

Mobile Apps Development for Journalists

Week 10, Thursday - Fall 2016
University of Texas at Austin
School of Journalism

Jeff Linwood
jlinwood@gmail.com
@jefflinwood

Agenda

- Location and Mapping Review
- Managers and Delegates
- Coding Exercise: Real-time Location Update App

Location and Mapping Review

- Key frameworks: CoreLocation and MapKit
- Need to put privacy setting in Info.plist
- Your app can respond to location changes by the user
- Your app can also display a map, and update it with pins

Location and Privacy

- Your iOS App will have to ask the user for permission to get their location
- Two choices for tracking user's location
 - Always - the application will get location updates in the background
 - When in use - only when the application is running

Provide Privacy in Info.plist

- Add a row, with a key value of either:
- `NSLocationWhenInUseUsageDescription`
- `NSLocationAlwaysUsageDescription`
- And a String value with a message such as "This app would like to use your location for...."

Getting Started with Location

- import CoreLocation framework
- Initialize a Location Manager
- Set the delegate on the location manager to self
- Request authorization from the user (the popup dialog)
- Tell the Location Manager to Start Updating Locations

Managers and Delegates

- We haven't worked with managers or delegates before
- CLLocationManager - responsible for getting location updates from the phone and passing them to the delegate
- CLLocationManagerDelegate - your code that does something with updates
- Need to set the delegate on the manager to self

Implementing a Delegate

```
import UIKit
import CoreLocation

class ViewController: UIViewController, CLLocationManagerDelegate {

    let locationManager = CLLocationManager()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        locationManager.delegate = self
        if (CLLocationManager.authorizationStatus() == .NotDetermined) {
            locationManager.requestWhenInUseAuthorization()
        }
        if (CLLocationManager.authorizationStatus() == .AuthorizedWhenInUse) {
            locationManager.startUpdatingLocation()
        }
    }

    func locationManager(manager: CLLocationManager, didUpdateLocations
locations: [CLLocation]) {
        print("Received Location Update")
    }
}
```


Apple Documentation

- CLLocationManager - <https://developer.apple.com/reference/corelocation/cllocationmanager>
- CLLocationManagerDelegate - [https://
developer.apple.com/reference/corelocation/
cllocationmanagerdelegate](https://developer.apple.com/reference/corelocation/cllocationmanagerdelegate)

In Class Exercise

- See LocationApp.zip in Canvas
- Files/Projects/Week 10

Testing the App

- In Simulator, go to the Debug Menu, and choose the Location sub menu, then pick the run, bike, or freeway drive.
- You can also set a custom location, or even use location track files