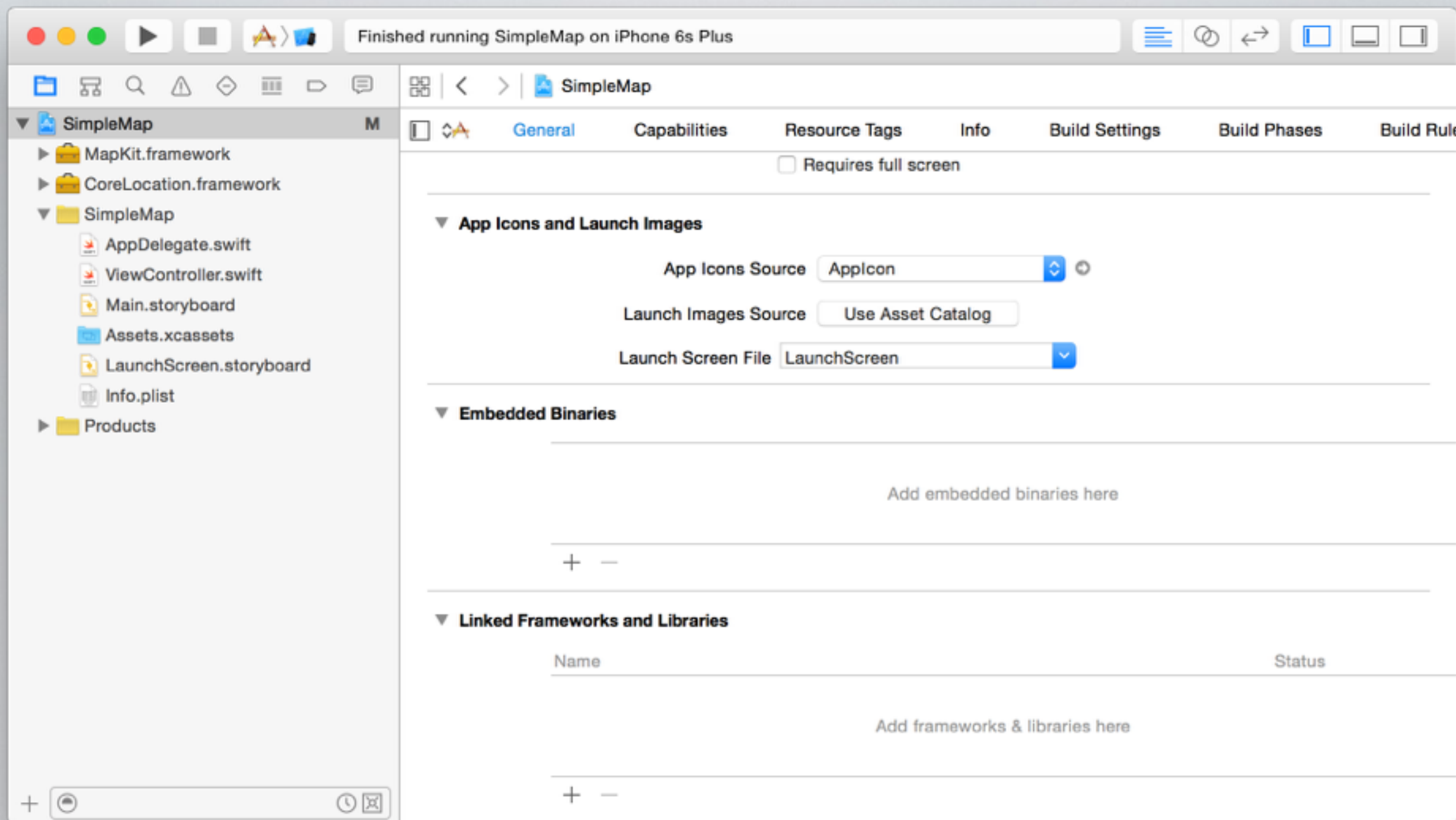
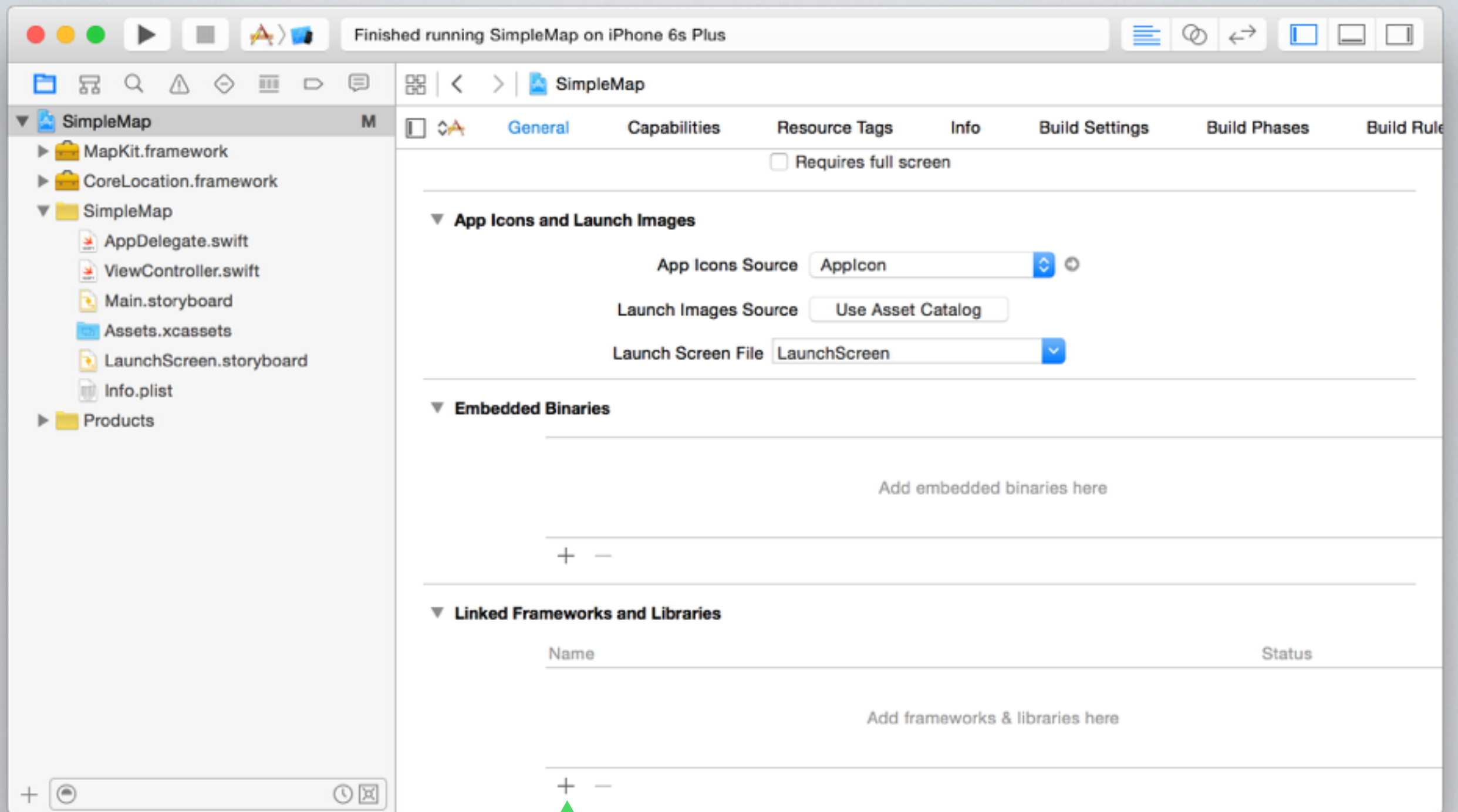


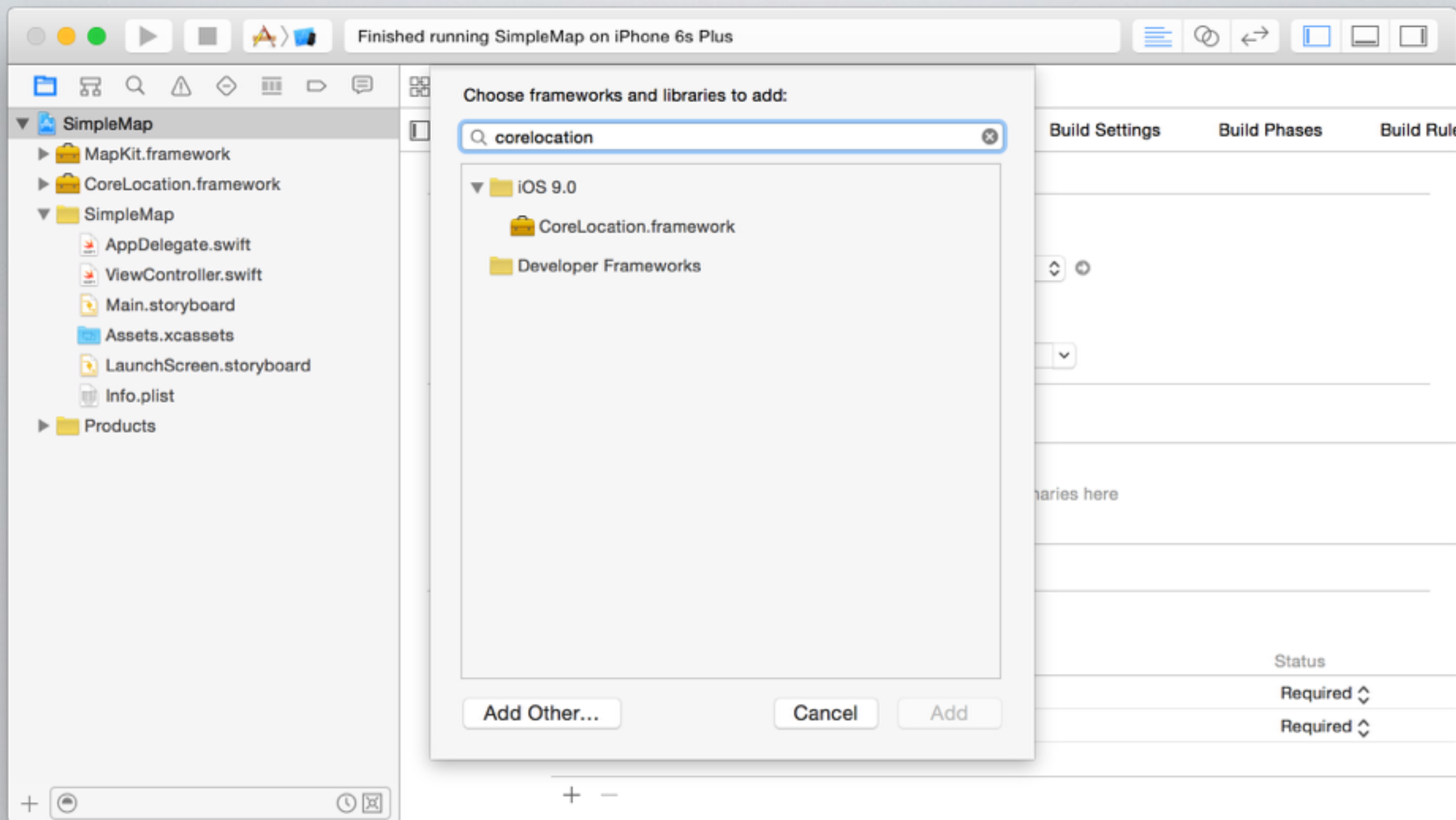
# CORELOCATION AND MAPKIT

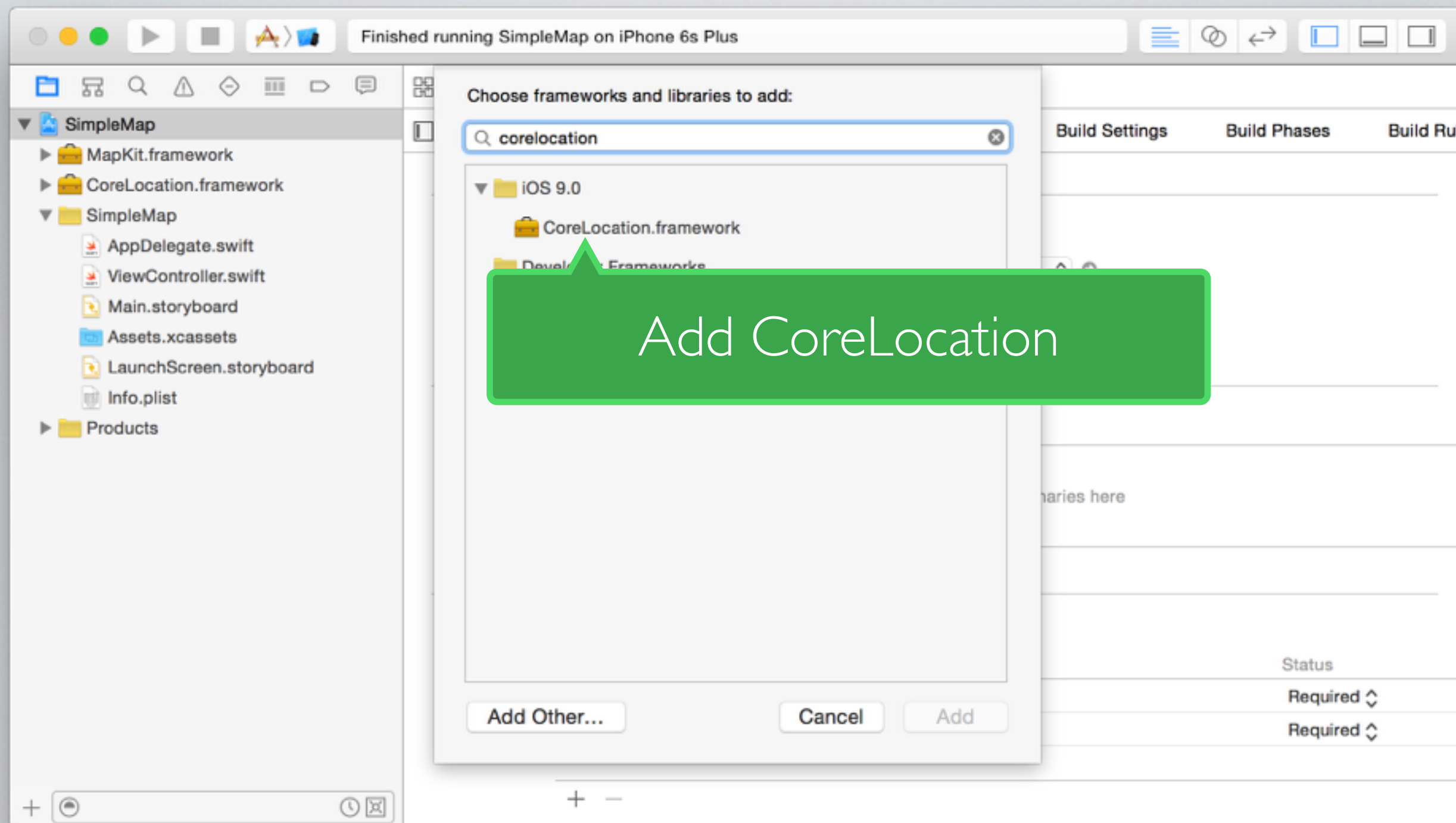
Nick Chen  
Fall 2015  
[talentspark.io](http://talentspark.io)



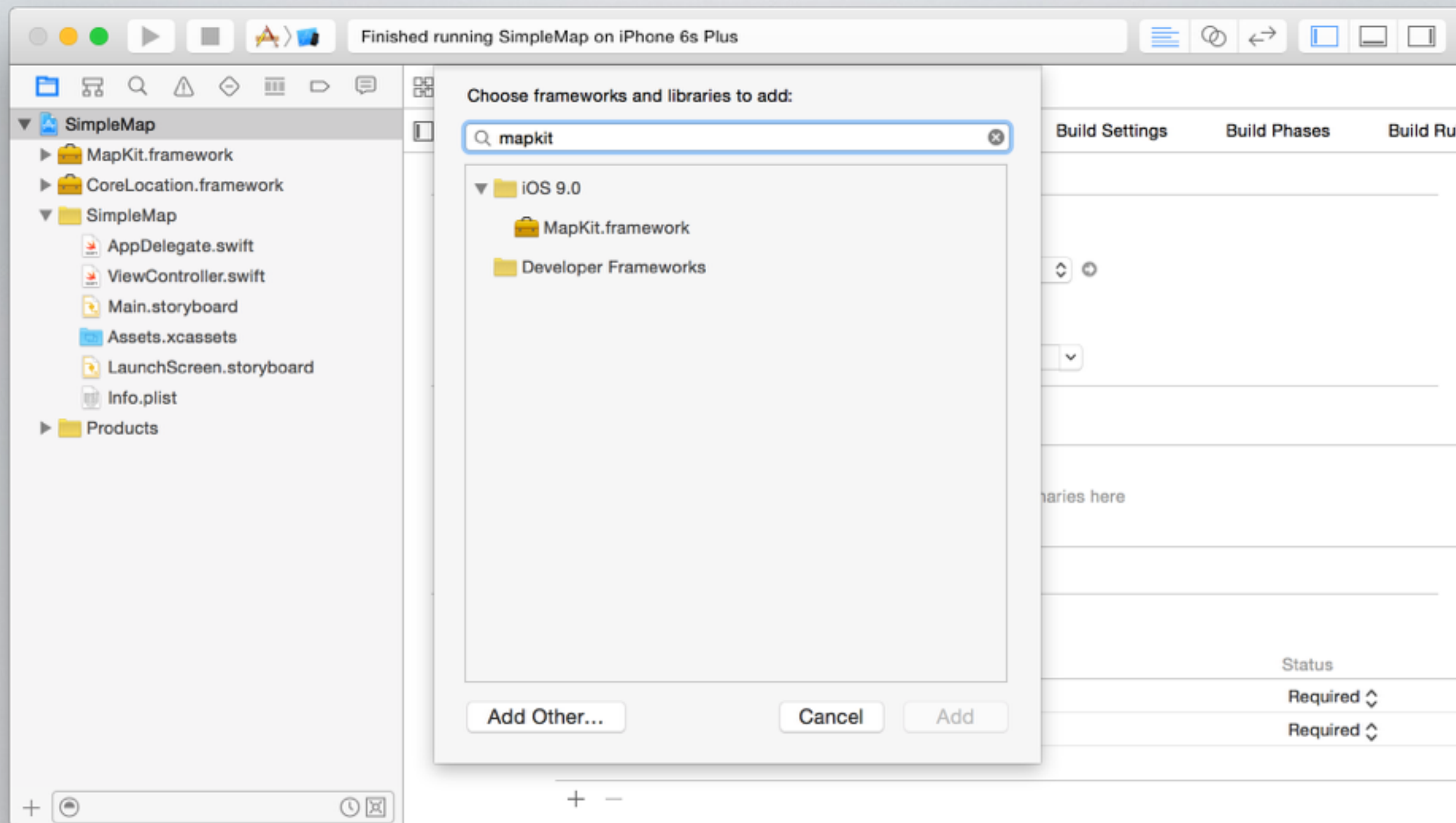


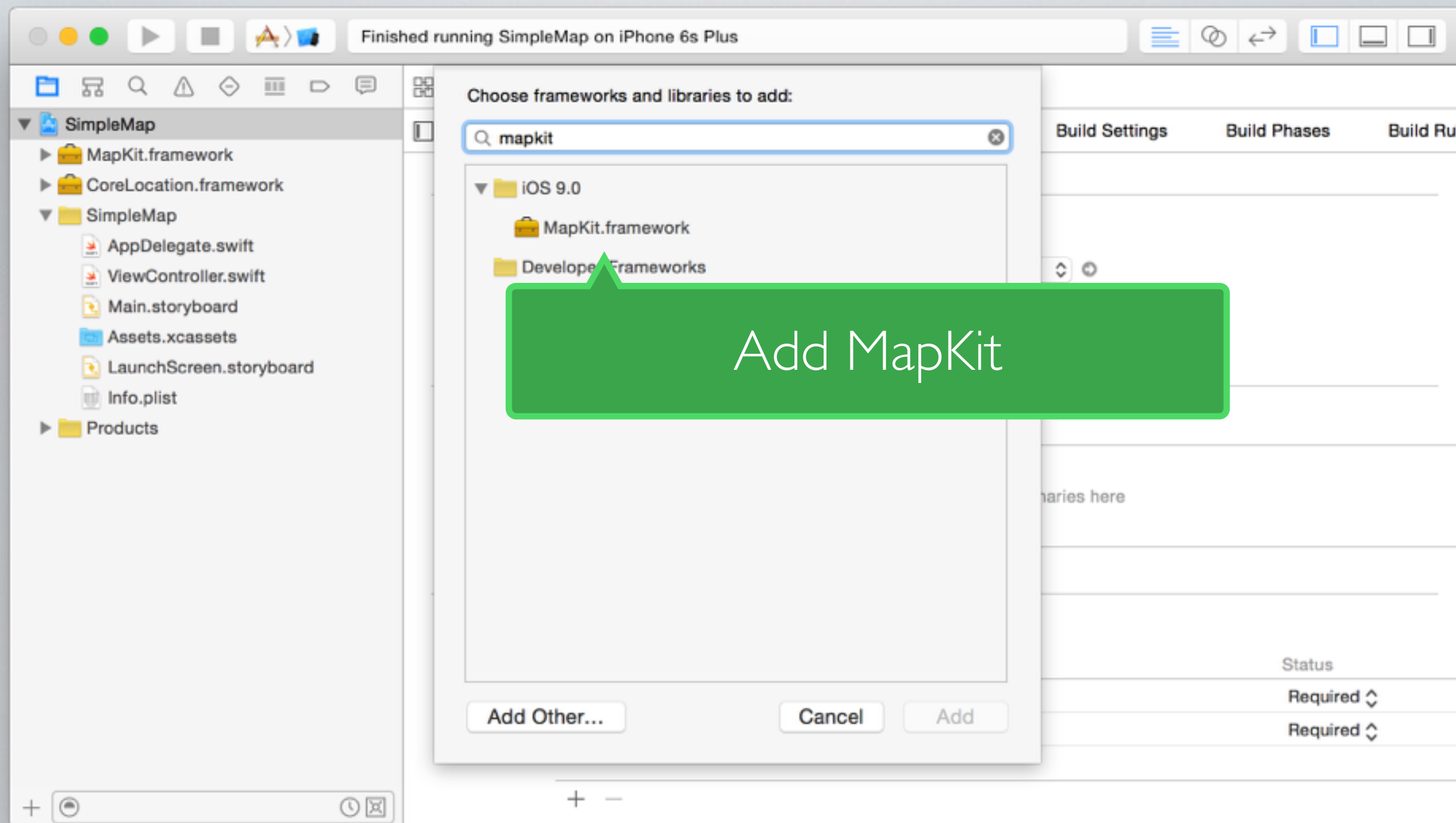
Click "+" to add frameworks

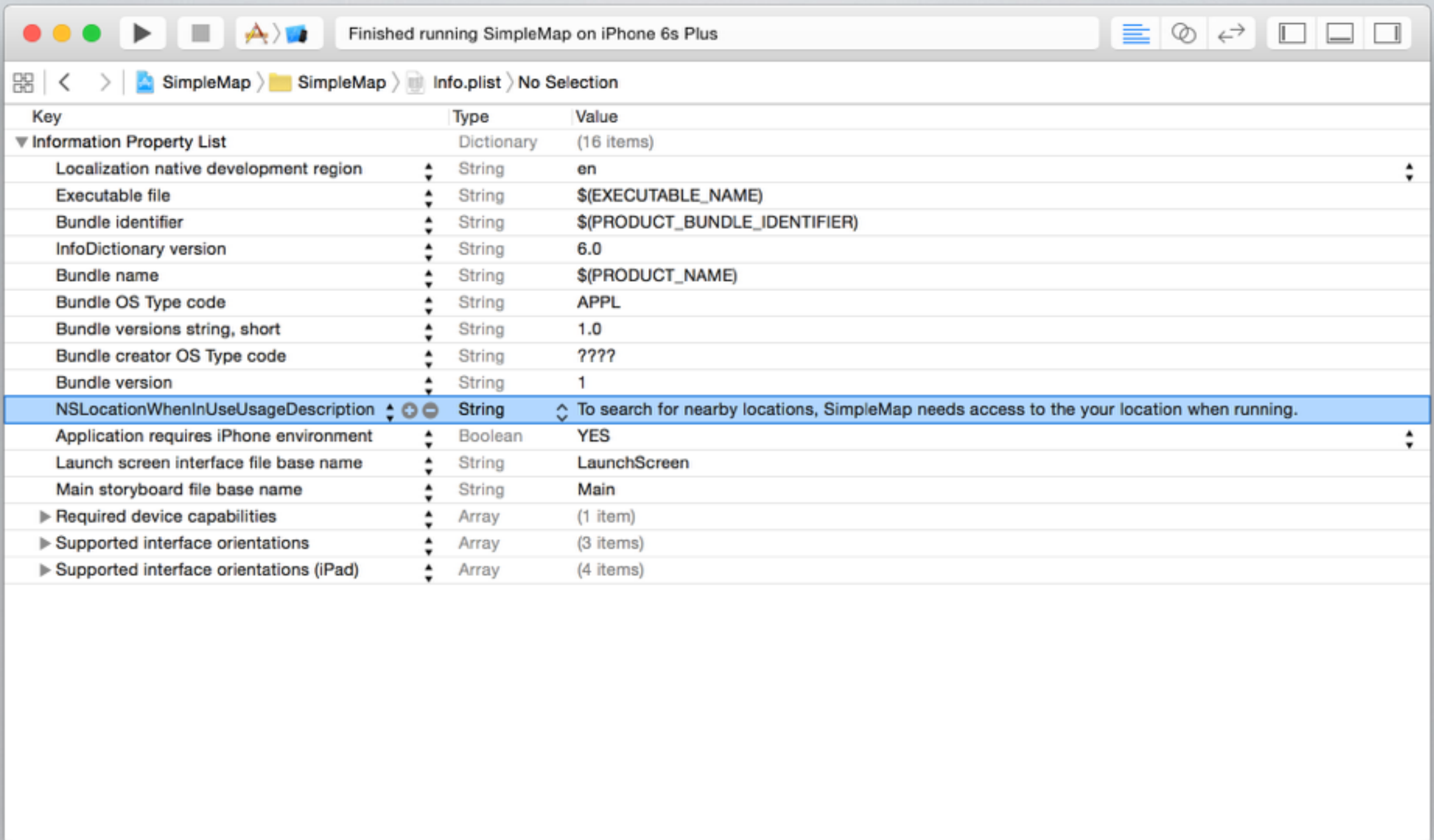












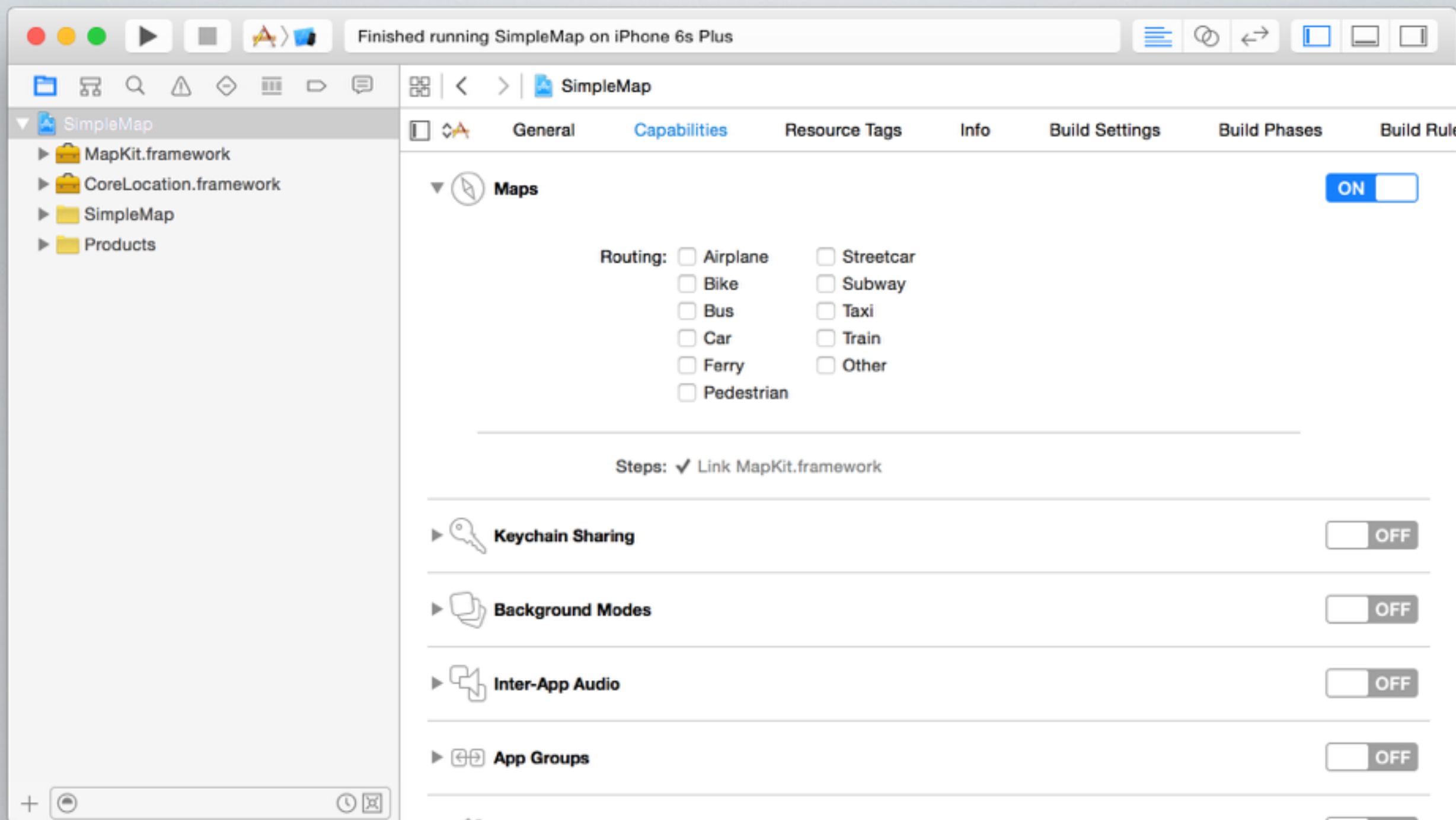


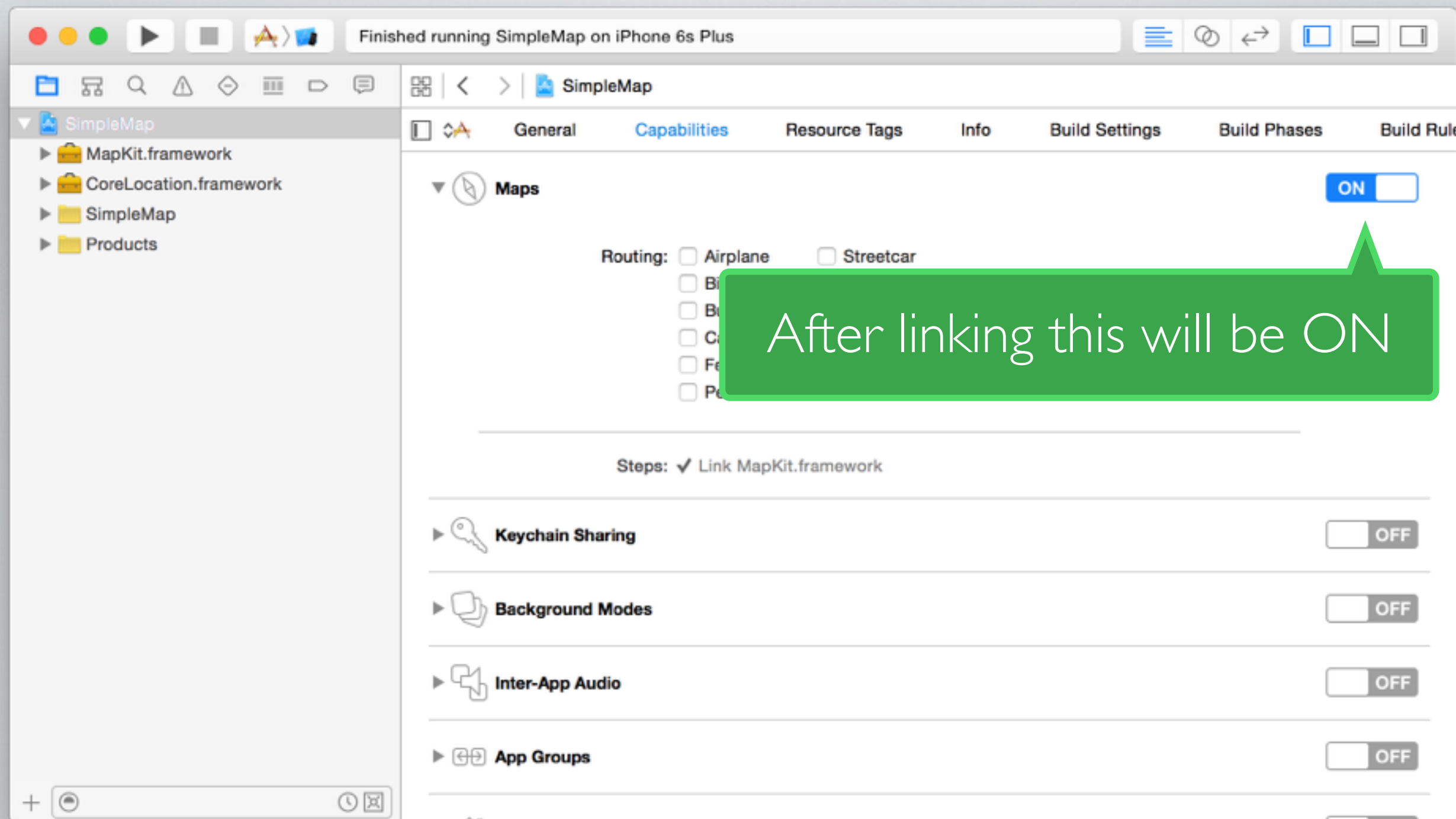
Finished running SimpleMap on iPhone 6s Plus

SimpleMap > SimpleMap > Info.plist > No Selection

Key	Type	Value
Information Property List	Dictionary	(16 items)
Localization native development region	String	en
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
NSLocationWhenInUseUsageDescription	String	To search for nearby locations, SimpleMap needs access to the your location when running.
Application requires iPhone environment	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main		
▶ Requir		
▶ Supp		
▶ Supp		

Need to ask for permission





Finished running SimpleMap on iPhone 6s Plus

SimpleMap > SimpleMap > ViewController.swift > ViewController

```
15 @IBOutlet weak var searchBar: UISearchBar!
16
17 let locationManager = CLLocationManager()
18
19 override func viewDidLoad() {
20     super.viewDidLoad()
21
22     self.locationManager.delegate = self
23     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest
24     self.locationManager.requestWhenInUseAuthorization()
25     self.locationManager.startUpdatingLocation()
26     self.mapView.showsUserLocation = true
27 }
28
29 override func didReceiveMemoryWarning() {
30     super.didReceiveMemoryWarning()
31     // Dispose of any resources that can be recreated.
32 }
33
34 // MARK: CLLocationManagerDelegate
35 func locationManager(manager: CLLocationManager, didUpdateLocations locations:
    [CLLocation]) {
36     let location = locations.last
```



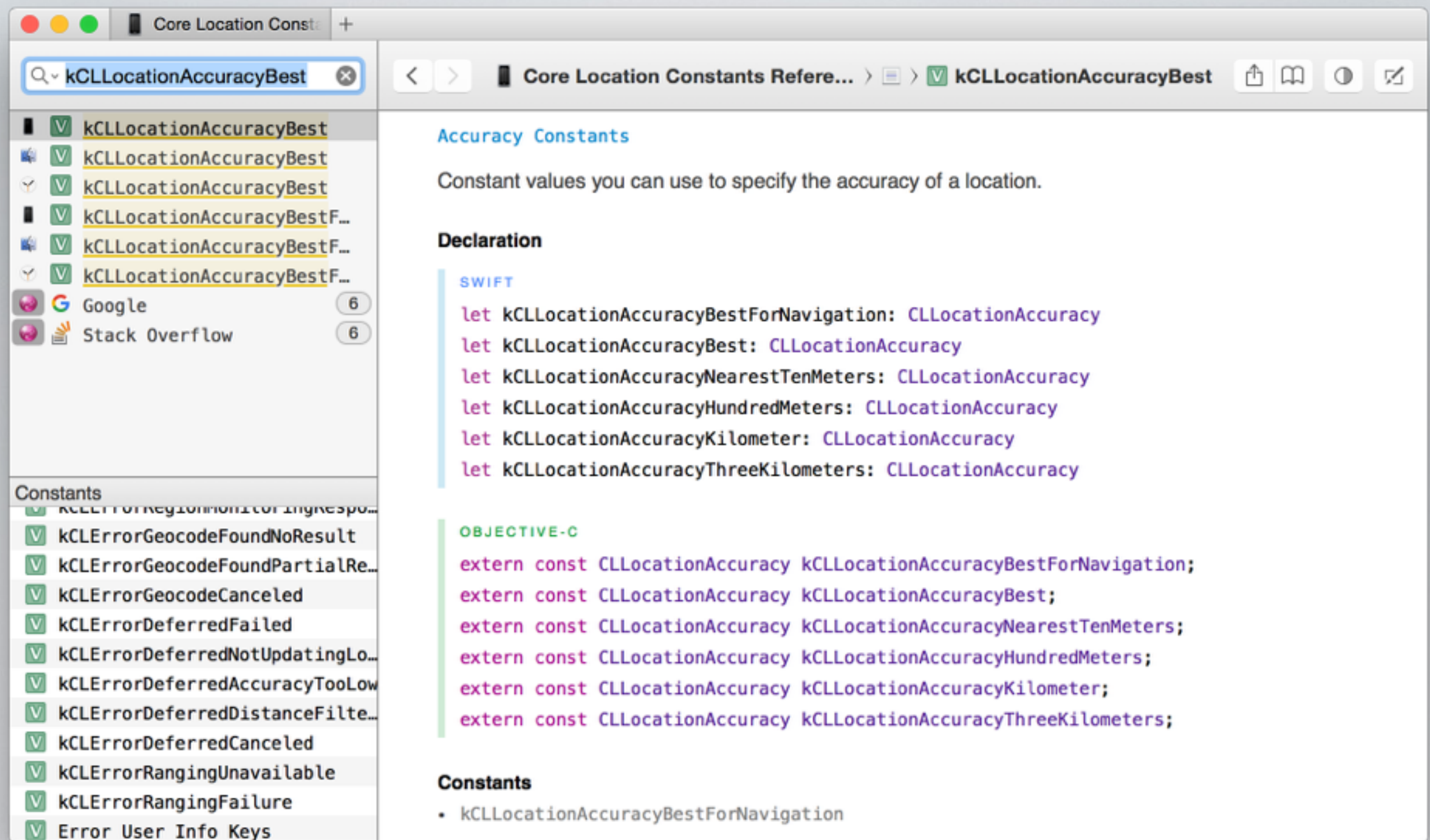
Finished running SimpleMap on iPhone 6s Plus

SimpleMap > SimpleMap > ViewController.swift > ViewController

```
15 @IBOutlet weak var searchBar: UISearchBar!
16
17 let locationManager = CLLocationManager()
18
19 override func viewDidLoad() {
20     super.viewDidLoad()
21
22     self.locationManager.delegate = self
23     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest
24     self.locationManager.requestWhenInUseAuthorization()
25     self.locationManager
26     self.mapView.showsUs
27 }
28
29 override func didReceiveMemoryWarning() {
30     super.didReceiveMemoryWarning()
31     // Dispose of any resources that can be recreated.
32 }
33
34 // MARK: CLLocationManagerDelegate
35 func locationManager(manager: CLLocationManager, didUpdateLocations locations:
    [CLLocation]) {
36     let location = locations.last
```

Desired accuracy





Finished running SimpleMap on iPhone 6s Plus

SimpleMap > SimpleMap > ViewController.swift > ViewController

```
15 @IBOutlet weak var searchBar: UISearchBar!
16
17 let locationManager = CLLocationManager()
18
19 override func viewDidLoad() {
20     super.viewDidLoad()
21
22     self.locationManager.delegate = self
23     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest
24     self.locationManager.requestWhenInUseAuthorization()
25     self.locationManager.startUpdatingLocation()
26     self.mapView.showsUserLocation = true
27 }
28
29 override func didReceiveMemoryWarning() {
30     super.didReceiveMemoryWarning()
31     // Dispose of any resources that can be recreated.
32 }
33
34 // MARK: CLLocationManagerDelegate
35 func locationManager(manager: CLLocationManager, didUpdateLocations locations:
    [CLLocation]) {
36     let location = locations.last
```



Finished running SimpleMap on iPhone 6s Plus

SimpleMap > SimpleMap > ViewController.swift > ViewController

```
15 @IBOutlet weak var searchBar: UISearchBar!
16
17 let locationManager = CLLocationManager()
18
19 override func viewDidLoad() {
20     super.viewDidLoad()
21
22     self.locationManager.delegate = self
23     self.locationManager.desiredAccuracy = kCLLocationAccuracyBest
24     self.locationManager.requestWhenInUseAuthorization()
25     self.locationManager.startUpdatingLocation()
26     self.mapView.showsUserLocation
27 }
28
29 override func didReceiveMemoryWarning() {
30     super.didReceiveMemoryWarning()
31     // Dispose of any resources that can be recreated.
32 }
33
34 // MARK: CLLocationManagerDelegate
35 func locationManager(manager: CLLocationManager, didUpdateLocations locations:
    [CLLocation]) {
36     let location = locations.last
```

Request Authorization

CLLocationManager

requestwhen

M requestWhenInUseAuthoriz...

M requestWhenInUseAuthoriz...

M requestMediaDataWhenR... (2)

M requestMediaDataWhenR... (2)

M fetchRequestWithEntityNa...

M fetchRequestWithEntityNa...

M fetchRequestWithEntityNa...

M beginRequestWithExtensio...

M requestWithURL:cachePoli...

M beginRequestWithExtensio...

M requestWithURL:cachePoli...

M beginRequestWithExtensio...

Requesting Authorization for Location...

M - requestWhenInUseAuthorization

M - requestAlwaysAuthorization

Determining the Availability of Services

M + authorizationStatus

M + locationServicesEnabled

M + deferredLocationUpdatesAvai...

M + significantLocationChangeMo...

M + headingAvailable

M + isMonitoringAvailableForCla...

M + isRangingAvailable

Accessing the Delegate

CLLocationManager Class...

M - requestWhenInUseAuthoriz...

- requestWhenInUseAuthorization

Requests permission to use location services while the app is in the foreground.

Declaration

SWIFT

func requestWhenInUseAuthorization()

OBJECTIVE-C

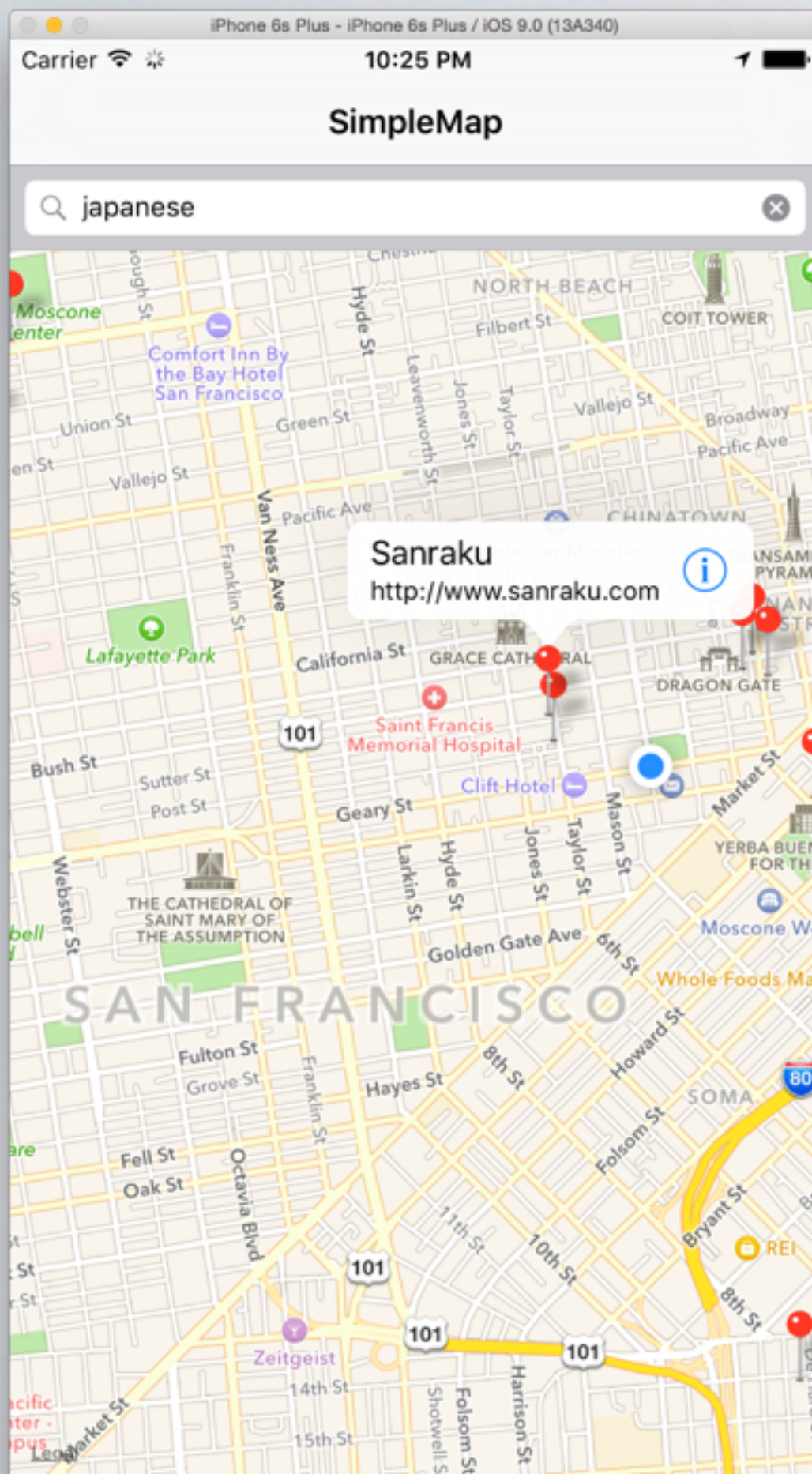
- (void)requestWhenInUseAuthorization

Discussion

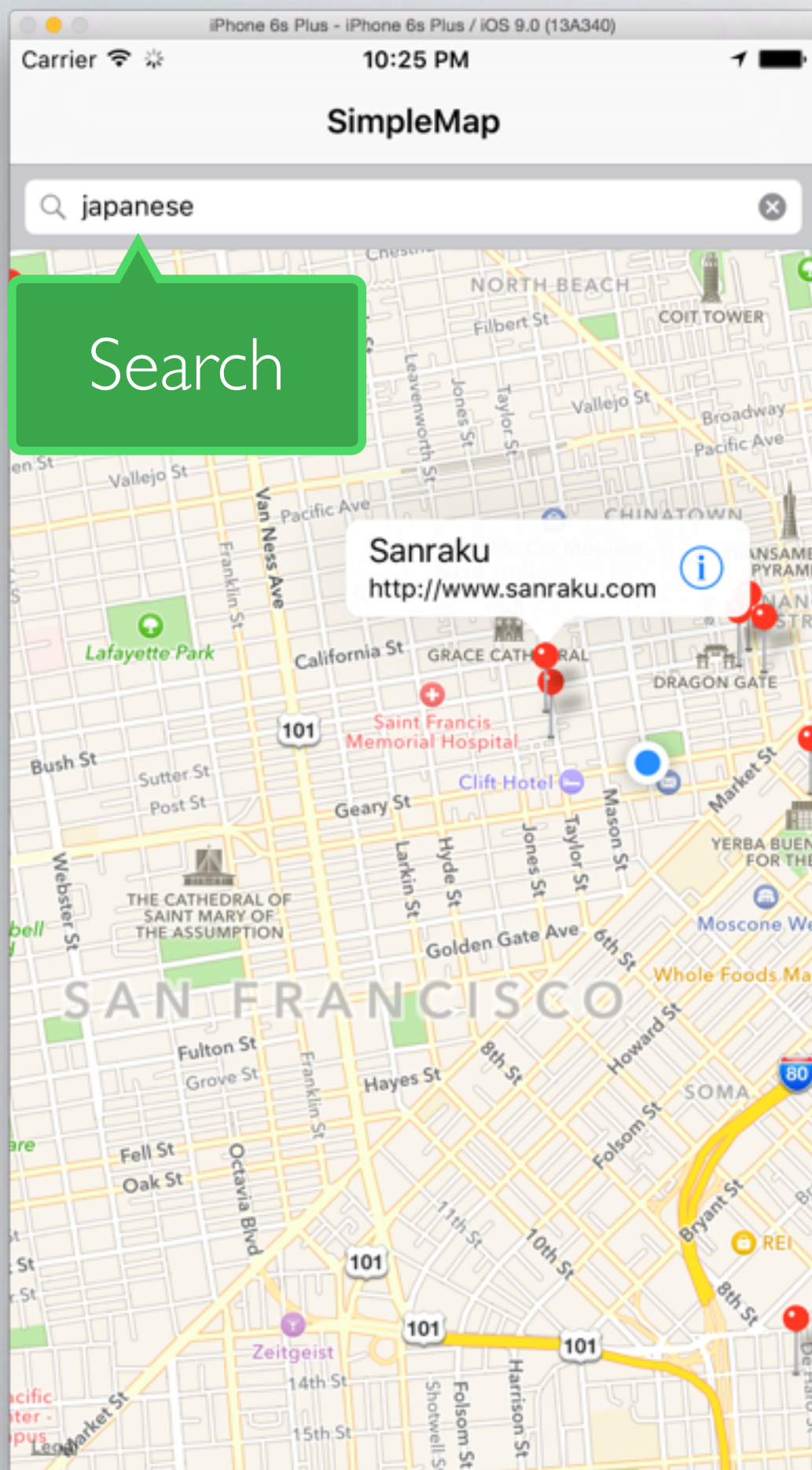
When the current authorization status is `kCLErrorAuthorizationStatusNotDetermined`, this method runs asynchronously and prompts the user to grant permission to the app to use location services. The user prompt contains the text from the `NSLocationWhenInUseUsageDescription` key in your app's `Info.plist` file, and the presence of that key is required when calling this method. After the status is determined, the location manager delivers the results to the delegate's `locationManager:didChangeAuthorizationStatus:` method. If the current authorization status is anything other than `kCLErrorAuthorizationStatusNotDetermined`, this method does nothing and does not call the `locationManager:didChangeAuthorizationStatus:` method.

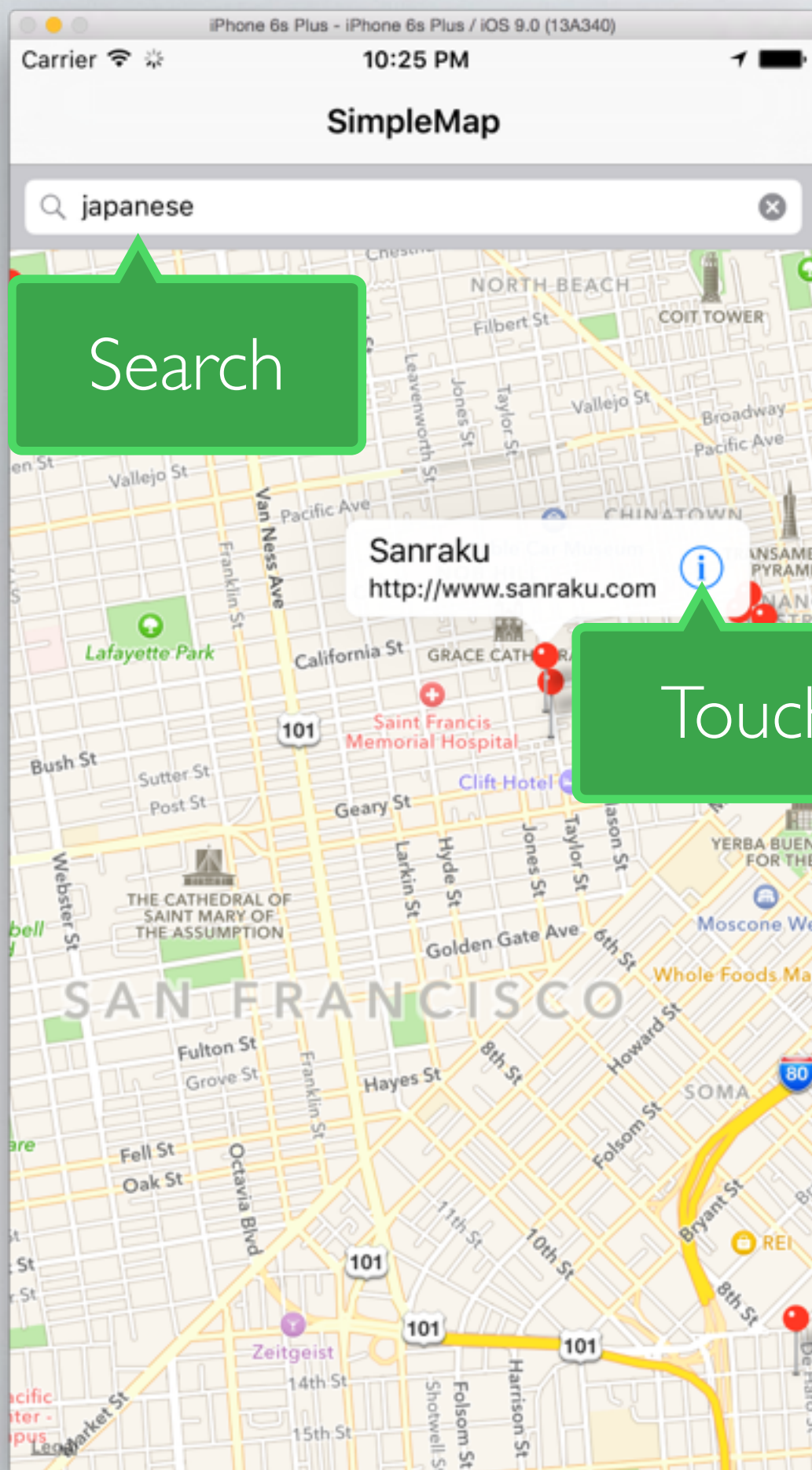
You must call this method or the `requestAlwaysAuthorization` method prior to using location services. If the user grants "when-in-use" authorization to your app, your app can start most (but



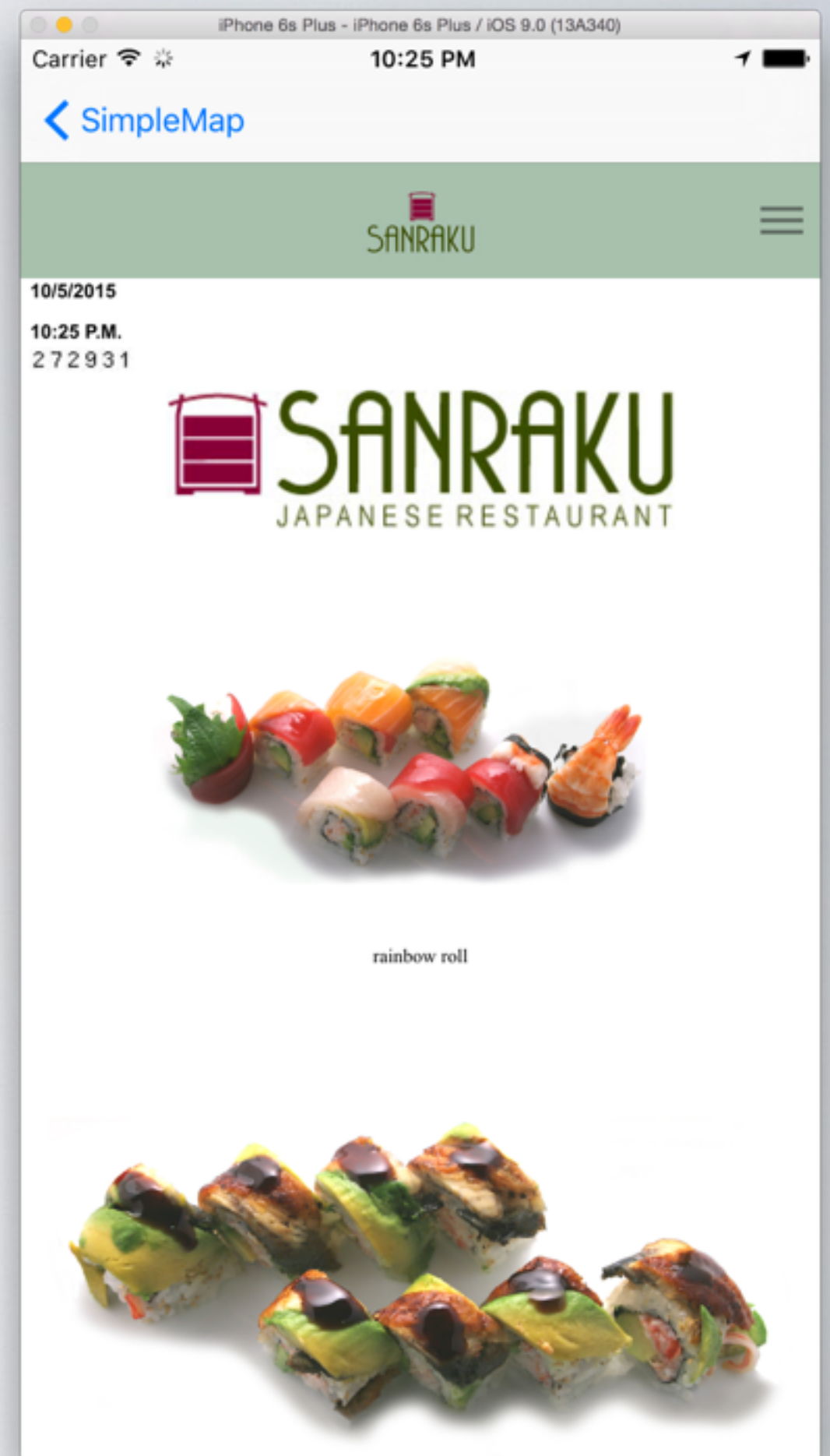
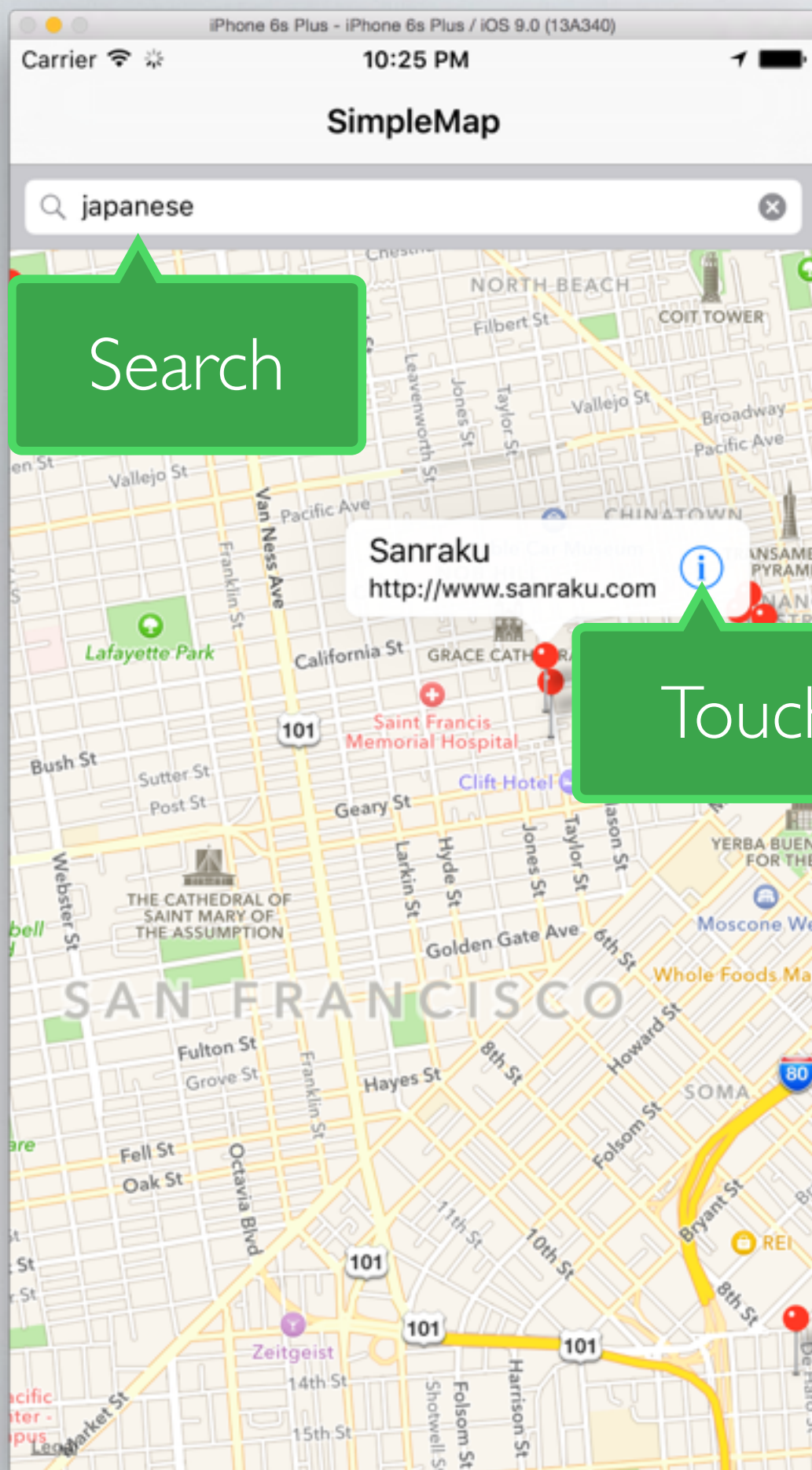




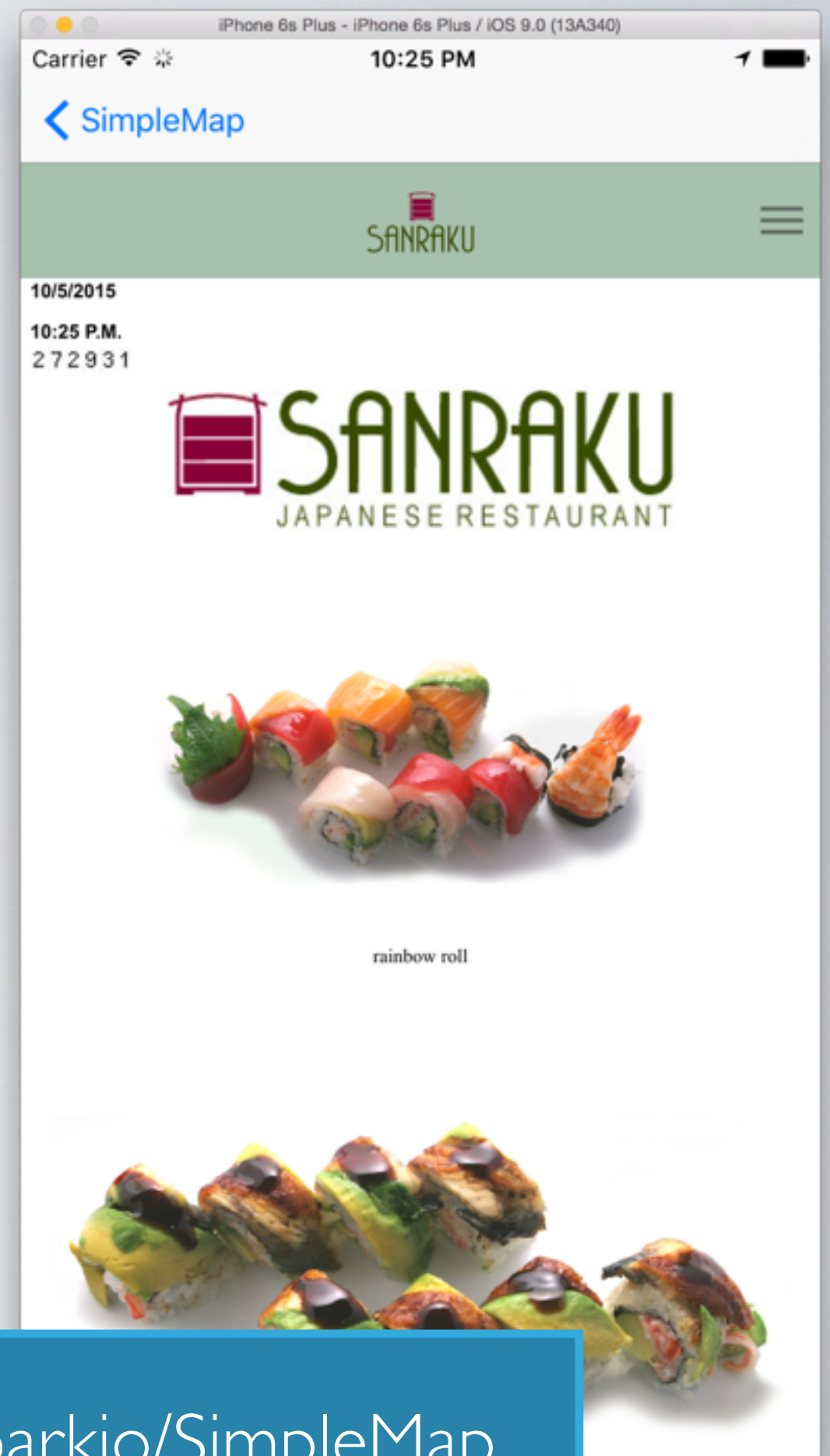
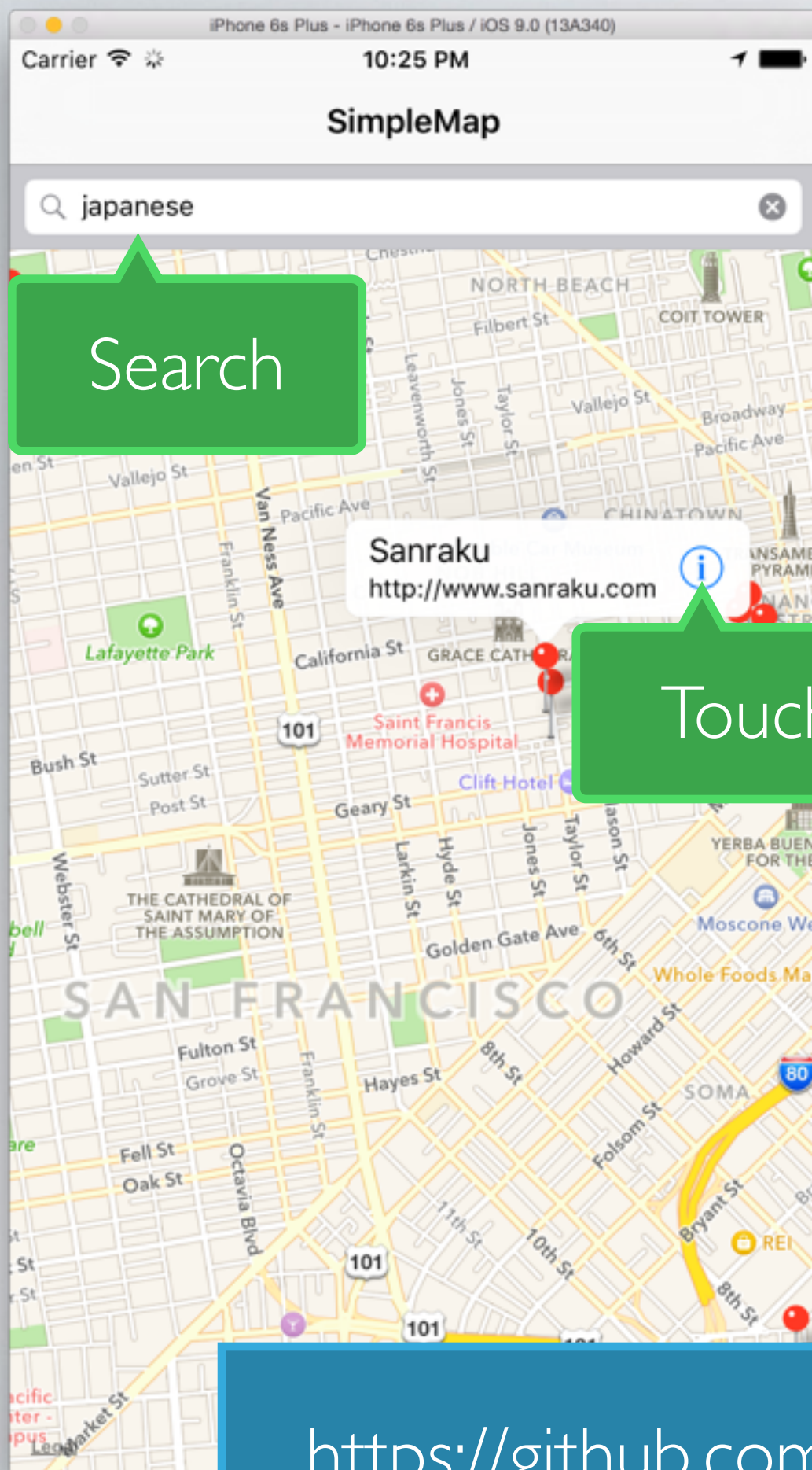




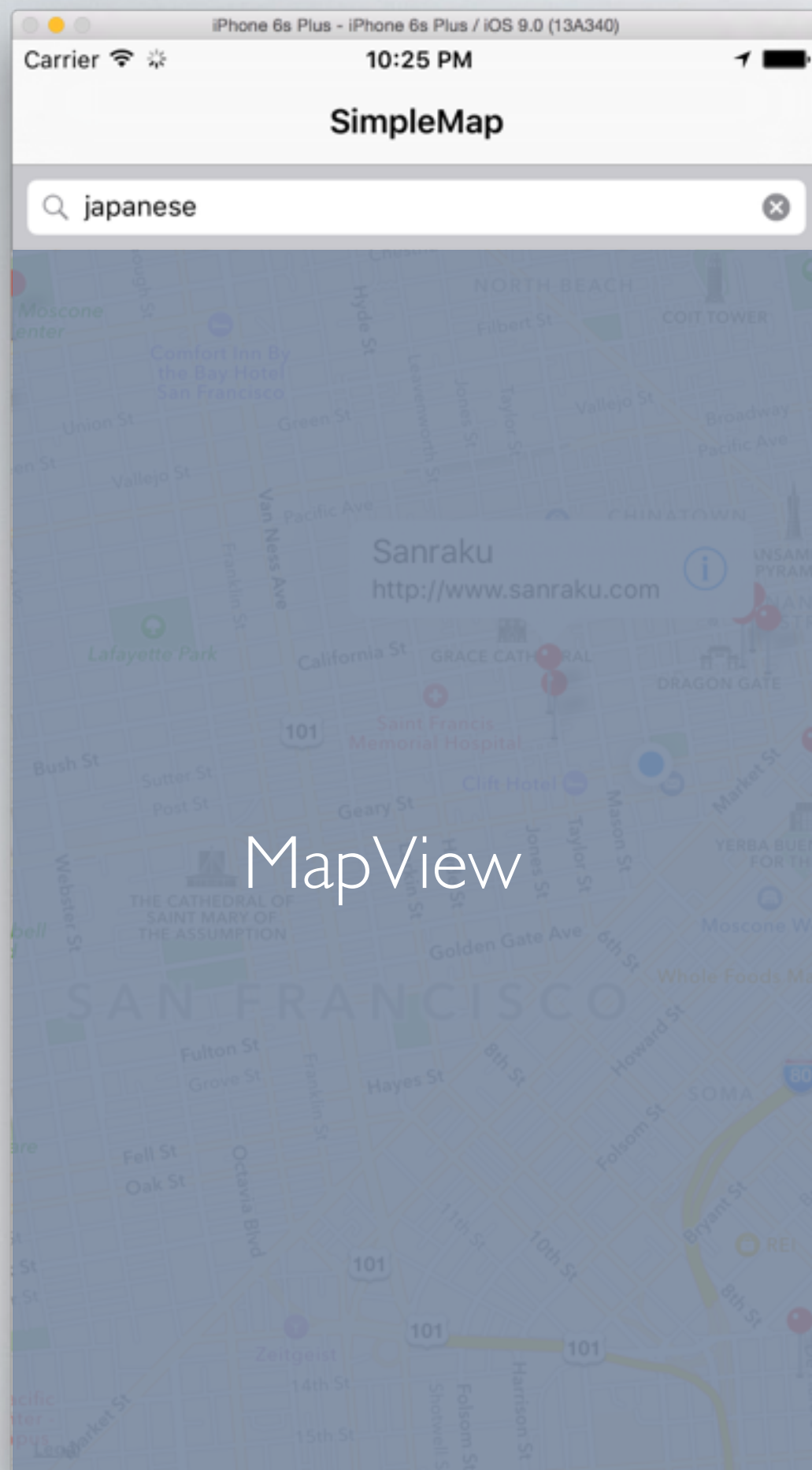






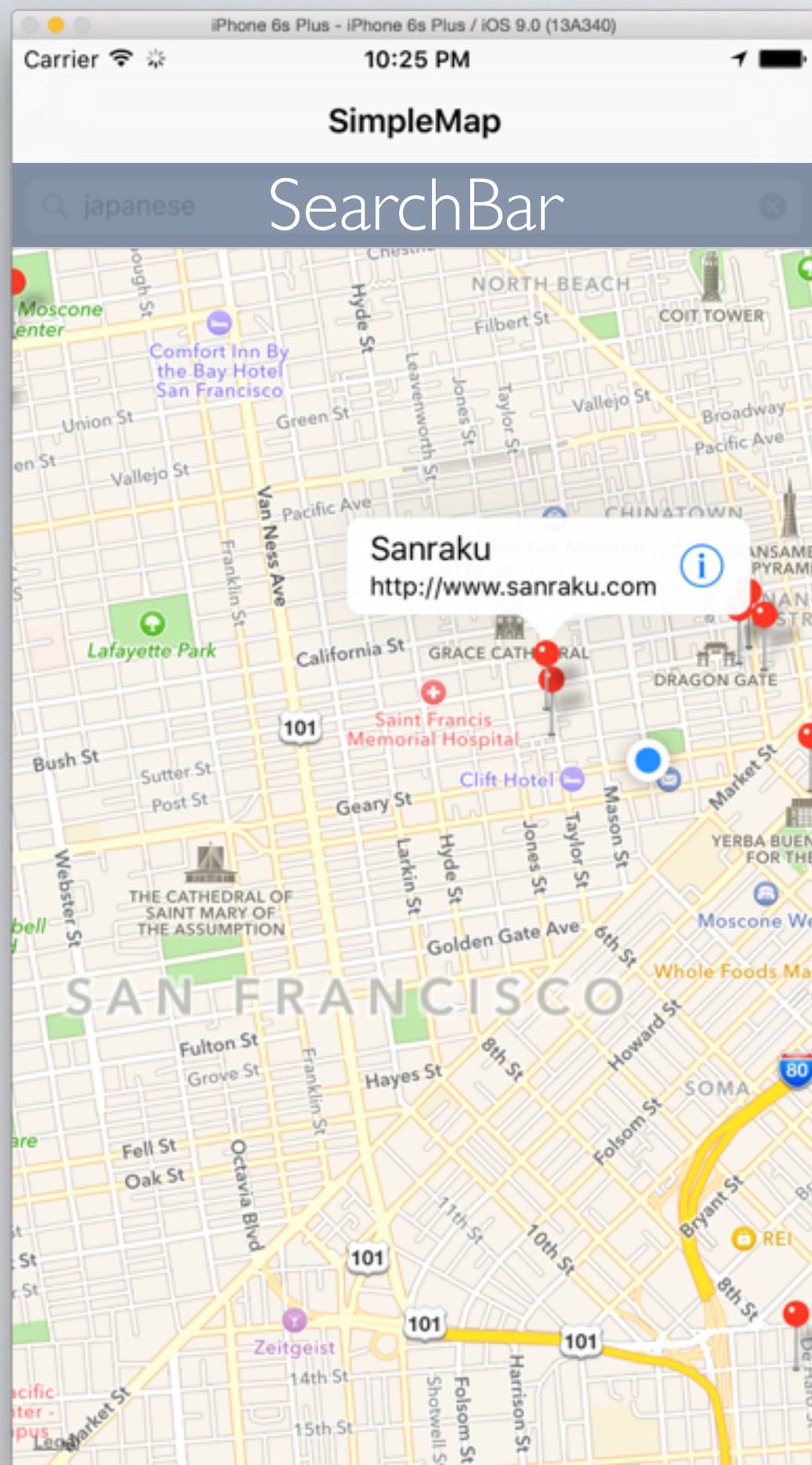


<https://github.com/talentsparkio/SimpleMap>

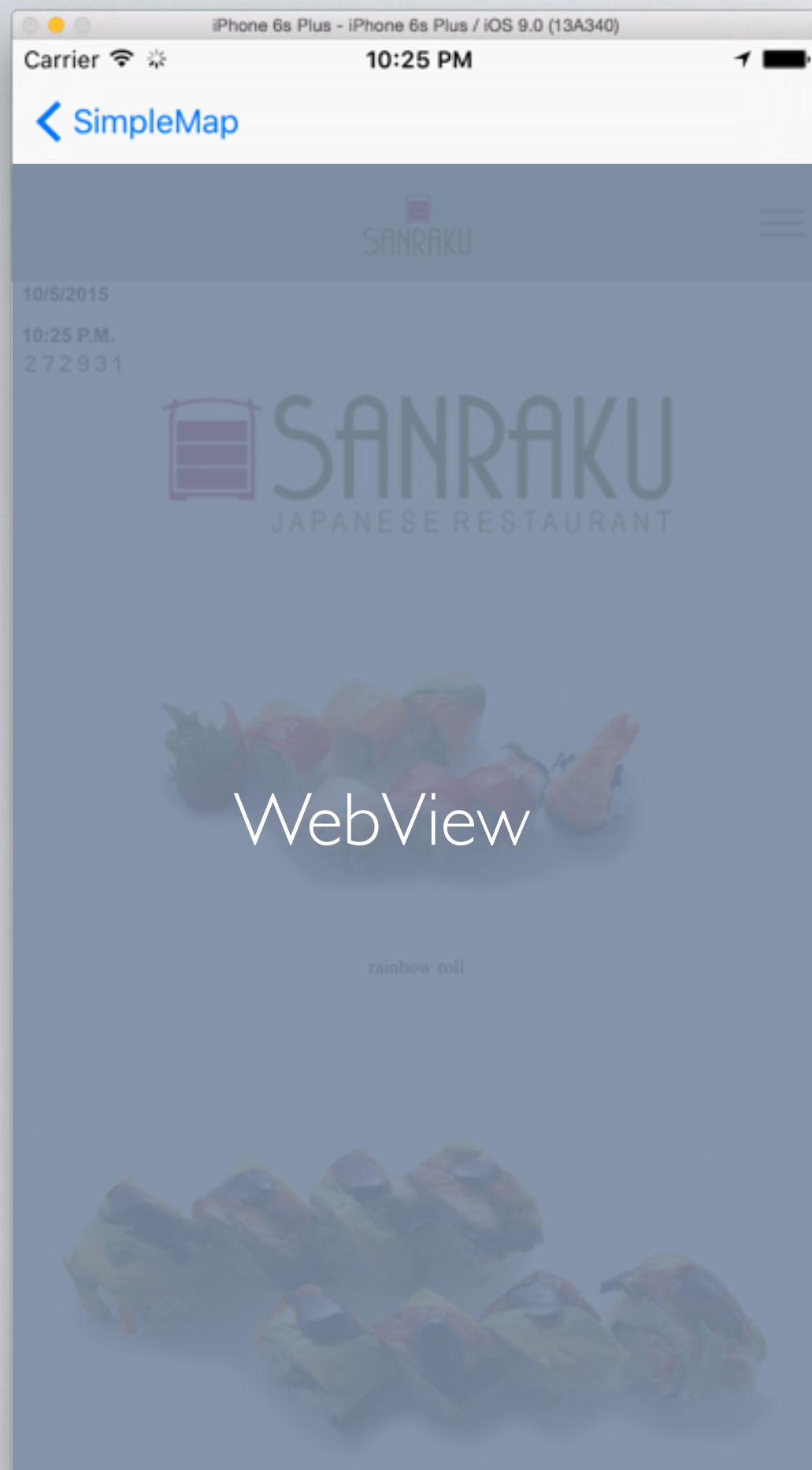


- Displays map {terrain, satellite, hybrid}
- Displays pins
- Displays routes





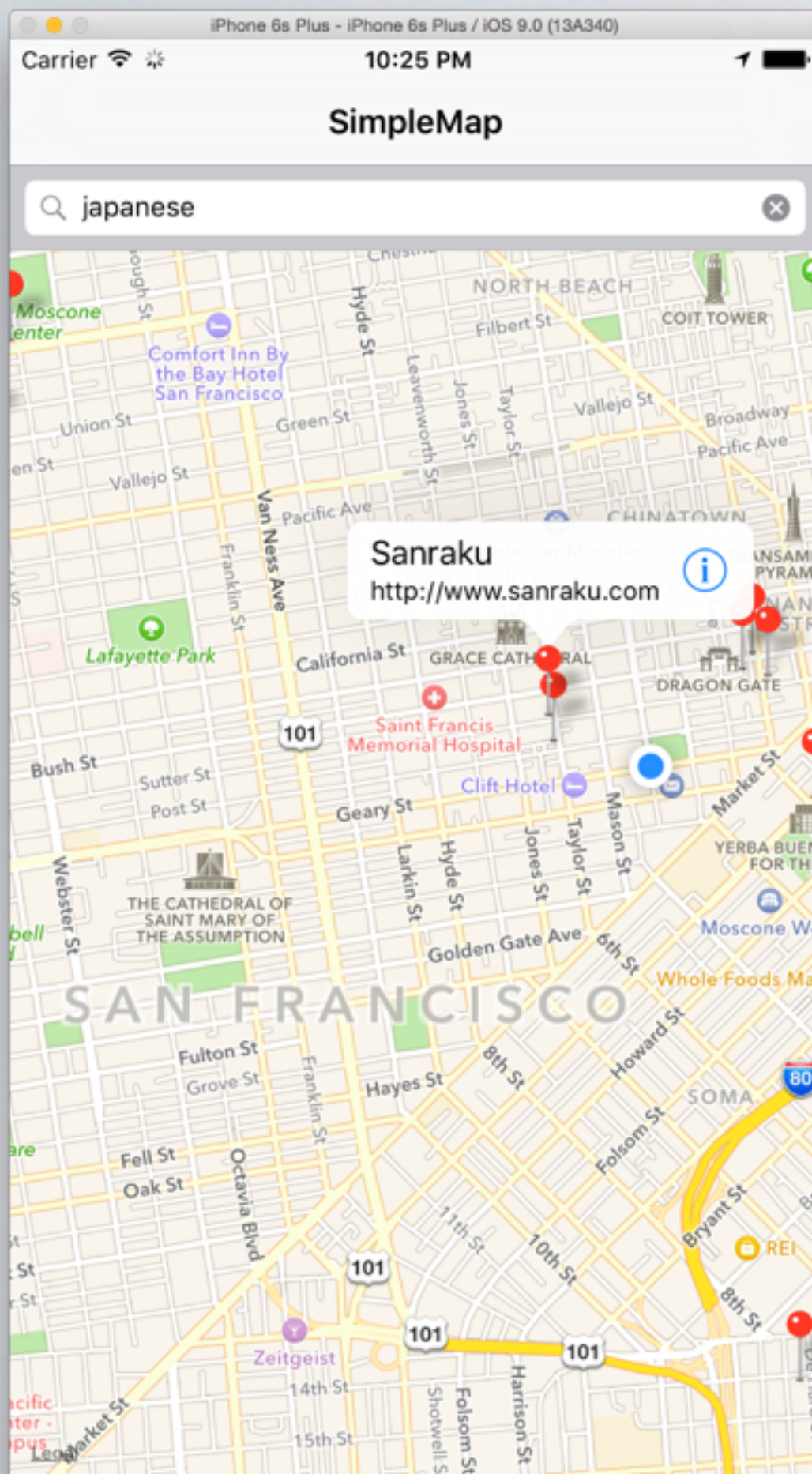
- General-purpose search bar
- Assign a delegate to see when text editing {starts, ends}
- Usually used with TableViews



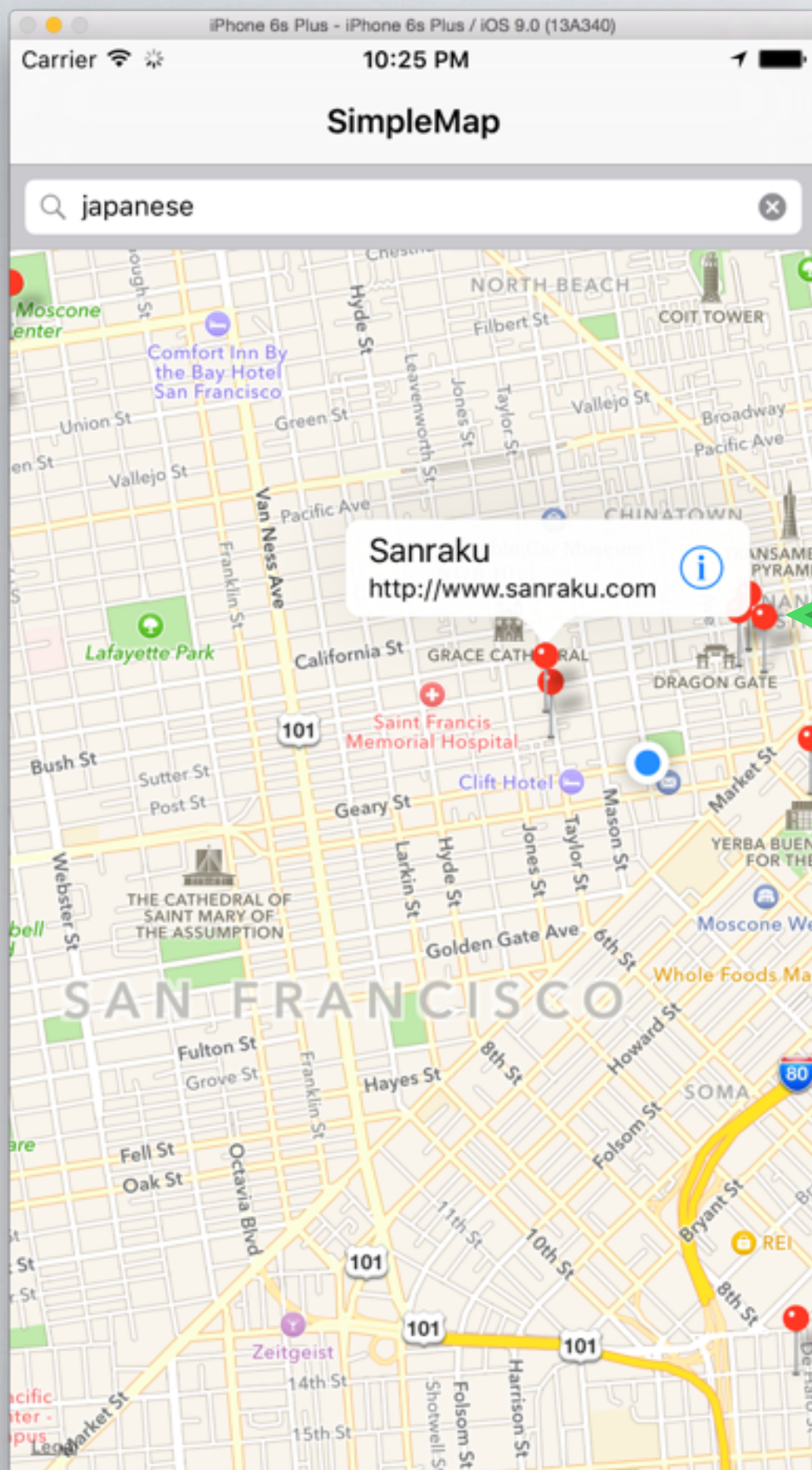
WebView

- Powered by WebKit
- Fully-capable browser
- Can be easily embedded
- Can customize actions



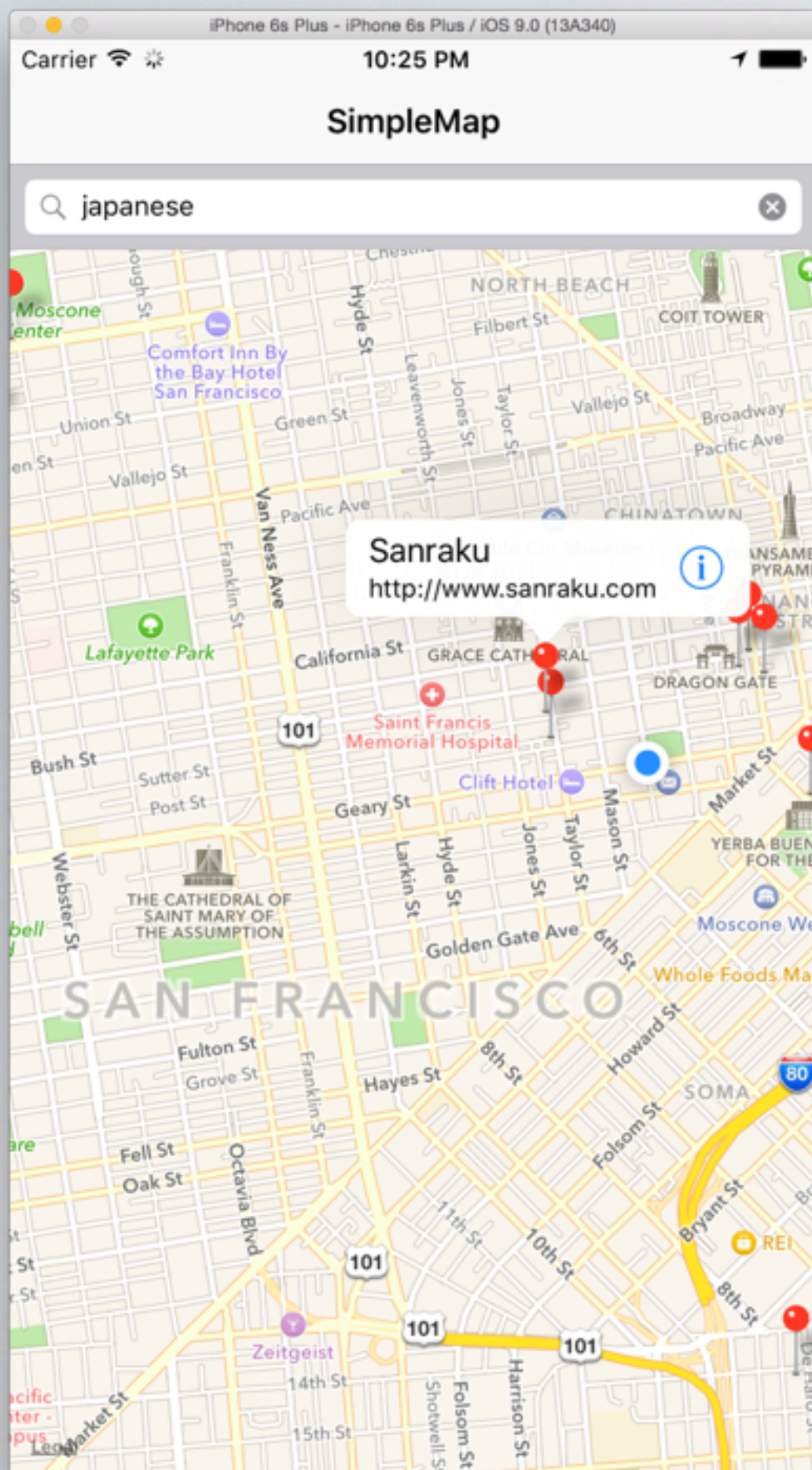




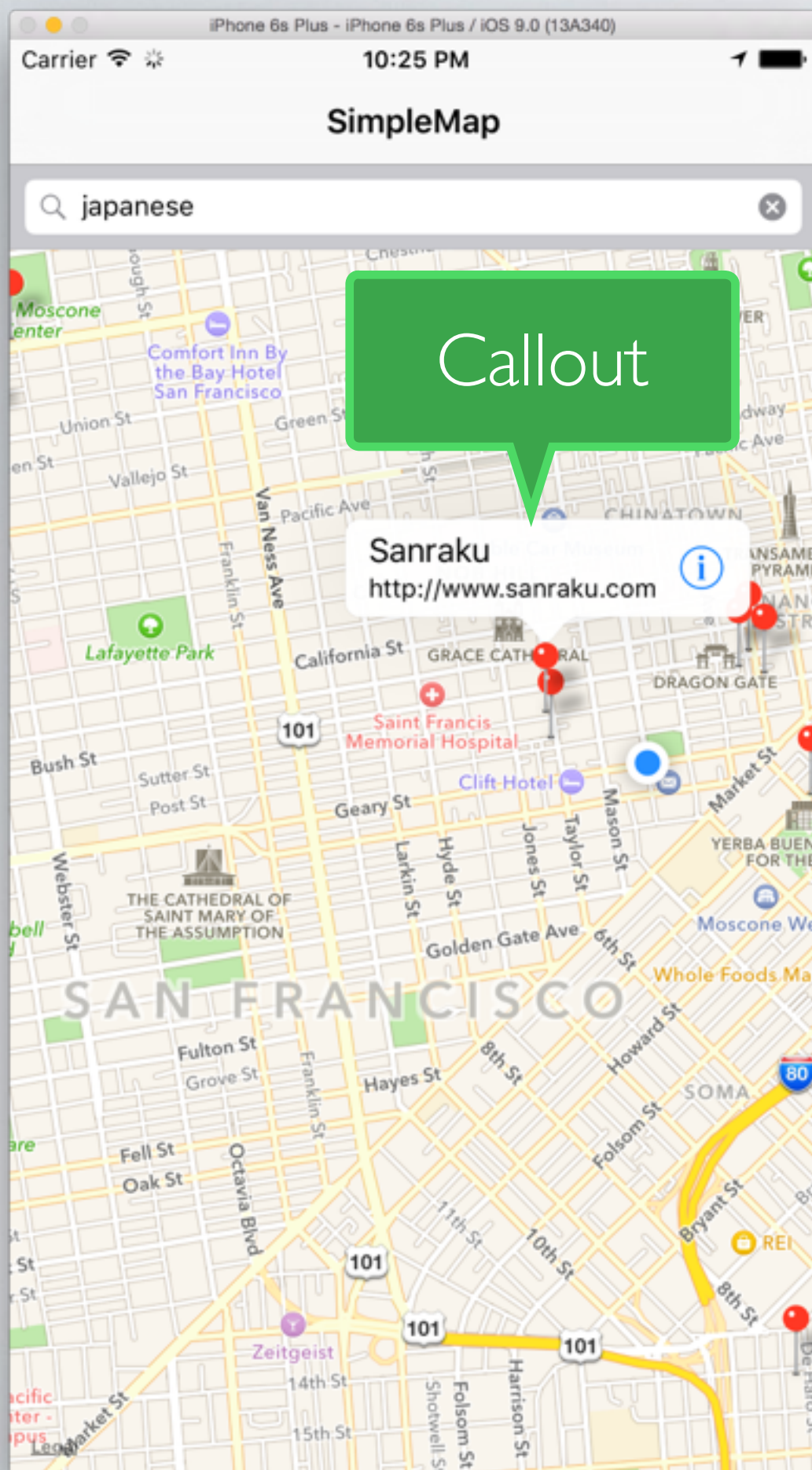


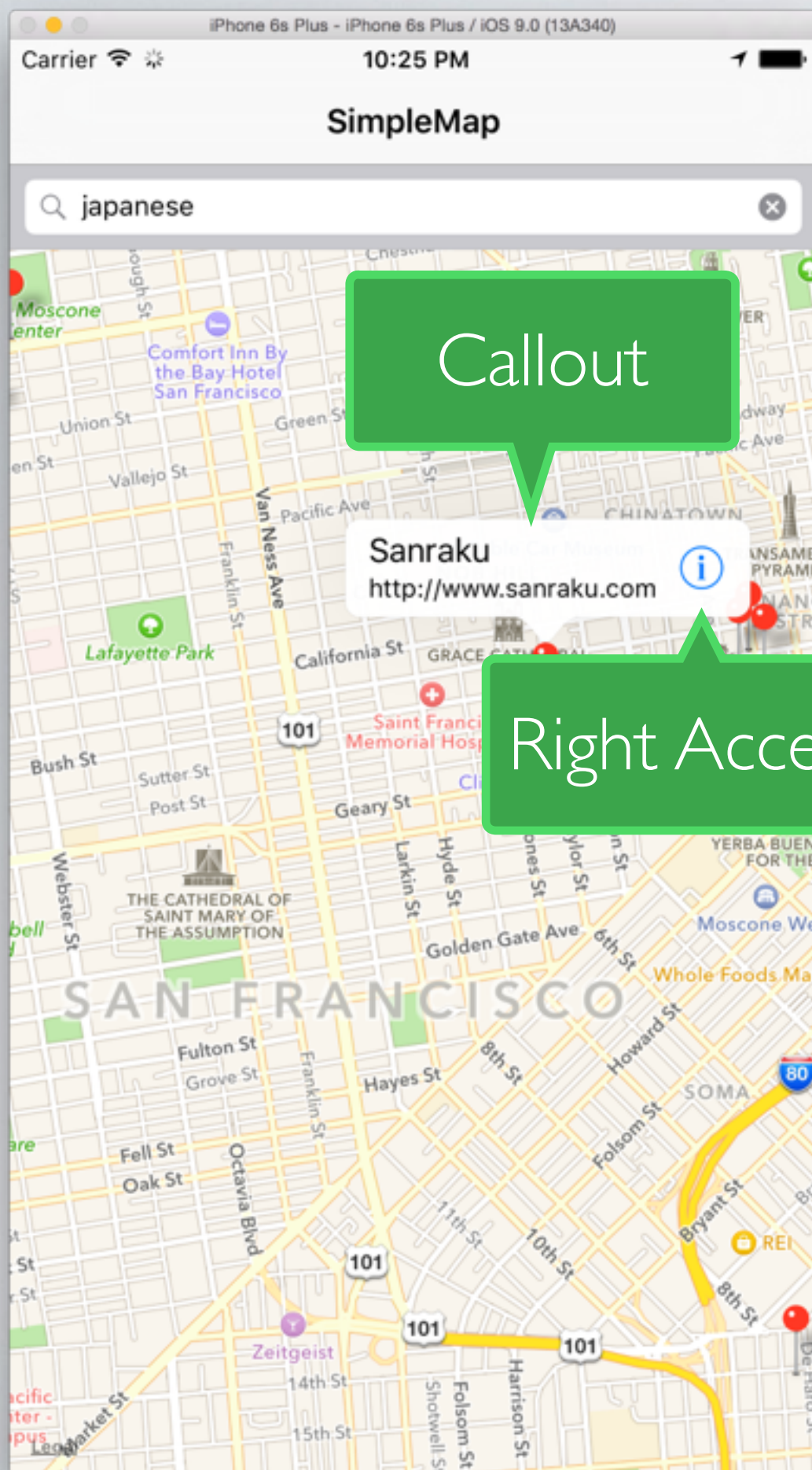
MKPinAnnotationView



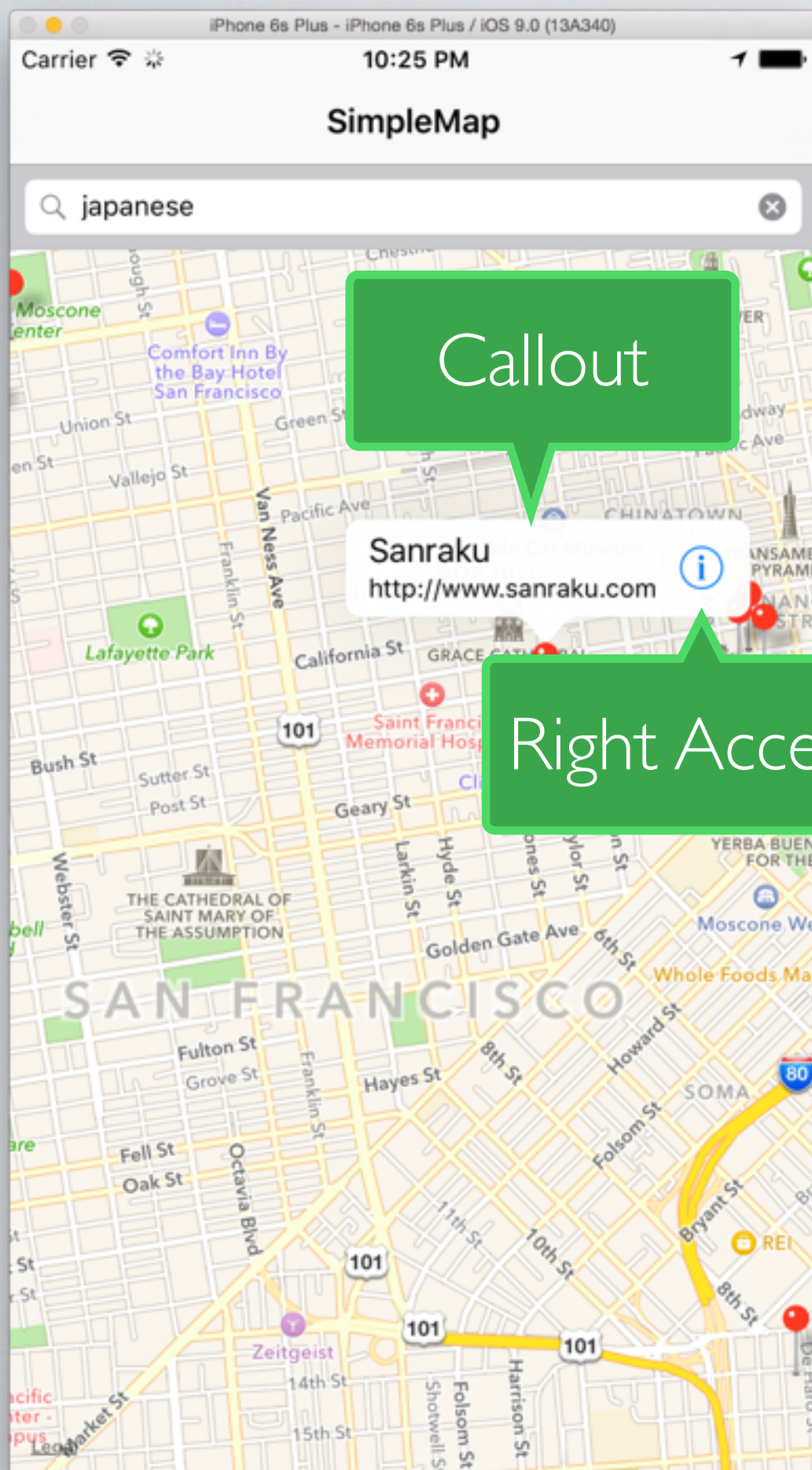






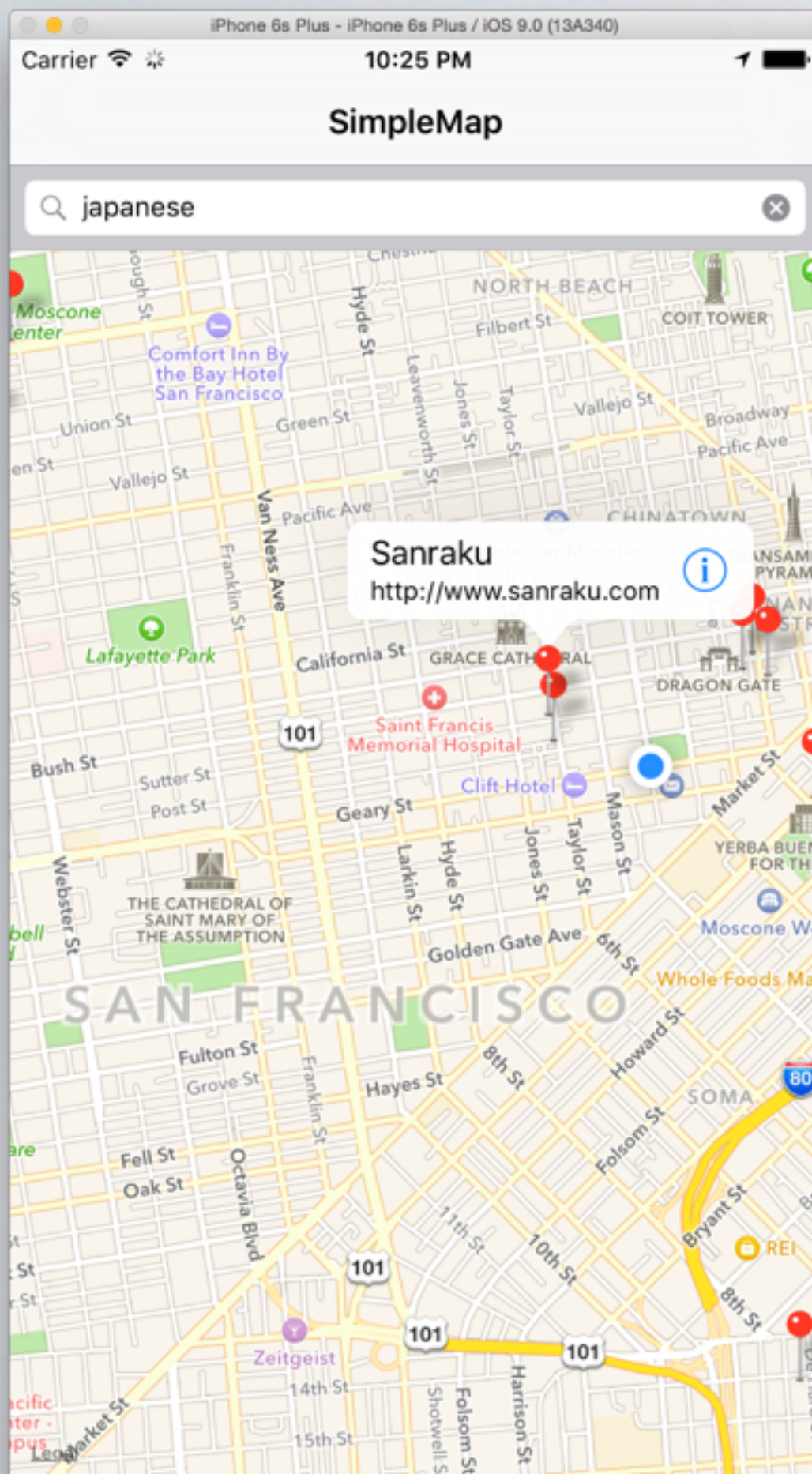




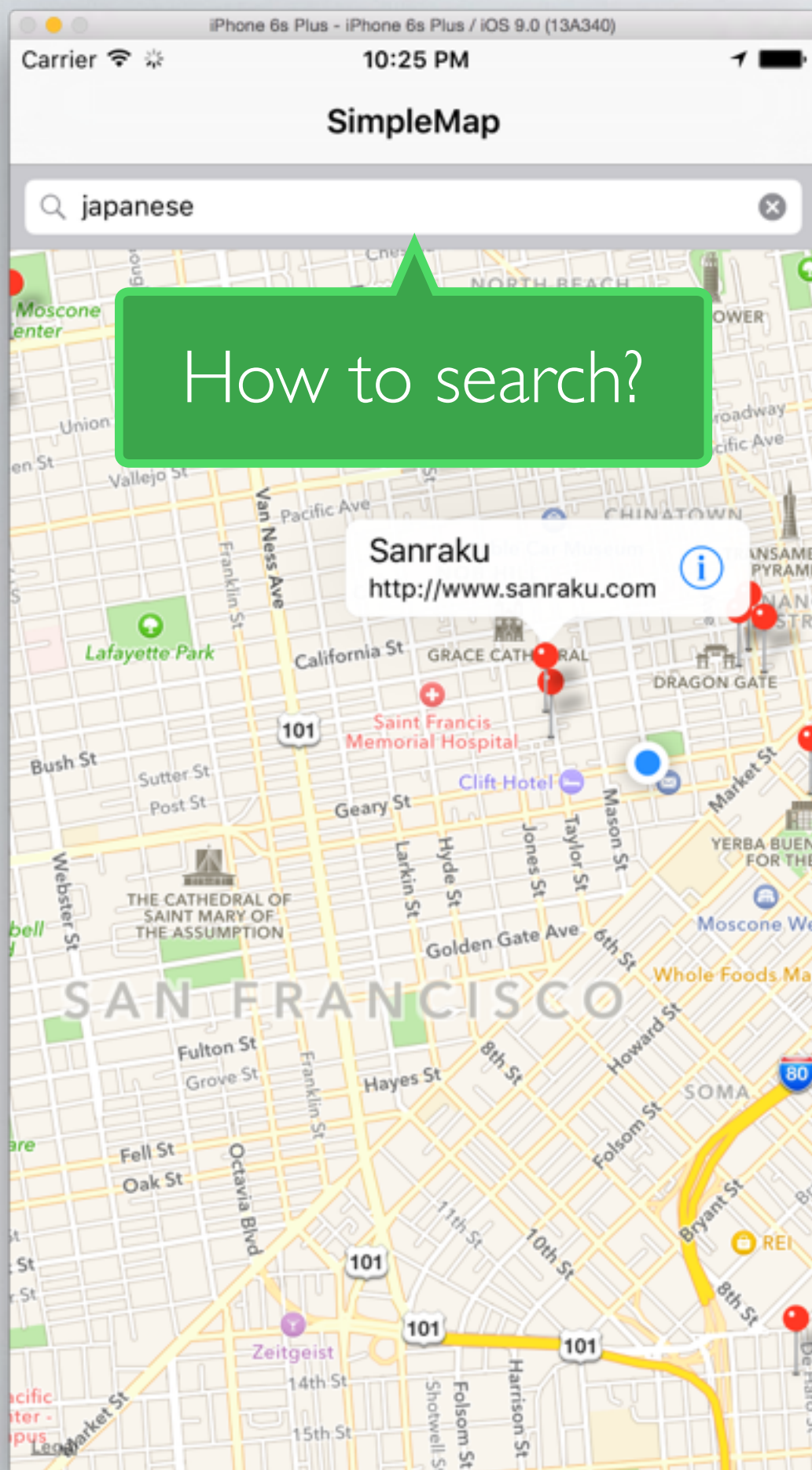


- annotation: MKAnnotation
- image: UIImage
- leftCalloutAccessoryView: UIView
- rightCalloutAccessoryView: UIView
- enabled: Bool
- centerOffset: CGPoint
- draggable: Bool

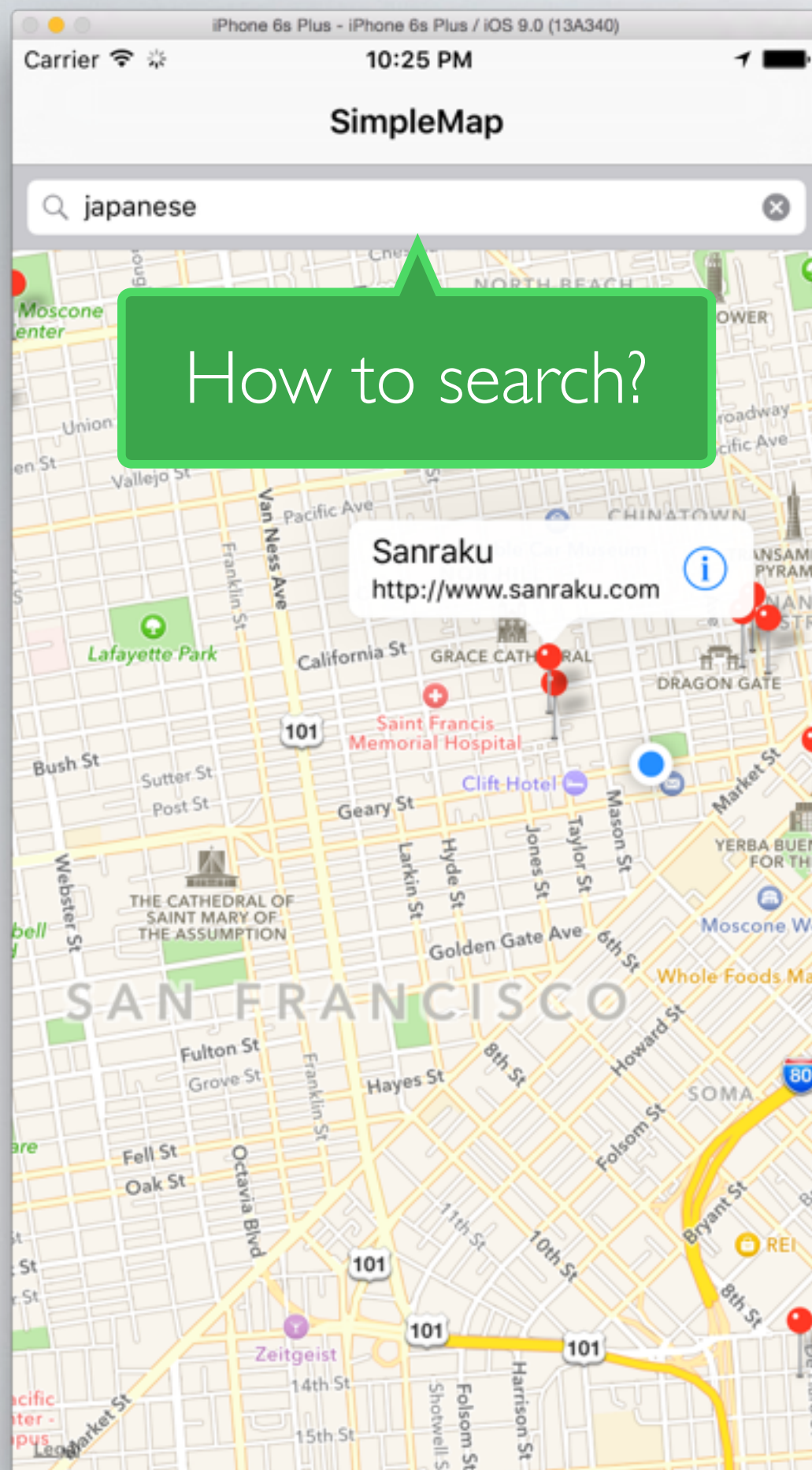












- MKLocalSearchRequest
  - naturalLanguageQuery
  - region
- MKLocalSearch
  - initWithRequest