184 5th Avenue,
New York,NY 10010
www.turntotech.io
nyc@turntotech.io

TURNTOTECH

Week 2 The Geo-location app

Goal: Allow a parent to set up a geo-fencing monitor for their child. The parent sets a point and a radius within which the child is supposed to be. If the child moves out of that, the parent is notified.

App user interface: There are two apps, parent and child.

The parent app does two things

1. Create parent userid
2. Set preferences for the parent
3. Get status

The child app does only one thing – once activated with the parent's userid, it constantly sends its location to a server.

Web Version of the above app

Visit http://protected-wildwood-8664.herokuapp.com/ and explore the various options. It is a fully working web version of both the parent and child apps. Our goal is to create iOS apps that communicate with the backend using JSON.

URI's for JSON Services required to build the two apps explained above

1. Create parent userid
    a. Assuming you have three properties userID, latitude and longitue
    b. Create JSON string from this Objective-C dictionary using NSDictionary to JSON conversion API.
        i.
        ```
        NSDictionary *userDetails = @{@"utf8": @"✓", @"authenticity_token":@"EvZva3cKnzo3Y0G5R3NktucCr99o/2UWOPVAmJYdBOc=", @"user":@{@"username":self.userID,@"latitude":self.latitude,@"longitude":self.longitude,@"radius":self.radius}, @"commit":@"Create User", @"action":@"update", @"controller":@"users"};
        ```
        ii. Convert to JSON String using NSDictionary to JSON conversion API
    c. HTTP POST above JSON to /users (/users is short for http://protected-wildwood-8664.herokuapp.com/users)
2. Update child's status
    a. Create JSON from this dictionary
        i.
        ```
        NSDictionary *childDict = @{@"utf8": @"□", @"authenticity_token":@"EvZva3cKnzo3Y0G5R3NktucCr99o/2UWOPVAmJYdBOc=", @"user":@{@"username":self.userID,@"current_lat":self.latitude,@"current_longitude":self.longitude}, @"commit":@"Create User", @"action":@"update", @"controller":@"users"};
        ```
    b. HTTP PATCH request – send above JSON to /users/username
3. Get status (from Parent app)
    a. Send HTTP GET request to /users/username.json
        i. Convert JSON to NSDictionary and extract value for "**Is in zone**"

Technologies we'll explore

1. HTTP, JSON and some debugging tools
2. HTTP calls using Objective-c
3. Using the device GPS
4. Synchronous vs. asynchronous calls. Pros and cons.
5. Underlying design patterns for asynchronous
6. XCode interface builder
7. Lifecycle of a simple objective-c app
8. User interface development for iOS
9. JSON-Obj-C conversions

Resources

HTTP related tools

http://curl.haxx.se/

Note: curl is typically pre-installed on a mac

Geolocation

Quick start

http://stackoverflow.com/questions/6894624/how-can-i-get-gps-location-in-iphone

Official doc

https://developer.apple.com/library/mac/documentation/CoreLocation/Reference/CoreLocation_Framework/CoreLocation_Framework.pdf

Networking

Quick start

http://codewithchris.com/tutorial-how-to-use-ios-nsurlconnection-by-example/

Official docs

https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/URLLoadingSystem/URLLoadingSystem.pdf

General App development

https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/RoadMapiOS.pdf

Deliverables

1. Build a simple app with a UI in XCode. You should be able to enter some information in a text field in the app running in the simulator and print that value using NSLog
2. Learn about **properties** in Objective-C. Write a paragraph explaining what a property is.
3. Build a simple app that constantly prints lat,long. Use the iOS simulator's location simulation options to simulate various locations
4. Build a simple app that asynchronously downloads the Google logo i.e. does a GET request

5. Build a simple app that asynchronously does an HTTP post. (Use the above information to create a user)
6. Build a simple app that converts an NSDictionary to JSON and JSON to NSDictionary
7. Build the parent app
8. Build the child app
9. Test the parent-child apps in various combinations