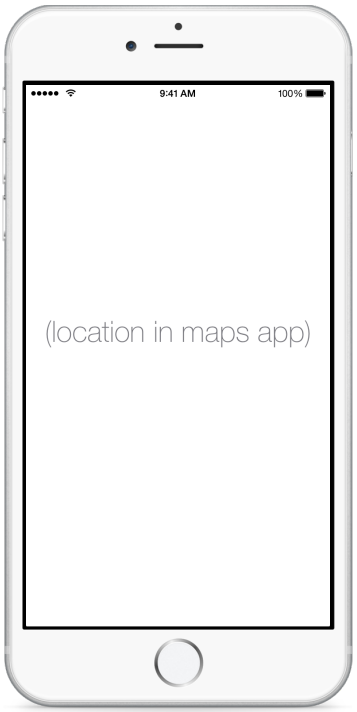
Found
Lesson 1

Description

Add a button to the interface that displays a location in the Maps app.



Learning Outcomes

- Practice creating interface elements and controller actions with Interface Builder.
- Define what a URL is, and discover how iOS behaves with certain URLs.
- Discover how to bring the Maps app to the foreground from another iOS app.
- Relate the running application to the `UIApplication` instance returned by `sharedApplication`.

Vocabulary

URL	`NSURL`	`UIApplication`

Materials

- **Found Lesson 1** Xcode project

Opening

How can we open a map location from our app?

Agenda

- Using Interface Builder and the Object Library (⌥⌘L), add a Button to the view labeled **Open Maps App with URL**.
- Add constraints to center the button horizontally, and to set its top vertical space relative to the main view.
- Using Interface Builder and the Assistant Editor (⌥⌘↩), create a connection from the button to a controller action called `openMapsAppWithURL:`.

```
@IBAction func openMapsAppWithURL(sender: UIButton) {
   if let url =
      NSURL(string: "http://maps.apple.com/?q=Yosemite") {
      let app = UIApplication.sharedApplication()
      app.openURL(url)
   }
}
```

- Explain the creation of an `NSURL` from a string, representing a URL or "the location of a resource", a map that one wishes to open, which iOS handles by opening the Maps app.
- Explain how the `q=` query string parameter represents a "query," and is parsed by the Maps app to search for a location on the map.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIApplication` class reference and the `openURL:` method.
- Explain how the call to `UIApplication sharedApplication` returns a reference to the app instance itself.
- Run the app (⌘R), tap the button, and observe how the Maps app enters the foreground.

Closing

What if the URL string is `http://maps.apple.com/?q=37.331686,-122.030656`? What do the numbers mean?

Modifications and Extensions

- Investigate the Apple URL Scheme Reference for map links. Experiment with Apple Maps parameters such as `near=`, to see what the Maps app displays.

Resources

Xcode Overview: Connect User Interface Objects to Code https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/edit_user_interface.html#//apple_ref/doc/uid/TP40010215-CH6-SW3

Cocoa Application Competencies for iOS: Target-Action https://developer.apple.com/library/ios/documentation/General/Conceptual/Devpedia-CocoaApp/TargetAction.html

Creating an Action Connection https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingAction.html

Apple URL Scheme Reference http://developer.apple.com/library/ios/featuredarticles/iPhoneURLScheme_Reference/Introduction/Introduction.html

The Swift Programming Language: Optional Binding https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html#//apple_ref/doc/uid/TP40014097-CH5-ID333

NSURL Class Reference http://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSURL_Class/Reference/Reference.html

UIApplication Class Reference http://developer.apple.com/library/ios/documentation/UIKit/Reference/UIApplication_Class/