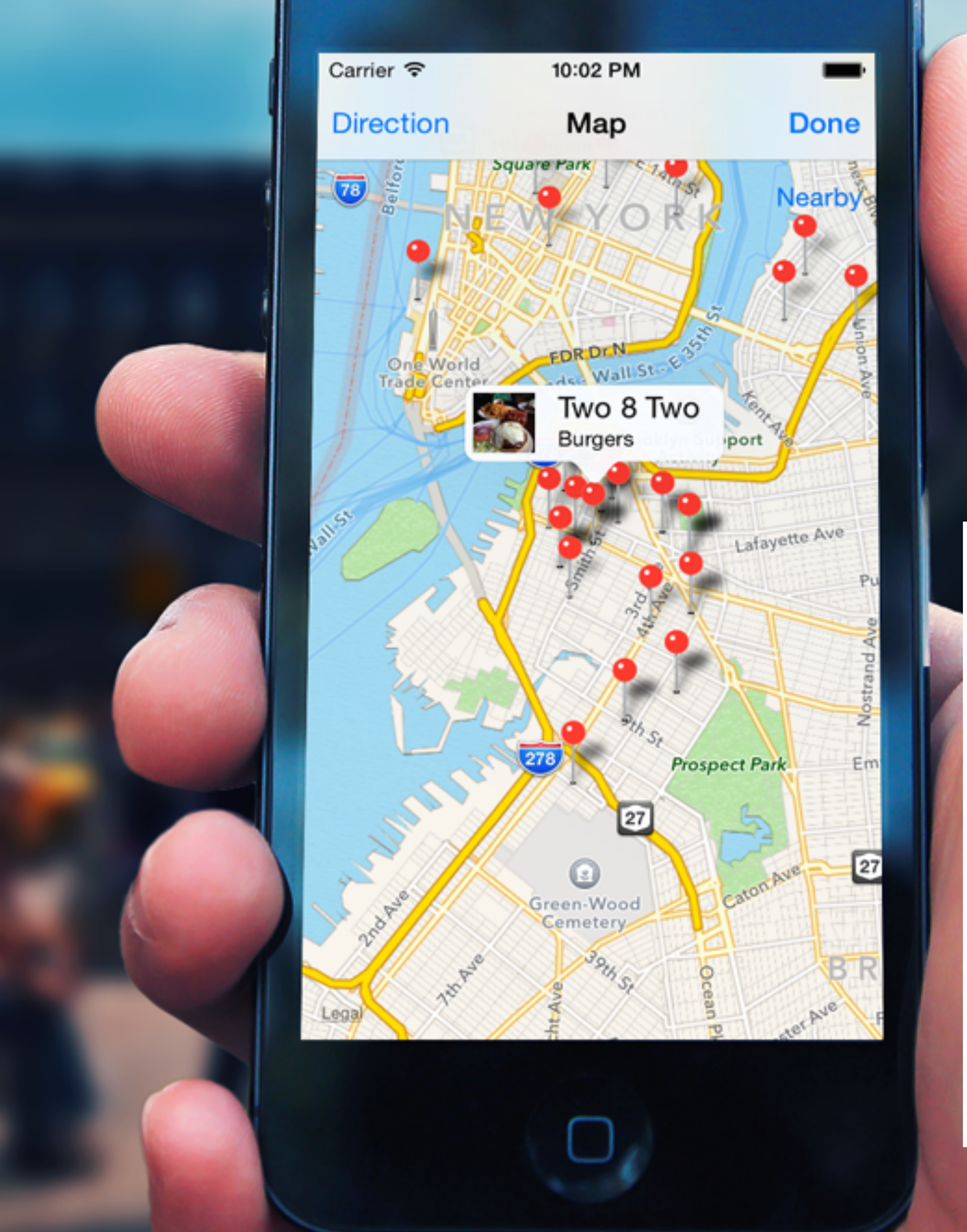


CHAPTER 9

SEARCH NEARBY POINTS OF INTEREST USING LOCAL SEARCH

Learn how to search nearby points of interest using the new Search API in the MapKit framework

LOCAL SEARCH API

Introduced in iOS 7, the new Search API (i.e MKLocalSearch) allows iOS developers to search points of interest and display them in maps. App developers can use this API to perform searches for locations, which can be name, address, or type, such as coffee or pizza.

The use of *MKLocalSearch* is very similar to the MKDirections API covered in the previous chapter. You'll first need to create an MKLocalSearchRequest object that will bundle your search query. You can also specify the map region to narrow down the search result. You then use the configured object to initialize an MKLocalSearch object and perform the search.

The search is performed remotely in an asynchronous way. Once Apple returns the search result (as an MKLocalSearchResponse object) to your app, the complete handler will be executed. In general, you'll need to parse the response object and display the search results in the map.

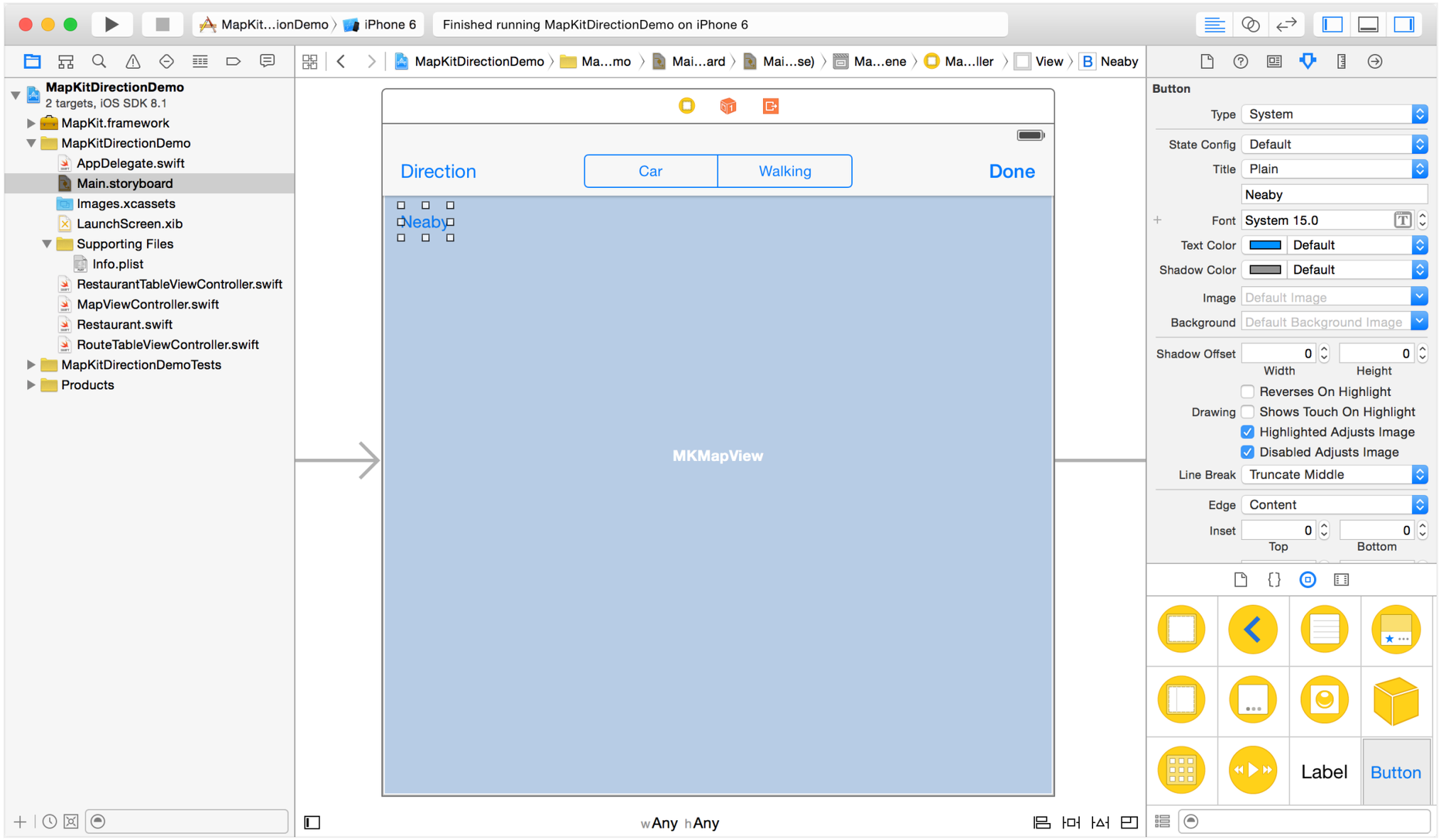
LOCAL SEARCH DEMO APP

There is no better way to understand local search than working on a demo project. Again we'll build on top of the previous project and add a Nearby feature. When you tap the Nearby button, the app searches for nearby restaurants and pins the places on the map.

To start with, first download the Xcode project template from https://www.dropbox.com/s/kzkyokepakkzx0i/MapKitDirectionDemo.zip?dl=0.

ADDING A NEARBY BUTTON IN STORYBOARD

Okay, let's get started. First go to Main.storyboard and add a button item to the map view. Name the button Nearby.



SEARCH NEARBY RESTAURANTS AND ADDING ANNOTATIONS

In the MapViewController class, we'll create an action method called *showNearby* for the nearby button. Insert the following code snippet in the class:

```
@IBAction func showNearby(sender: AnyObject) {
    let searchRequest = MKLocalSearchRequest()
    searchRequest.naturalLanguageQuery = restaurant.category
    searchRequest.region = mapView.region

    let localSearch = MKLocalSearch(request: searchRequest)
    localSearch.startWithCompletionHandler { (response, error) -> Void in

        if error != nil {
            println("Error: \(error.localizedDescription)")
            return
        }

        let mapItems = response.mapItems as! [MKMapItem]
        var nearbyAnnotations:[MKAnnotation] = []
        if mapItems.count > 0 {
            for item in mapItems {
                // Add annotation
                let annotation = MKPointAnnotation()
                annotation.title = item.name
                annotation.subtitle = item.phoneNumber
                annotation.coordinate = item.placemark.location.coordinate
                nearbyAnnotations.append(annotation)
            }
        }
        self.mapView.showAnnotations(nearbyAnnotations, animated: true)
    }
}
```

To perform a local search, here are the two things you need to do:

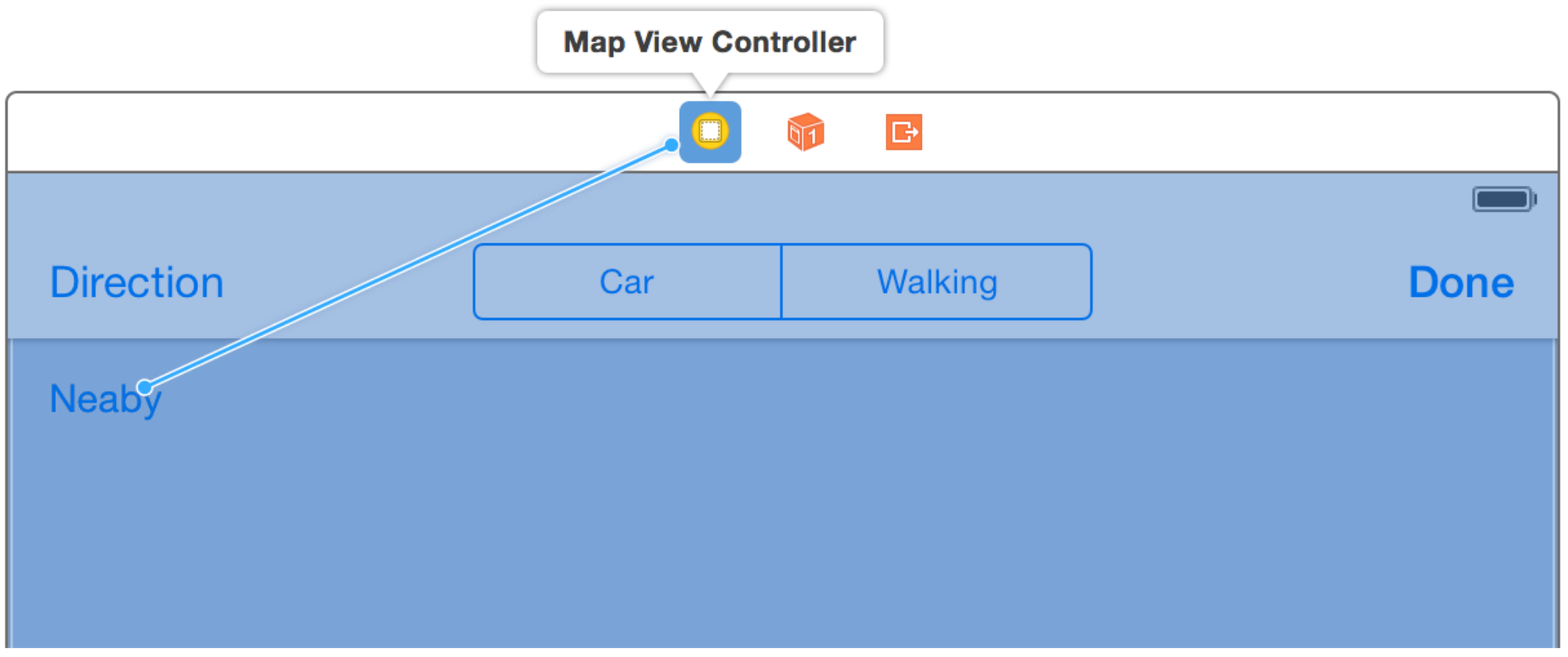
- Specify your search parameters in an MKLocalSearchRequest object. You are allowed to specify the search criteria in natural language by using the *naturalLanguageQuery* parameter. For example, if you want to search for a nearby “cafe”, you can specify “cafe” in the search parameter. Since we want to search for similar types of restaurants, we specify the *restaurant.category* in the query.

- Initiate the local search by creating an MKLocalSearch object with the search parameters. An MKLocalSearch object is used to initiate a map-based search operation and delivers the results back to your app asynchronously.

In the *showNearby* method, we lookup the nearby restaurants that are of the same type as the one previously selected. Furthermore, we specify the current region of the map view as the search region.

We then initialize the search by creating the MKLocalSearch object and invoking the *startWithCompletionHandler:* method. When the search completes, the closure will be called and the results are delivered as an array of MKMapItem. In the body of the closure, we loop through the items (i.e. nearby restaurants) and highlight them on the map using annotations.

Okay, you’re almost ready to test the app. Just go to the storyboard and connect the “Nearby” button with the showNearby method.



TESTING THE DEMO APP

Now hit the Run button to compile and run your app. Select a restaurant to bring up the map view. Tap the Nearby button and the app should show you the nearby restaurants.

Cool, right? With just a few lines of code, you took your Map app to the next level. If you're going to embed a map within your app, try to explore the local search API.

For your reference, you can download the complete Xcode project from https://www.dropbox.com/s/la3872e8jffou9q/MapKitLocalSearch.zip?dl=0.