Warmup Assignment: Setup (for mac)

Due Tuesday, October 2nd, at 11:59 PM.

Overview

This assignment is just an environment setup. Your task is to follow the steps in this assignment, and make sure that your environment works properly before the deadline. As simple as this assignment may seem, installing required software can be tricky sometimes, and you certainly could face roadblocks. Please try to start the assignment early, and if you have any issues, drop by the office hours on Monday. Of course, feel free to ask friends and peers as much as you would like about the assignment. No honor code violation should be feared :)

Installation Instructions

(Please note that this is the **macOS** installation instructions. Other operating systems are in different documents)

There are 3 main parts to make sure that you can write and test React Native code:
Part 1: “Create React Native Apps”, which allows you to build and compile your code.
Part 2: A word editor to edit the code.
Part 3: A testing environment, which could be your phone or a simulator.

Part 1: “Create React Native Apps”

1. Install Homebrew:
    a. Open your terminal, and paste the following line there “`/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`". Let it finish the installation
2. Install node:
    a. If you already have node, open your terminal, and type in “`node -v`”. If your version is higher than 8.0, you can skip the “Install node” step. If your version is lower, you will need to uninstall node before moving on to the next step. To remove node:
        i. go to **/usr/local/lib** and delete any **node** and **node_modules**
        ii. go to **/usr/local/include** and delete any **node** and **node_modules** directory
        iii. if you installed with **brew install node**, then run **brew uninstall node** in your terminal
        iv. check your Home directory for any **local** or **lib** or **include** folders, and delete any **node** or **node_modules** from there
        v. go to **/usr/local/bin** and delete any **node** executable

   b. If you do not already have node, or just deleted your existing version, open your terminal, and execute the following commands
      ```
      brew update
      brew install node
      brew link node --force
      npm install -g npm
      ```
3. Run the following line in your command prompt “`npm install -g create-react-native-app`”. Wait for the command to finish running, and make sure you don’t get any errors.

Part 2: Download a word editor

This part is up to your own preference. If you already have a word editor you like and prefer, you can use it. Some of the suggested word editors are Sublime Text and Atom.

Part 3: Testing Environment

1. Setting up an iPhone simulator
   a. Install Xcode through the Apple App Store. It’ll take a while. Next, open up Xcode. In the top bar, got to XCode > preferences and click the Components tab. Install iOS 12.x simulator from the list.
   b. Once this is done, you should be able to open “simulator” as a separate app on your macOS without needing XCode to be running.
2. Creating a project and running it
   a. Run `npm install expo-cli --global` .This will allow you to use any phone as a testing platform for your apps!
   b. In you terminal, change the directory to somewhere where you want to create a project. Then, run `create-react-native-app AppName`. When prompted, choose blank project. Once it’s done, you will have your first React Native project!
   c. To launch the project, execute
      ```
      cd AppName
      expo start
      ```
      A browser window should open. This is your running app console.
   d. On your laptop, launch an iOS simulator (You can find it by searching “simulator” in spotlight).
   e. From the browser window that opened, click “Run on iOS simulator”.

Editing the app

Using your preferred text editor, open the App.js file in your project directory. This is your app code. You will see a Text tag saying “Open up App.js to start working on your app!”. Change it to whatever you want, and save the app. You will notice that the emulator immediately restarts.

Future projects and runs

If you think that the setup is complicated, you are not alone (It took me a long time to figure different bugs when I first installed it). What's nice is that future runs will not take anywhere nearly as much. All you need is:

1. Launch the simulator.
2. Run `expo start` in the app directory
3. Click "Run on iOS simulator".
4. Start coding!

Deliverables

Either show up at the office hours on Monday, or send us a screenshot of your phone with the text changed. If you face any roadblocks that you cannot solve by Monday office hours, make sure to stop by for help.

Notes

1. **Hot Reloading:** This is not a required feature, but we highly recommend it. In the emulator with your app open, click "`cmd+d`", and enable hot reloading, but disable live reloading. Hot reloading will reflect whatever changes you make, but it will keep the state of the app. Live reload will reload the whole app. Hot reload is helpful with faster refresh times and it's suitable for UI tweaks. You can also reload manually from the same menu "`cmd+d`".
2. **Testing on an actual device (iOS or Android):**
    a. On your phone, install the Expo client.
    b. Make sure your phone and laptop are on the same network.
    c. Once it is done, on your computer, start the project. Click on the "Send link with email/SMS". Enter your email/phone, and wait until you get the email/phone with the link to open. Once you open the link, it will open the already installed client app, and you should see a progress percentage on your laptop and phone. Loading the app will take a few minutes. Once it's done, you should see your app.
    d. You can access the settings menu mentioned above by shaking the device since there is no keyboard.
    e. The app on your phone will refresh immediately when you change the App.js file! You can do Hot reloading or live reloading. It's up to you.
3. **Testing on an Android emulator:**
    a. This will require installing Android Studio. If you are interested, check here under Android Emulator.