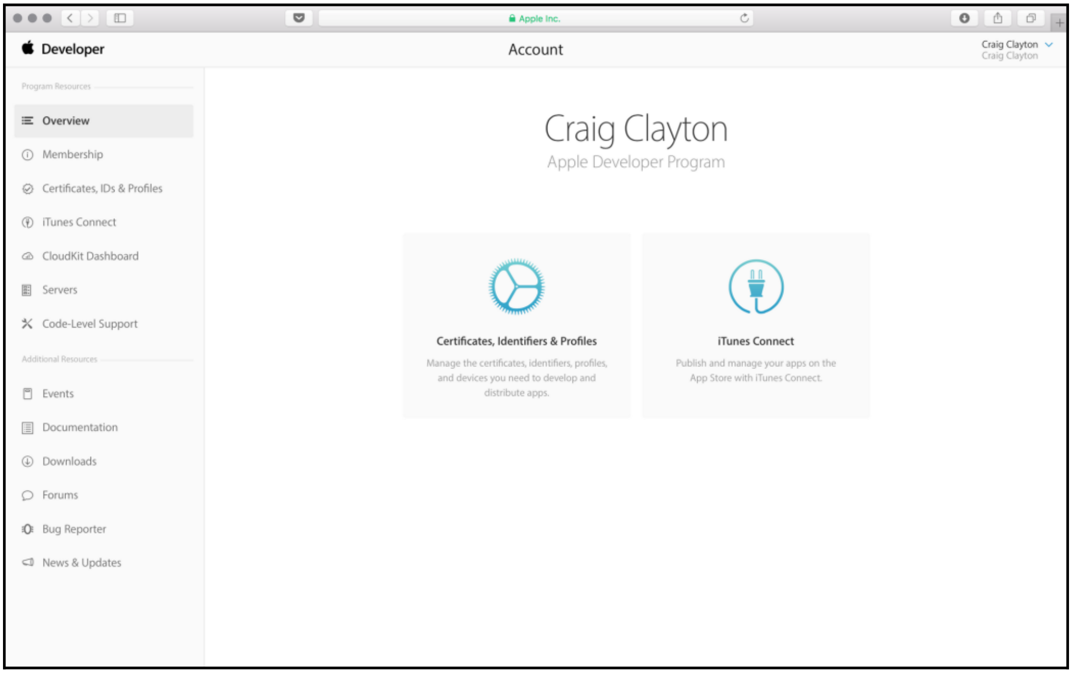
● Saved to disk

□ Let me specify key pair information



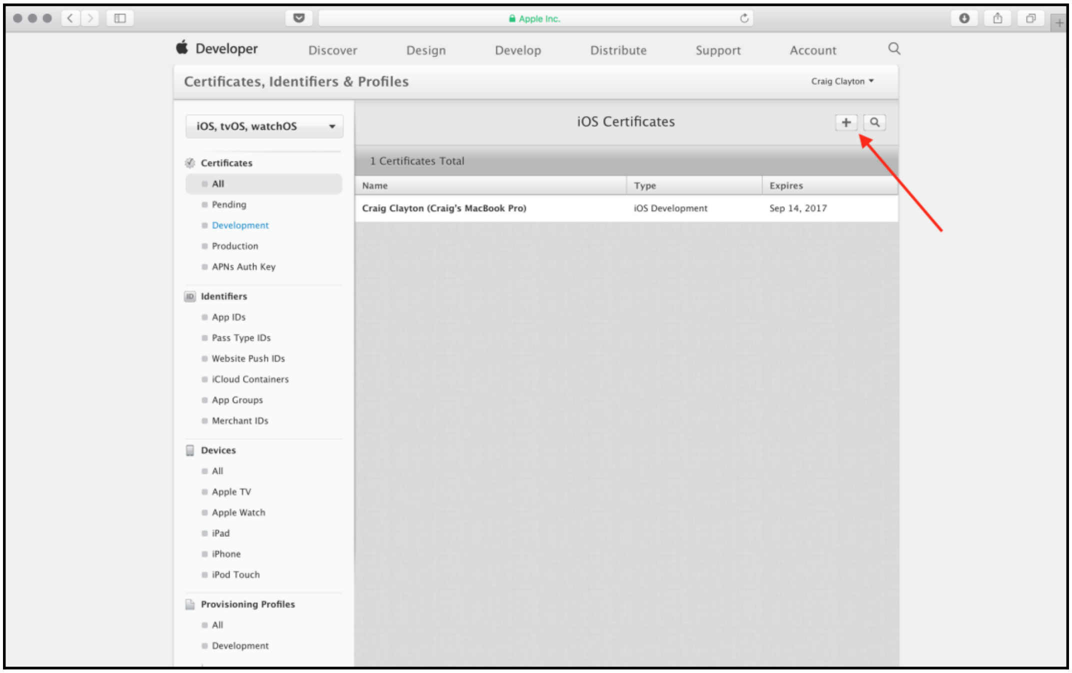
Conclusion

Your certificate request has been created on disk.



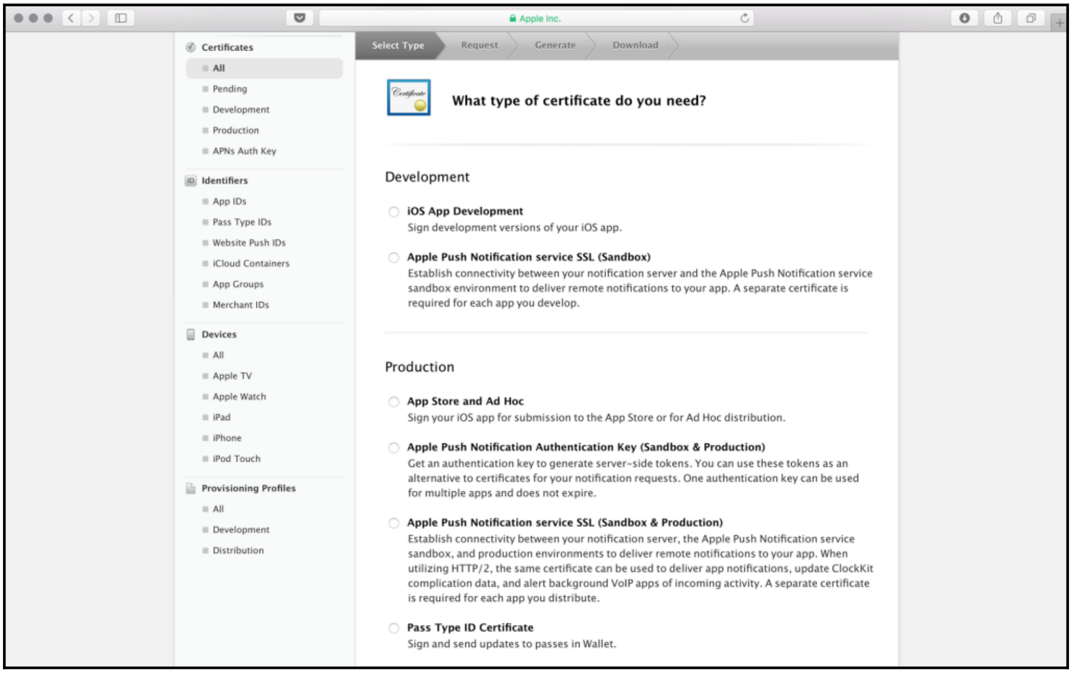
Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.

Publish and manage your apps on the App Store with iTunes Connect.



1 Certificates Total

Name	Type	Expires
Craig Clayton (Craig's MacBook Pro)	iOS Development	Sep 14, 2017



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.

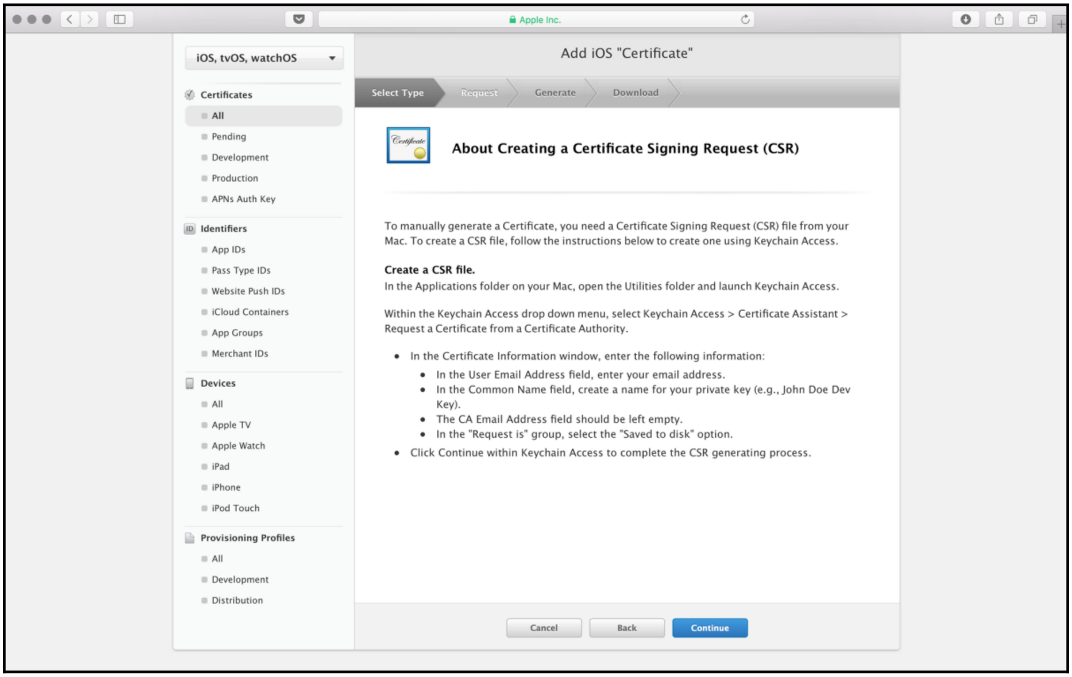
Production

○ **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.



Add iOS "Certificate"

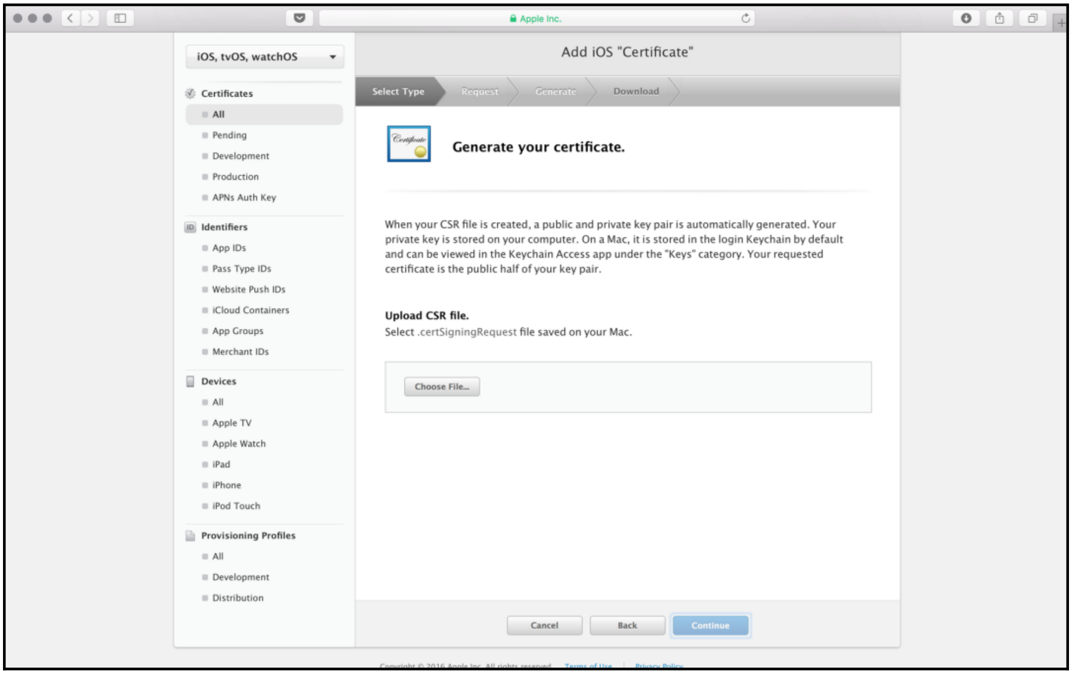
About Creating a Certificate Signing Request (CSR)

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.



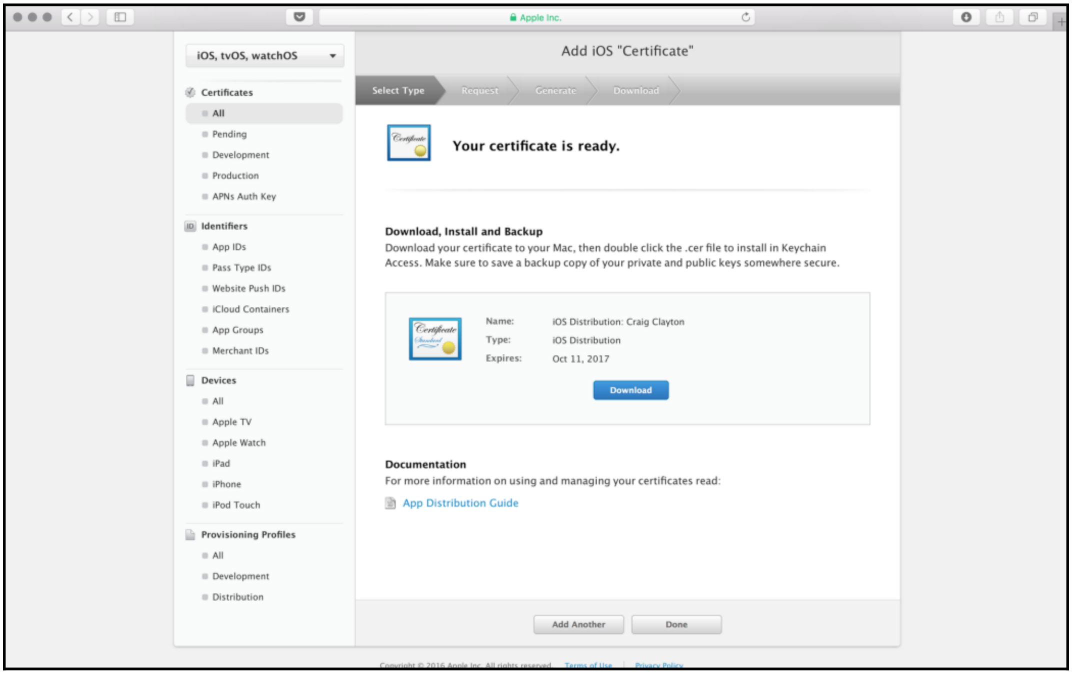
Add iOS "Certificate"

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.



Add iOS "Certificate"

Your certificate is ready.

Download, Install and Backup

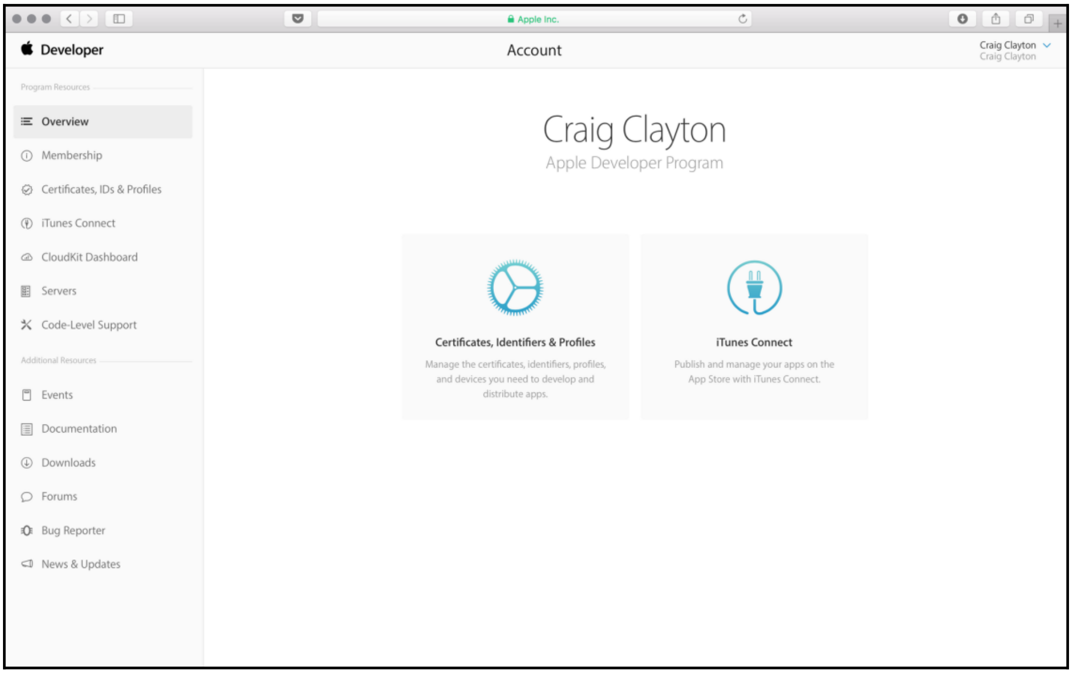
Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.

Name: iOS Distribution: Craig Clayton
Type: iOS Distribution
Expires: Oct 11, 2017

Documentation

For more information on using and managing your certificates read:

Copyright © 2016 Apple Inc. All rights reserved.



Account

Craig Clayton

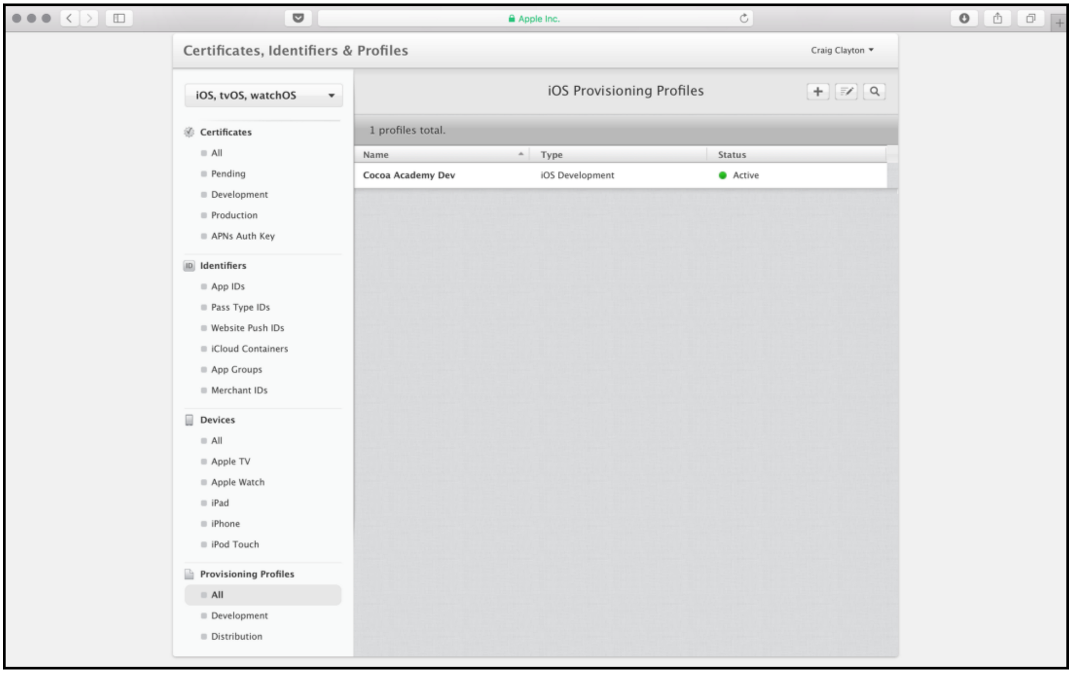
Apple Developer Program

Certificates, Identifiers & Profiles

Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.

iTunes Connect

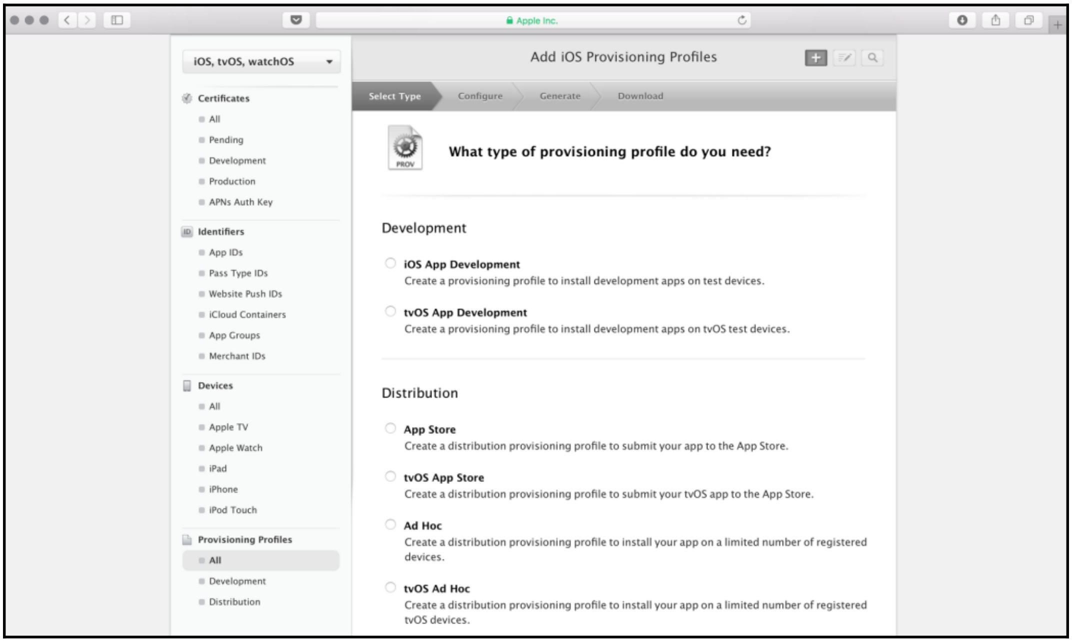
Publish and manage your apps on the App Store with iTunes Connect.



iOS Provisioning Profiles

1 profiles total.

Name	Type	Status
Cocoa Academy Dev	iOS Development	● Active



Add iOS Provisioning Profiles

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.

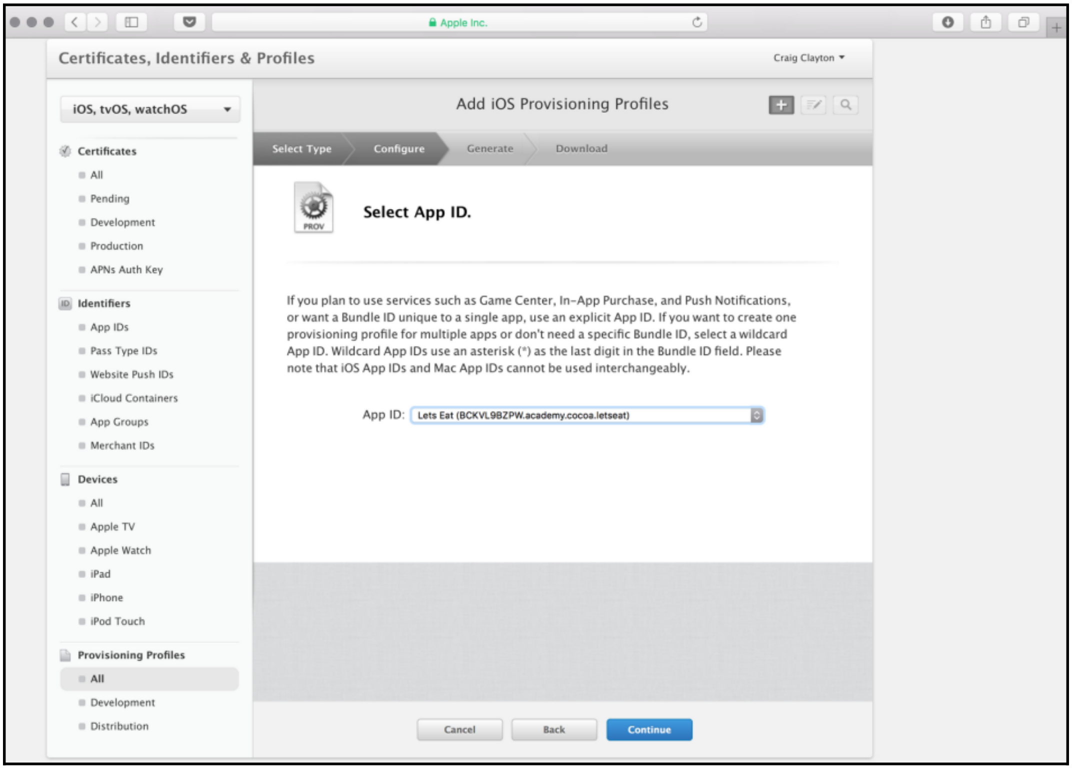
Distribution

○ **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.

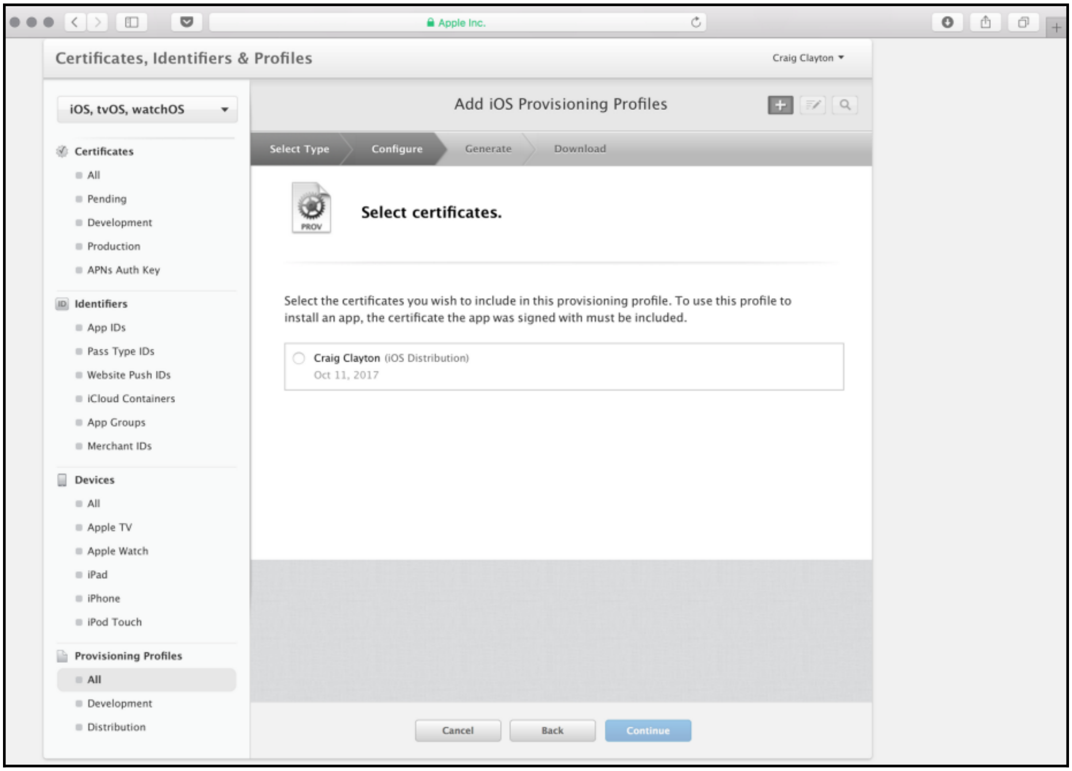
○ **tvOS Ad Hoc**
Create a distribution provisioning profile to install your app on a limited number of registered tvOS devices.



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.

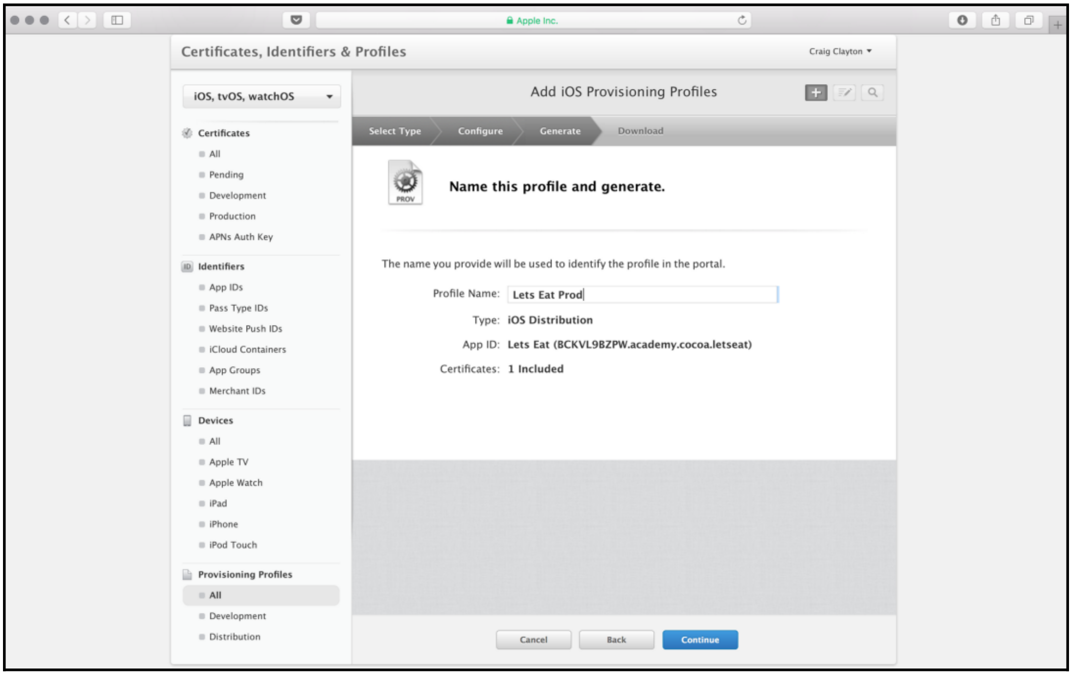
App ID: Lets Eat (BCKVL9BZPW.academy.cocoa.letseat)



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



Name this profile and generate.

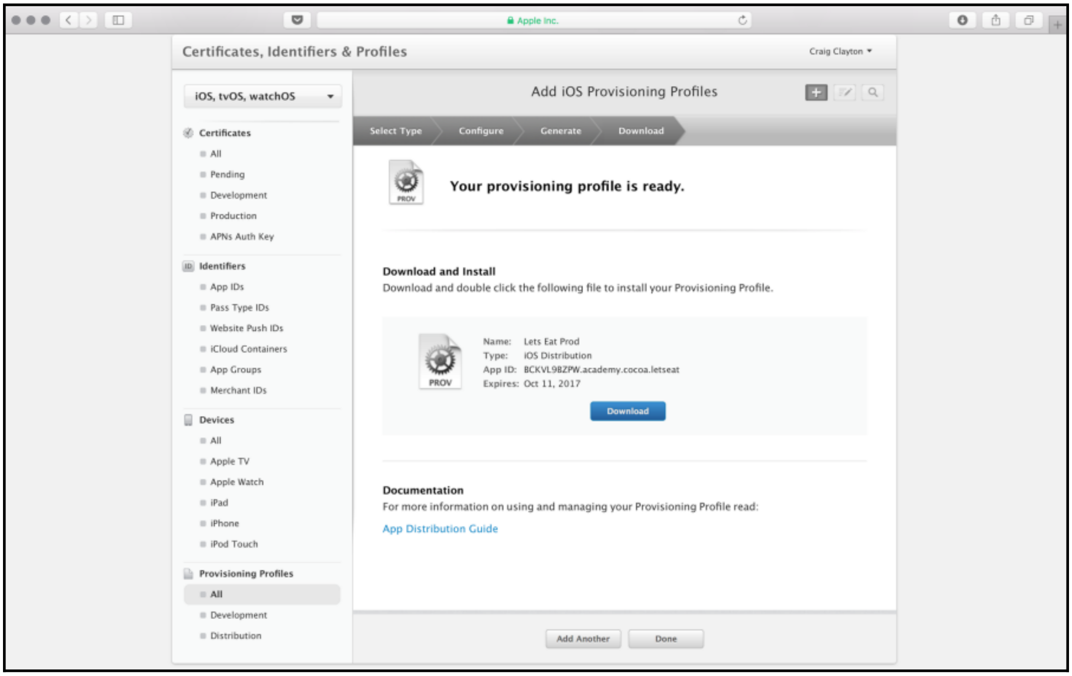
The name you provide will be used to identify the profile in the portal.

Profile Name: Lets Eat Prod

Type: iOS Distribution

App ID: Lets Eat (BCKVL9BZPW.academy.cocoa.letseat)

Certificates: 1 Included



Your provisioning profile is ready.

Download and Install

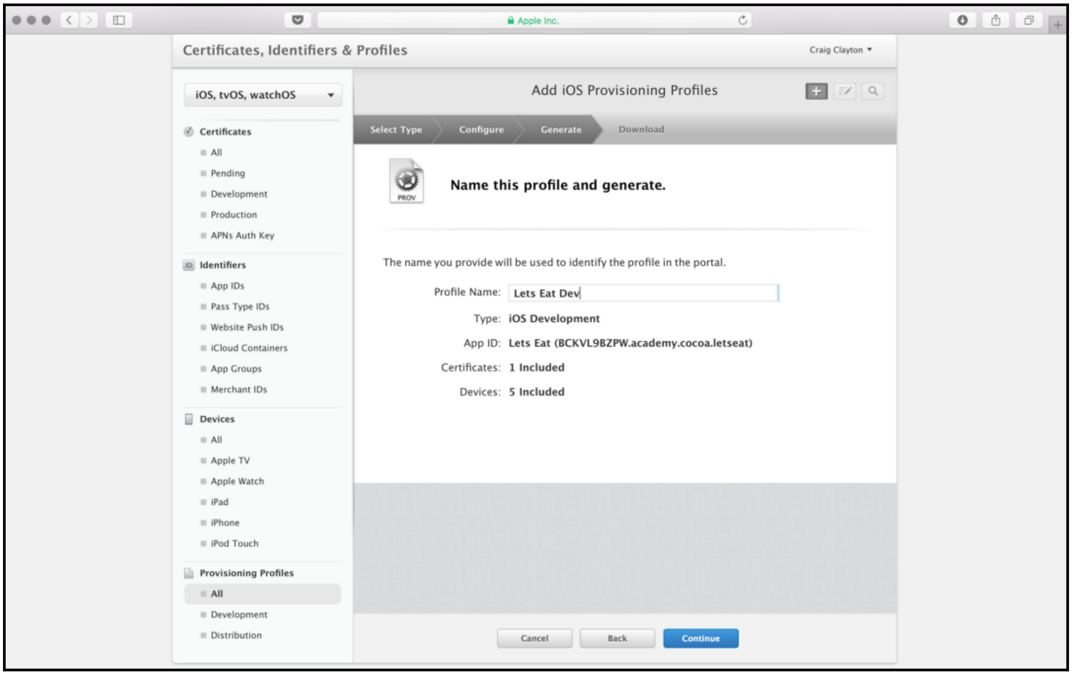
Download and double click the following file to install your Provisioning Profile.

Name: Lets Eat Prod
Type: iOS Distribution
App ID: BCKVL9BZPW.academy.cocoa.letseat
Expires: Oct 11, 2017

Documentation

For more information on using and managing your Provisioning Profile read:

App Distribution Guide



Name this profile and generate.

The name you provide will be used to identify the profile in the portal.

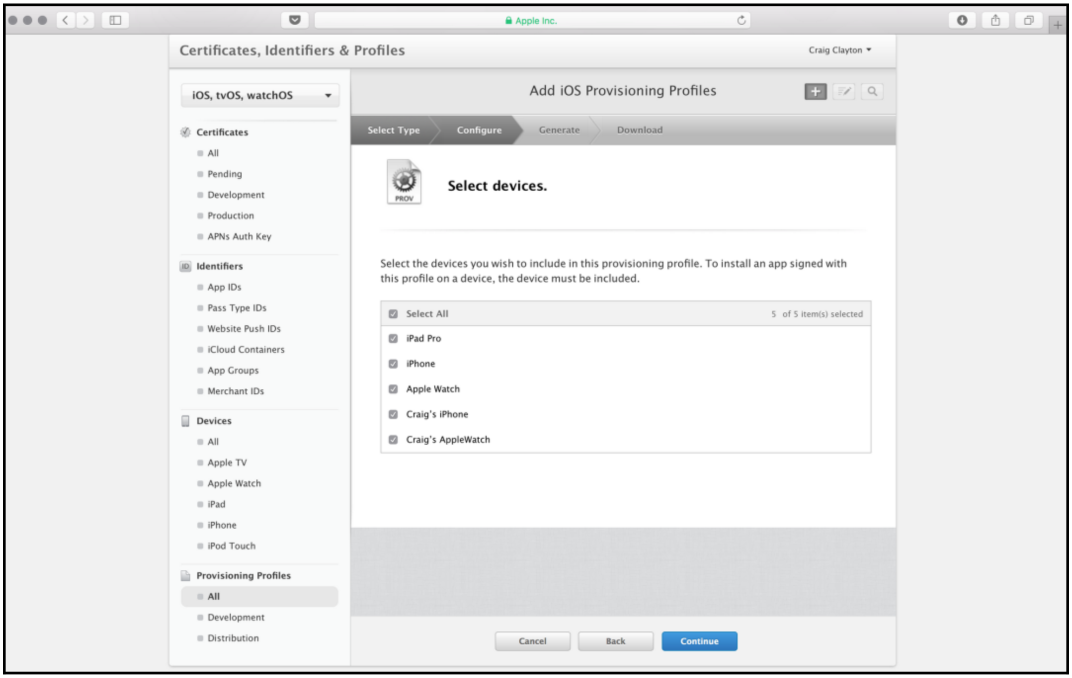
Profile Name: Lets Eat Dev

Type: iOS Development

App ID: Lets Eat (BCKVL9BZPW.academy.cocoa.letseat)

Certificates: 1 Included

Devices: 5 Included



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

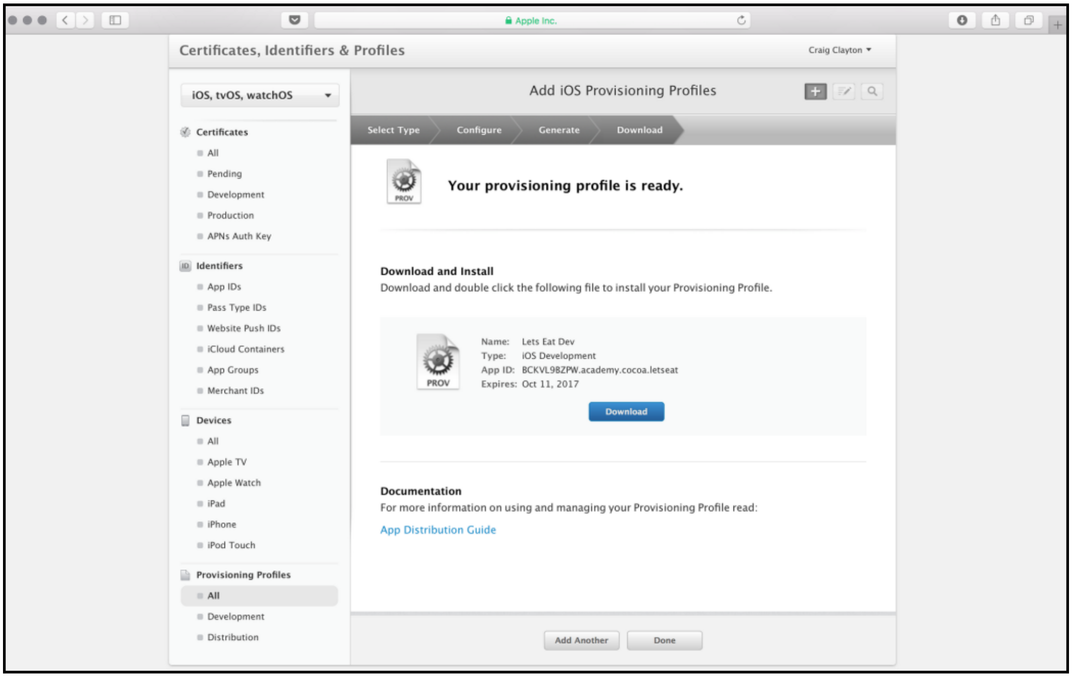
☑ iPad Pro

☑ iPhone

☑ Apple Watch

☑ Craig's iPhone

☑ Craig's AppleWatch



Your provisioning profile is ready.

Download and Install

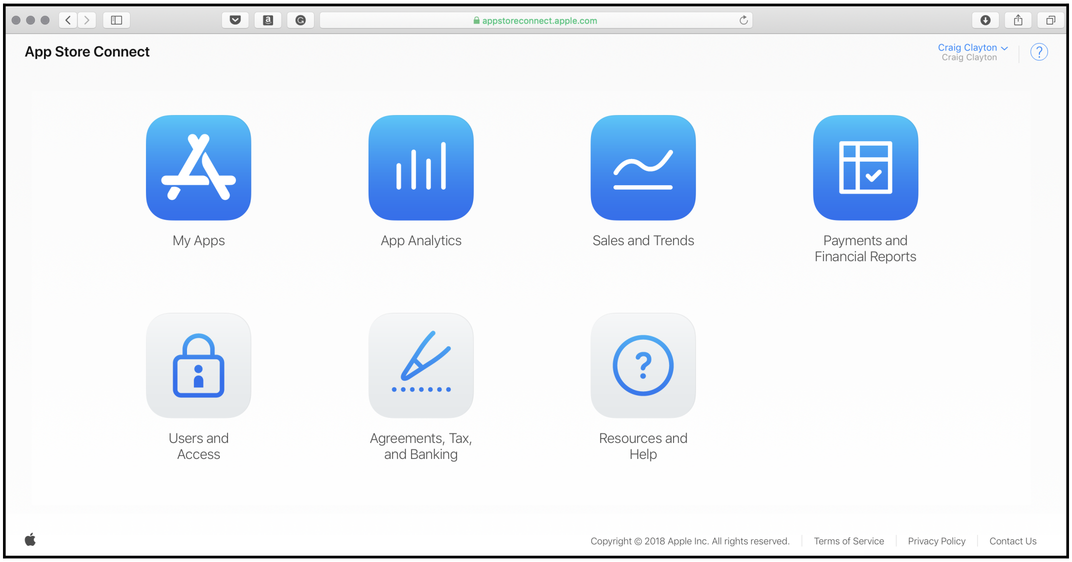
Download and double click the following file to install your Provisioning Profile.

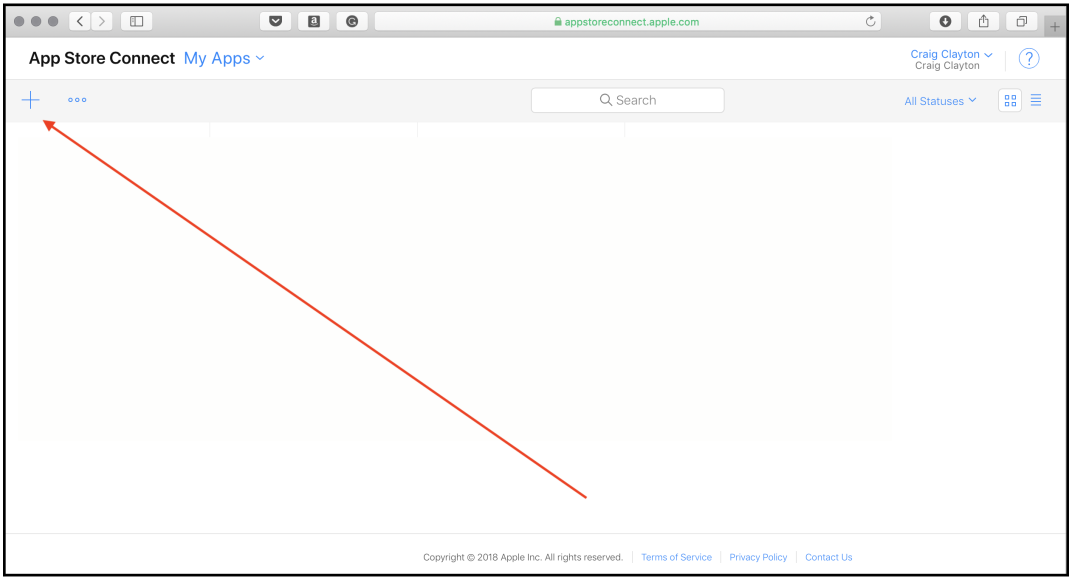
Name: Lets Eat Dev
Type: iOS Development
App ID: BCKVL9BZPW.academy.cocoa.letseat
Expires: Oct 11, 2017

Documentation

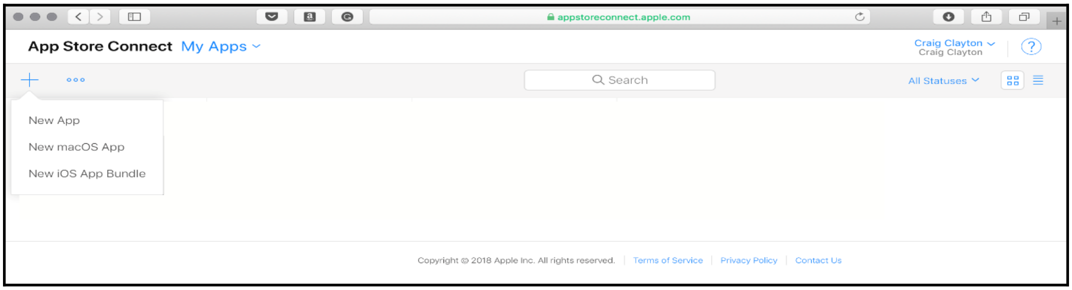
For more information on using and managing your Provisioning Profile read:

App Distribution Guide

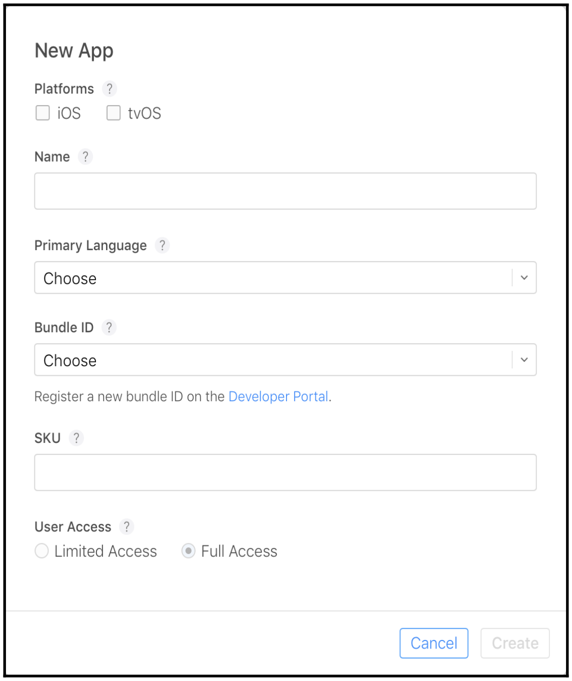




appstoreconnect.apple.com
App Store Connect
My Apps
Craig Clayton
Craig Clayton
Search
All Statuses
Copyright © 2018 Apple Inc. All rights reserved.
Terms of Service
Privacy Policy
Contact Us



appstoreconnect.apple.com
App Store Connect
My Apps
Craig Clayton
Craig Clayton
Search
All Statuses
New App
New macOS App
New iOS App Bundle
Copyright © 2018 Apple Inc. All rights reserved.
Terms of Service
Privacy Policy
Contact Us



New App

Platforms ?

□ iOS □ tvOS

Name ?

Primary Language ?

Choose

Bundle ID ?

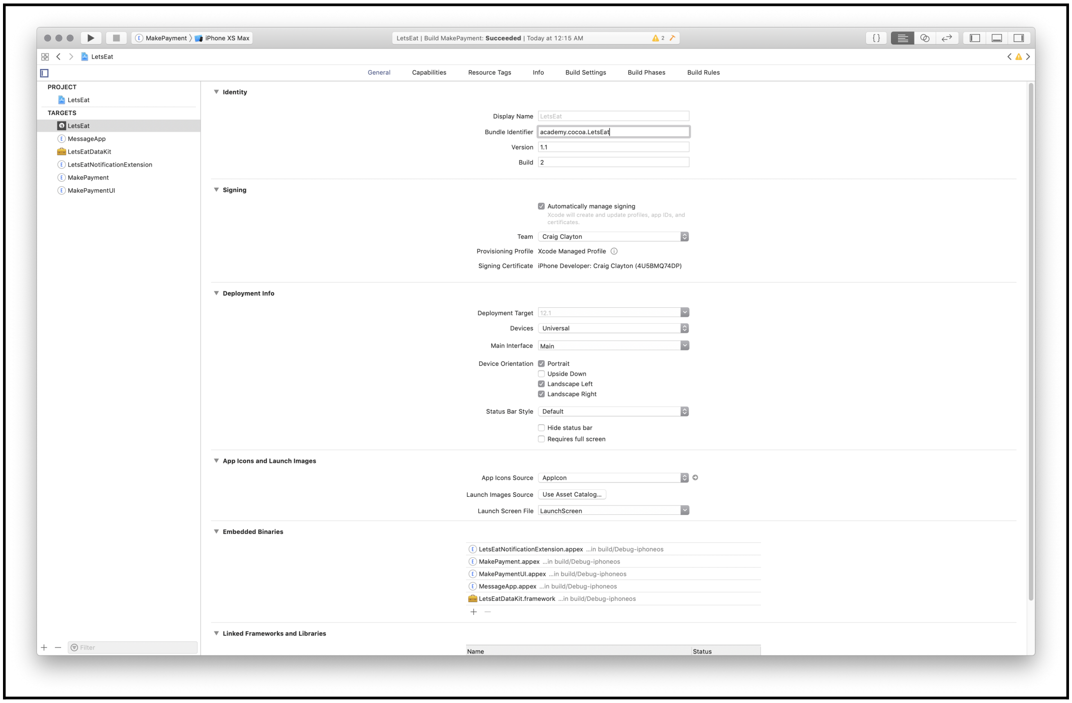
Choose

Register a new bundle ID on the Developer Portal.

SKU ?

User Access ?

○ Limited Access ● Full Access



Identity

Display Name LetsEat

Bundle Identifier academy.cocoa.LetsEat

Version 1.1

Build 2

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 12.1

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

LetsEatNotificationExtension.appex ...in build/Debug-iphoneos

MakePayment.appex ...in build/Debug-iphoneos

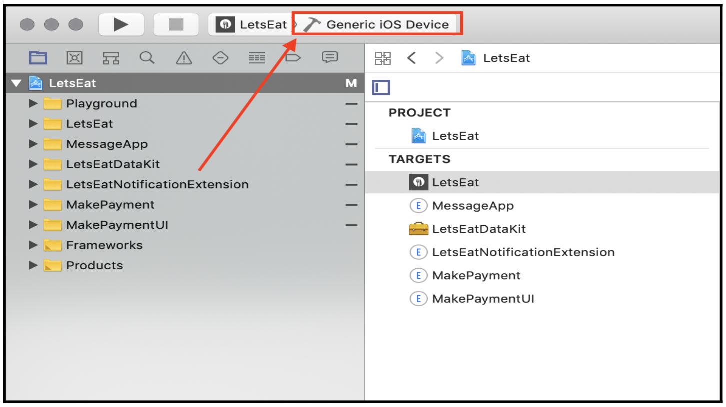
MakePaymentUI.appex ...in build/Debug-iphoneos

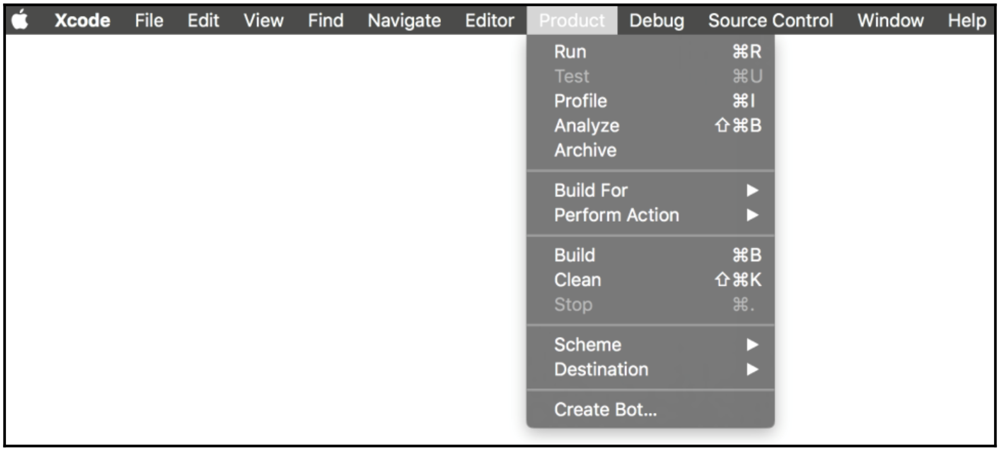
MessageApp.appex ...in build/Debug-iphoneos

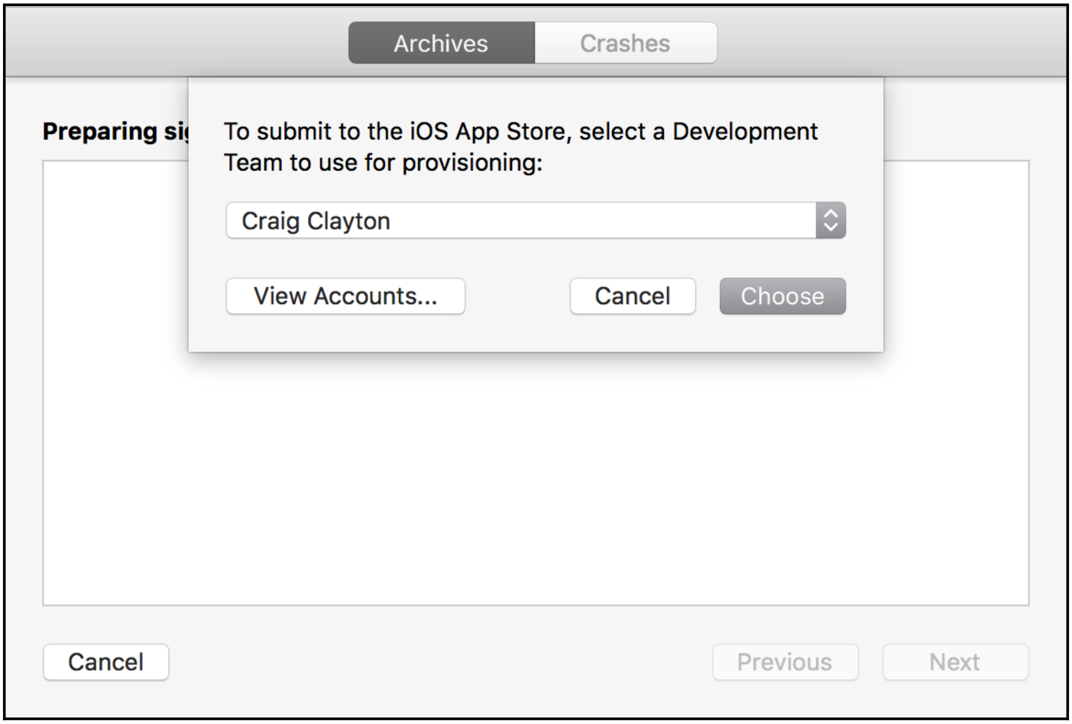
LetsEatDataKit.framework ...in build/Debug-iphoneos

Linked Frameworks and Libraries

Name	Status

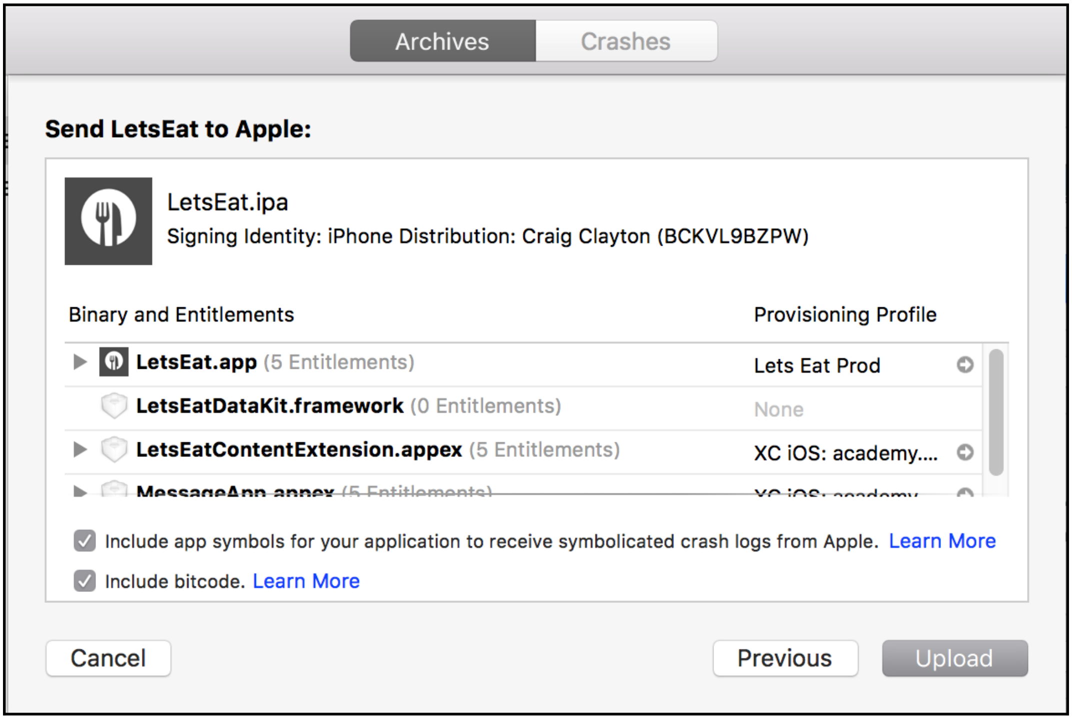






To submit to the iOS App Store, select a Development Team to use for provisioning:

Craig Clayton



Send LetsEat to Apple:

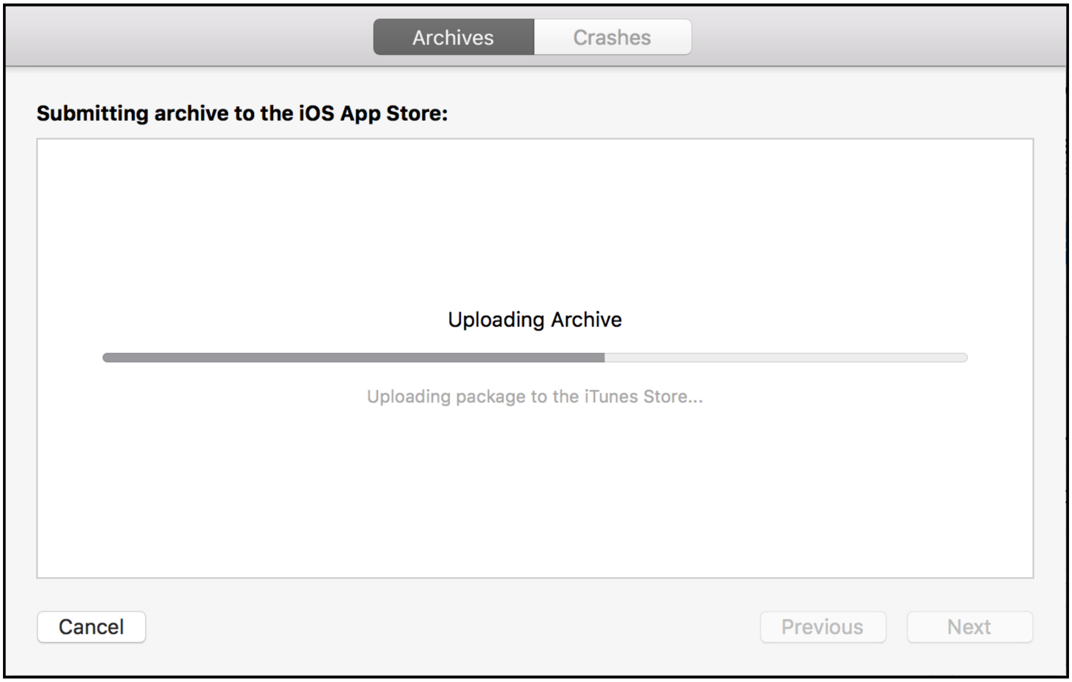
LetsEat.ipa

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....
▶ **MessageApp.appex** (5 Entitlements)	XC iOS: academy

☑ Include app symbols for your application to receive symbolicated crash logs from Apple. Learn More

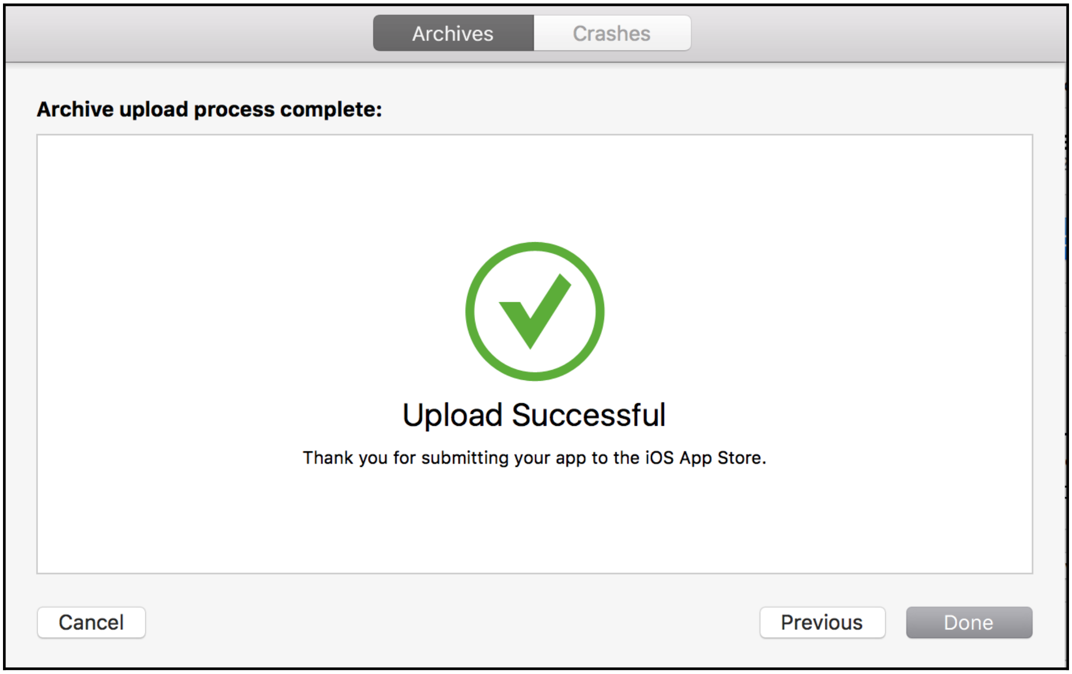
☑ Include bitcode. Learn More



Submitting archive to the iOS App Store:

Uploading Archive

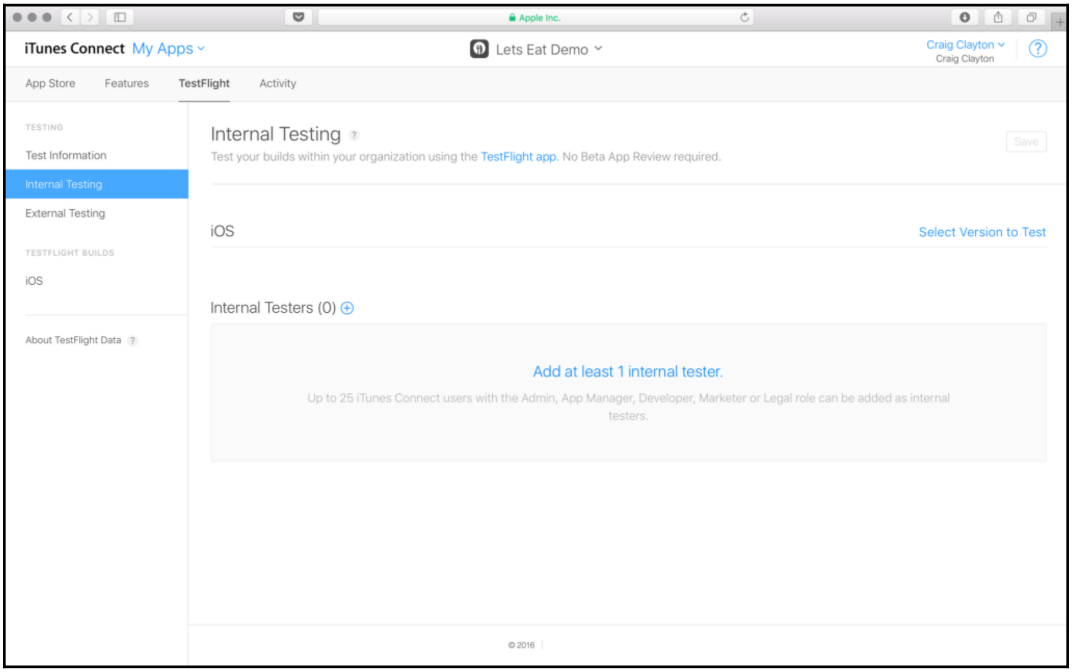
Uploading package to the iTunes Store...



Archive upload process complete:

Upload Successful

Thank you for submitting your app to the iOS App Store.

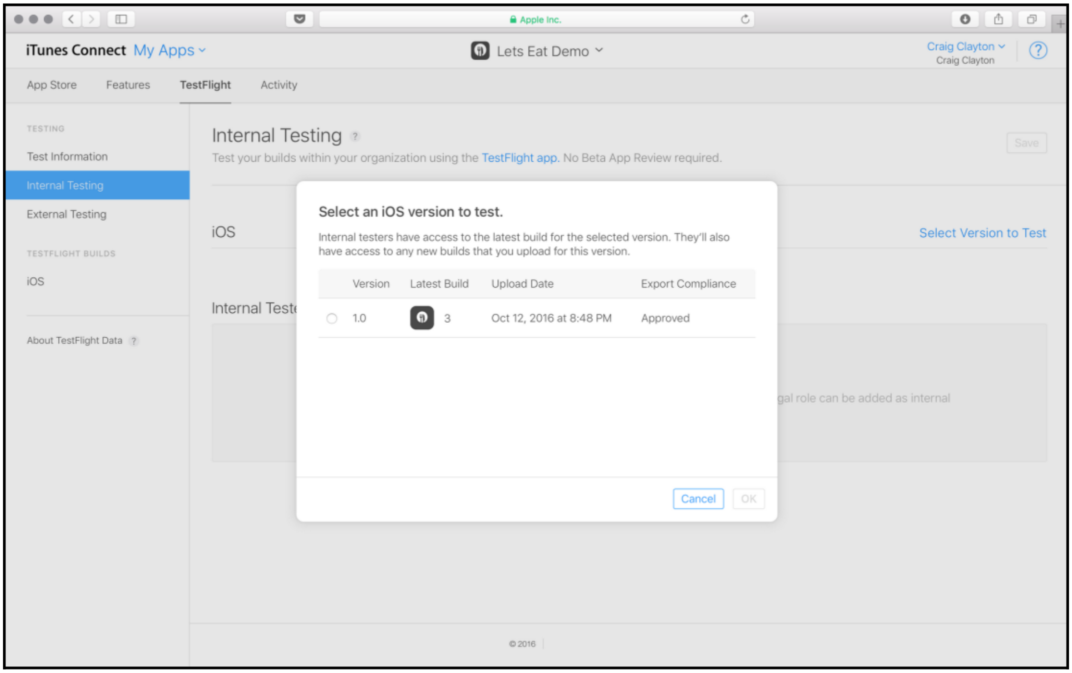


Internal Testing

Test your builds within your organization using the TestFlight app. No Beta App Review required.

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.



Internal Testing ?

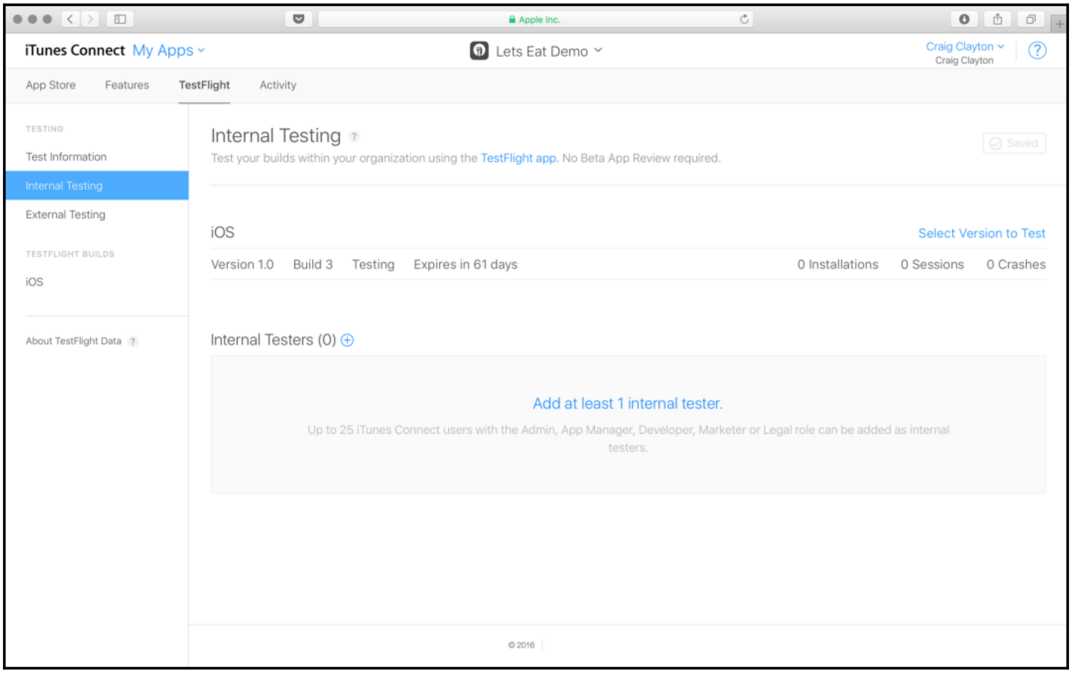
Test your builds within your organization using the TestFlight app. No Beta App Review required.

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



Internal Testing ?

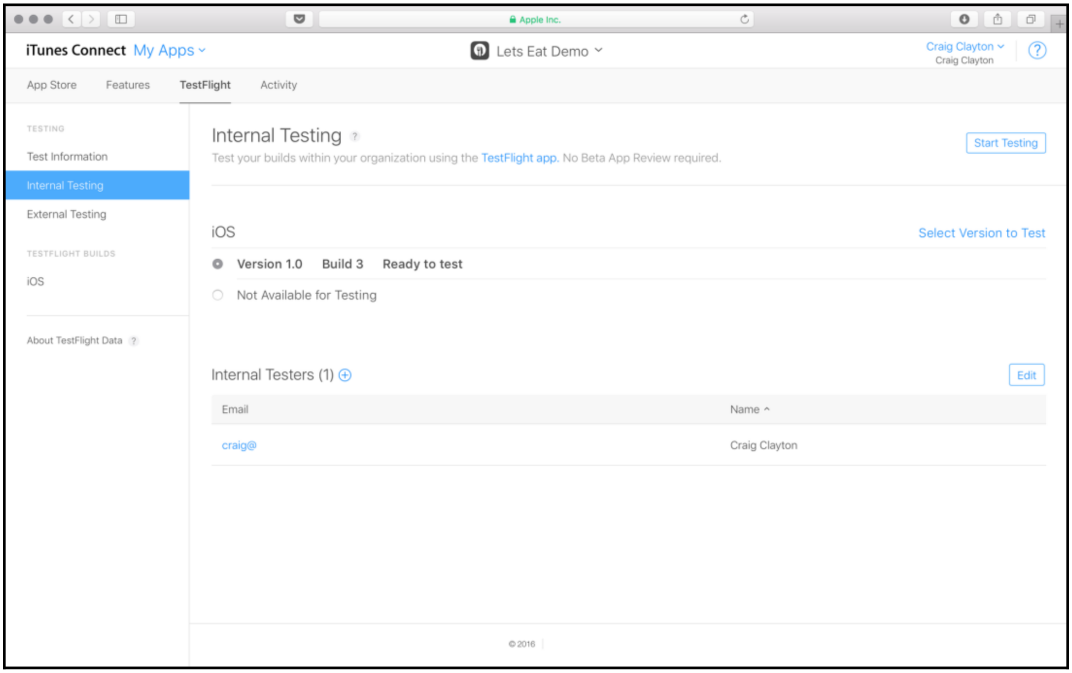
Test your builds within your organization using the TestFlight app. No Beta App Review required.

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.



Internal Testing

Test your builds within your organization using the TestFlight app. No Beta App Review required.

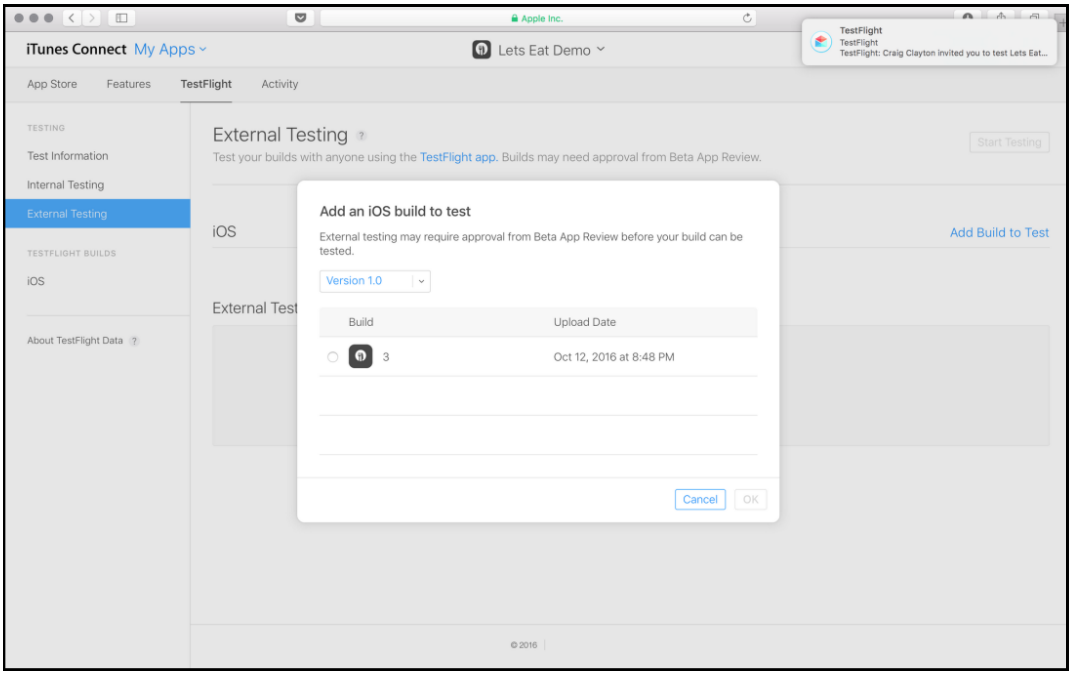
iOS

● Version 1.0 Build 3 Ready to test

○ Not Available for Testing

Internal Testers (1)

Email	Name
craig@	Craig Clayton



TestFlight
TestFlight
TestFlight: Craig Clayton invited you to test Lets Eat...

External Testing

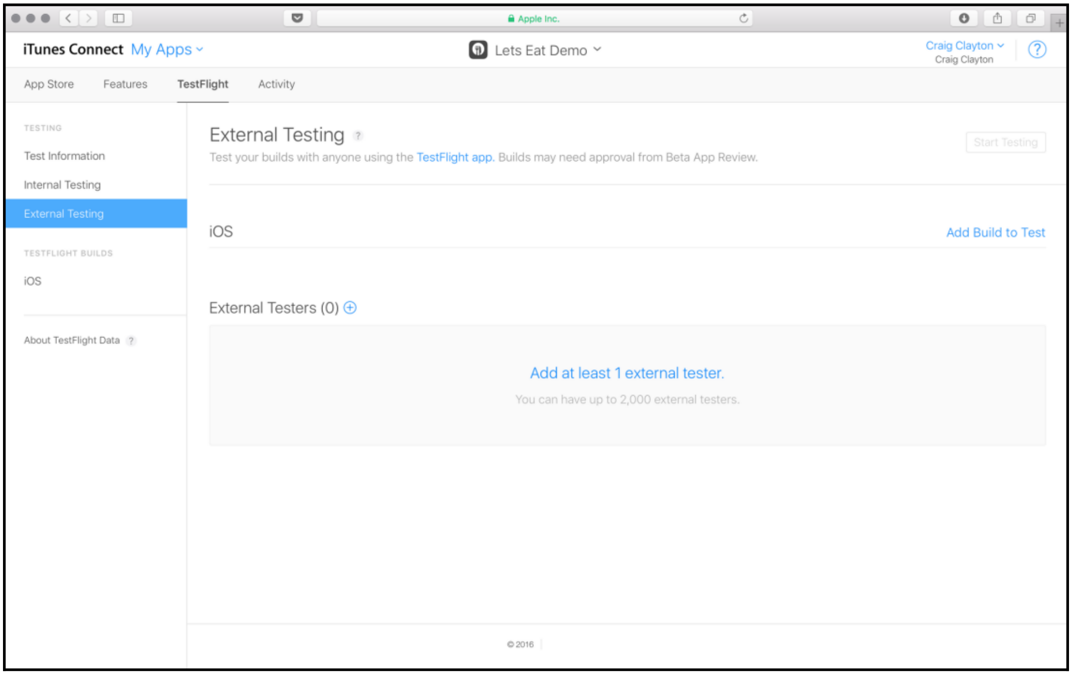
Test your builds with anyone using the TestFlight app. Builds may need approval from Beta App Review.

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



External Testing

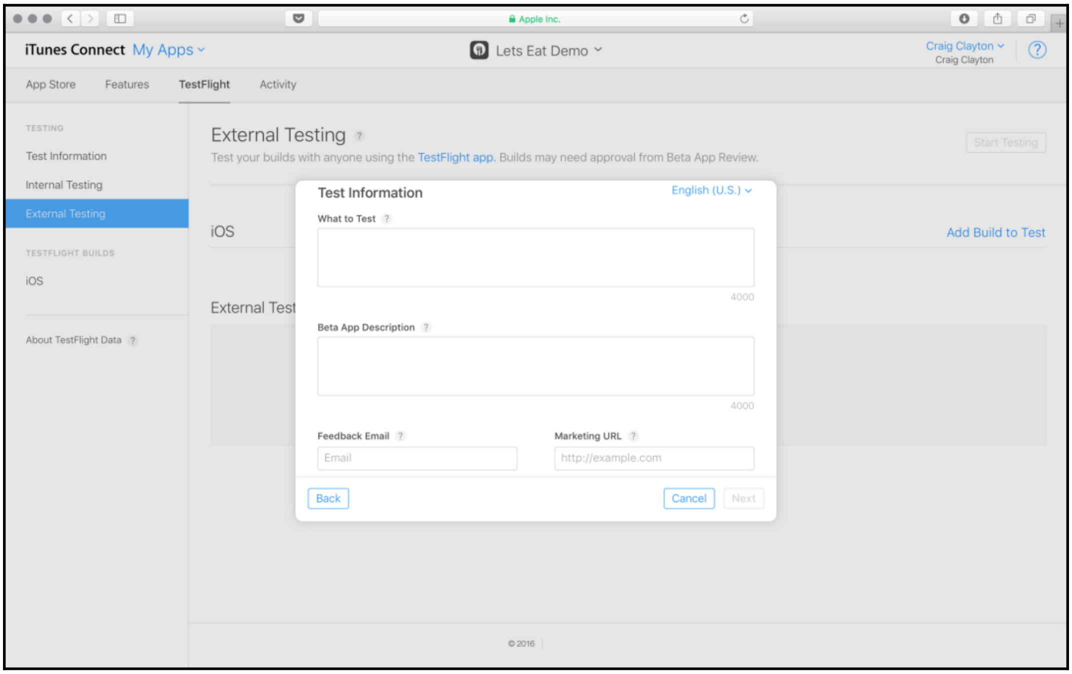
Test your builds with anyone using the TestFlight app. Builds may need approval from Beta App Review.

iOS

External Testers (0)

Add at least 1 external tester.

You can have up to 2,000 external testers.



External Testing

Test your builds with anyone using the TestFlight app. Builds may need approval from Beta App Review.

iOS

External Test

Test Information

What to Test

4000

Beta App Description

4000

Feedback Email: Email

Marketing URL: http://example.com

Index