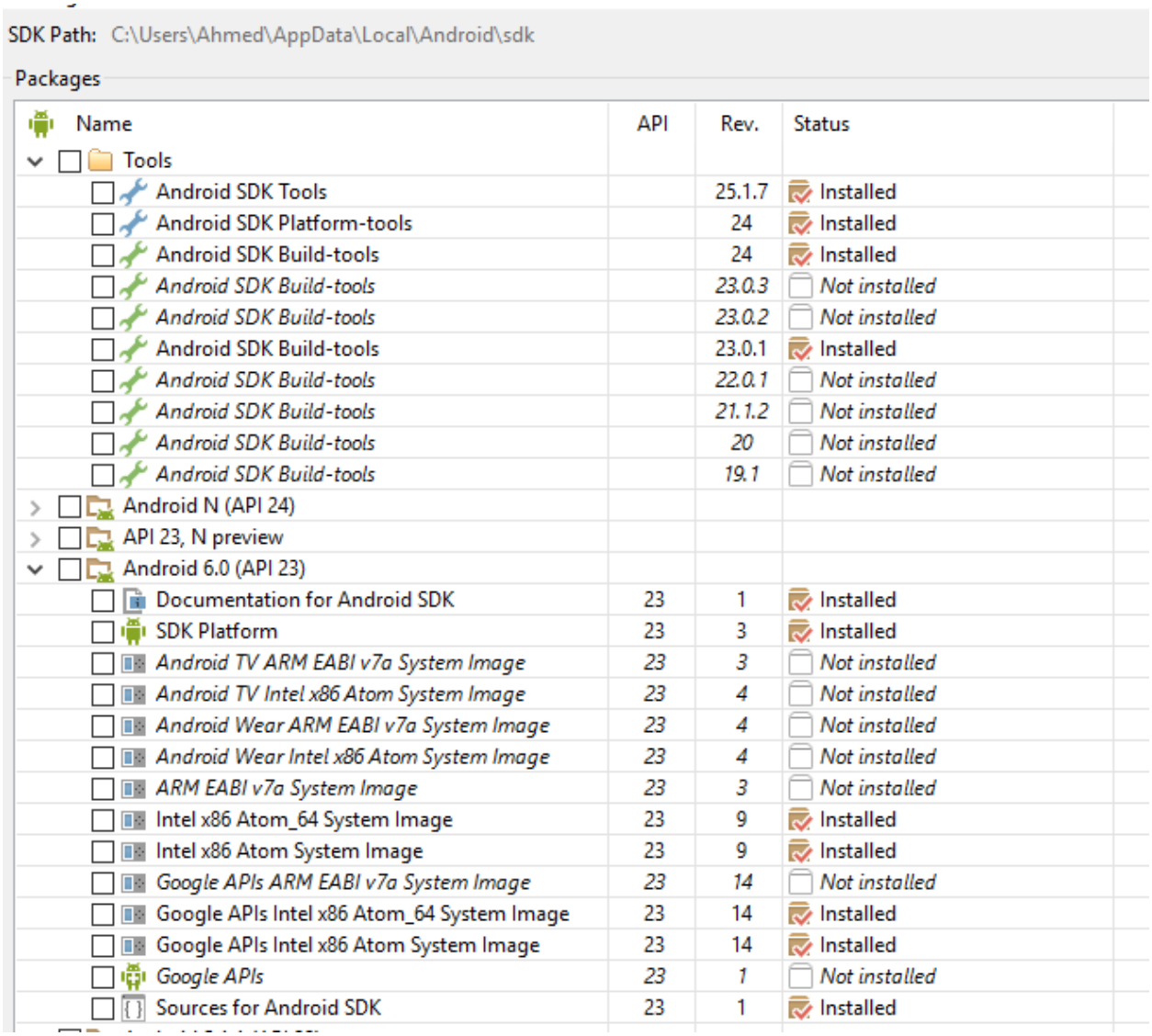
React Native Basic Guide (Windows and Android)

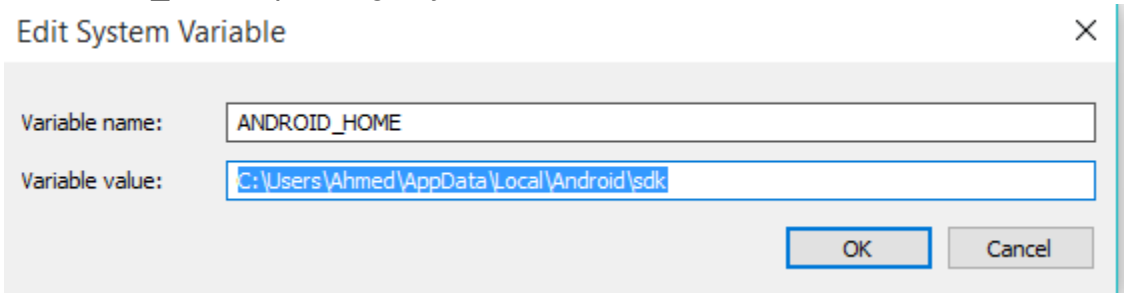
WARNING: This guide is intended to be a rough guide and *should* work for most machines. You may need to do additional troubleshooting to get your machine setup as issues arise.

1. At this point you should have each of the following downloaded and installed:
    a. React-native-cli (installed via npm install react-native-cli –g)
    b. Android Studio 2.0
    c. Java Development Kit
    d. Each of the following SDKs as shown in this picture:



Name	API	Rev.	Status
Tools			
Android SDK Tools		25.1.7	Installed
Android SDK Platform-tools		24	Installed
Android SDK Build-tools		24	Installed
Android SDK Build-tools		*23.0.3*	*Not installed*
Android SDK Build-tools		*23.0.2*	*Not installed*
Android SDK Build-tools		23.0.1	Installed
Android SDK Build-tools		*22.0.1*	*Not installed*
Android SDK Build-tools		*21.1.2*	*Not installed*
Android SDK Build-tools		*20*	*Not installed*
Android SDK Build-tools		*19.1*	*Not installed*
Android N (API 24)			
API 23, N preview			
Android 6.0 (API 23)			
Documentation for Android SDK	23	1	Installed
SDK Platform	23	3	Installed
Android TV ARM EABI v7a System Image	*23*	*3*	*Not installed*
Android TV Intel x86 Atom System Image	*23*	*4*	*Not installed*
Android Wear ARM EABI v7a System Image	*23*	*4*	*Not installed*
Android Wear Intel x86 Atom System Image	*23*	*4*	*Not installed*
ARM EABI v7a System Image	*23*	*3*	*Not installed*
Intel x86 Atom_64 System Image	23	9	Installed
Intel x86 Atom System Image	23	9	Installed
Google APIs ARM EABI v7a System Image	*23*	*14*	*Not installed*
Google APIs Intel x86 Atom_64 System Image	23	14	Installed
Google APIs Intel x86 Atom System Image	23	14	Installed
Google APIs	*23*	*1*	*Not installed*
Sources for Android SDK	23	1	Installed

2. Next, you should have the following variables set in your Environment Variables
    a. ANDROID_HOME pointing to your Android SDK folder.



    b. PATH variable pointing to each of the following:
        i. Java Development Kit Bin
        ii. Android SDK Tools
        iii. Android Platform Tools
        iv. Gradle Bin

As a reference see Ahmed's path variables below:

C:\Program Files\Java\jdk1.8.0_91\bin;

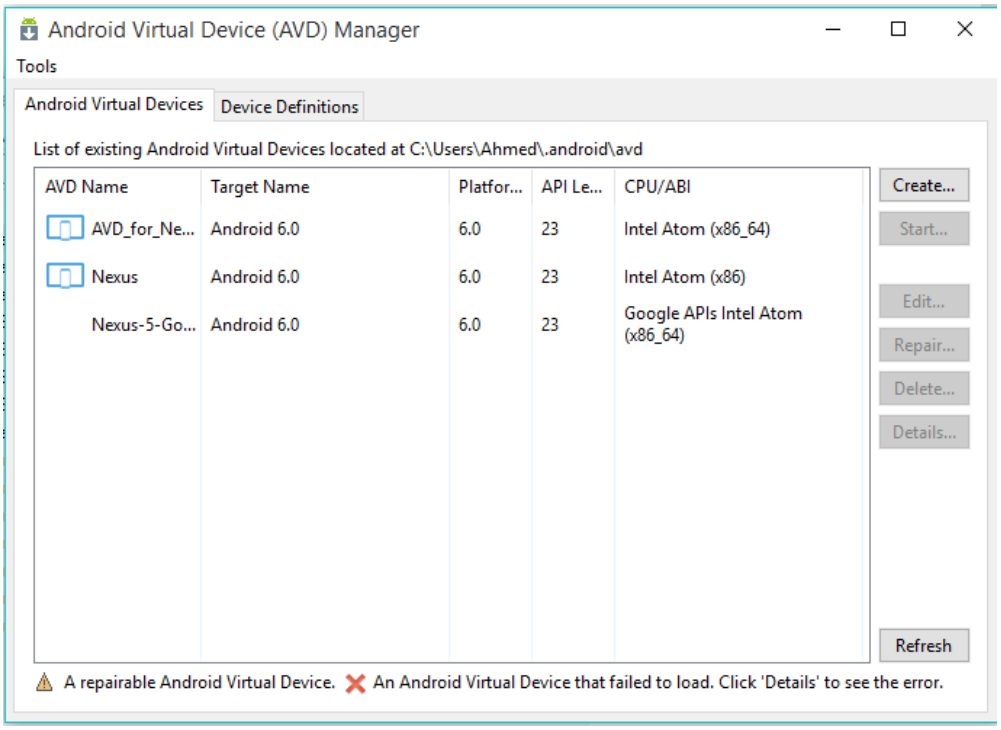
C:\Users\Ahmed\AppData\Local\Android\sdk\tools;

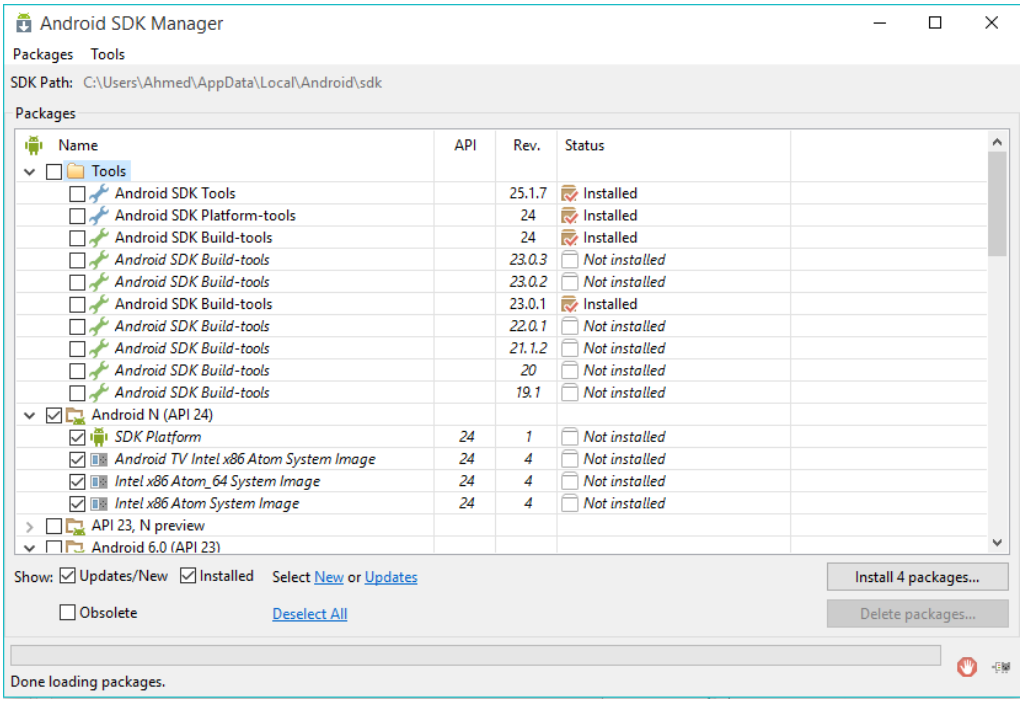
C:\Users\Ahmed\AppData\Local\Android\sdk\platform-tools;

C:\Program Files\Android\Android Studio\gradle\gradle-2.10\bin;

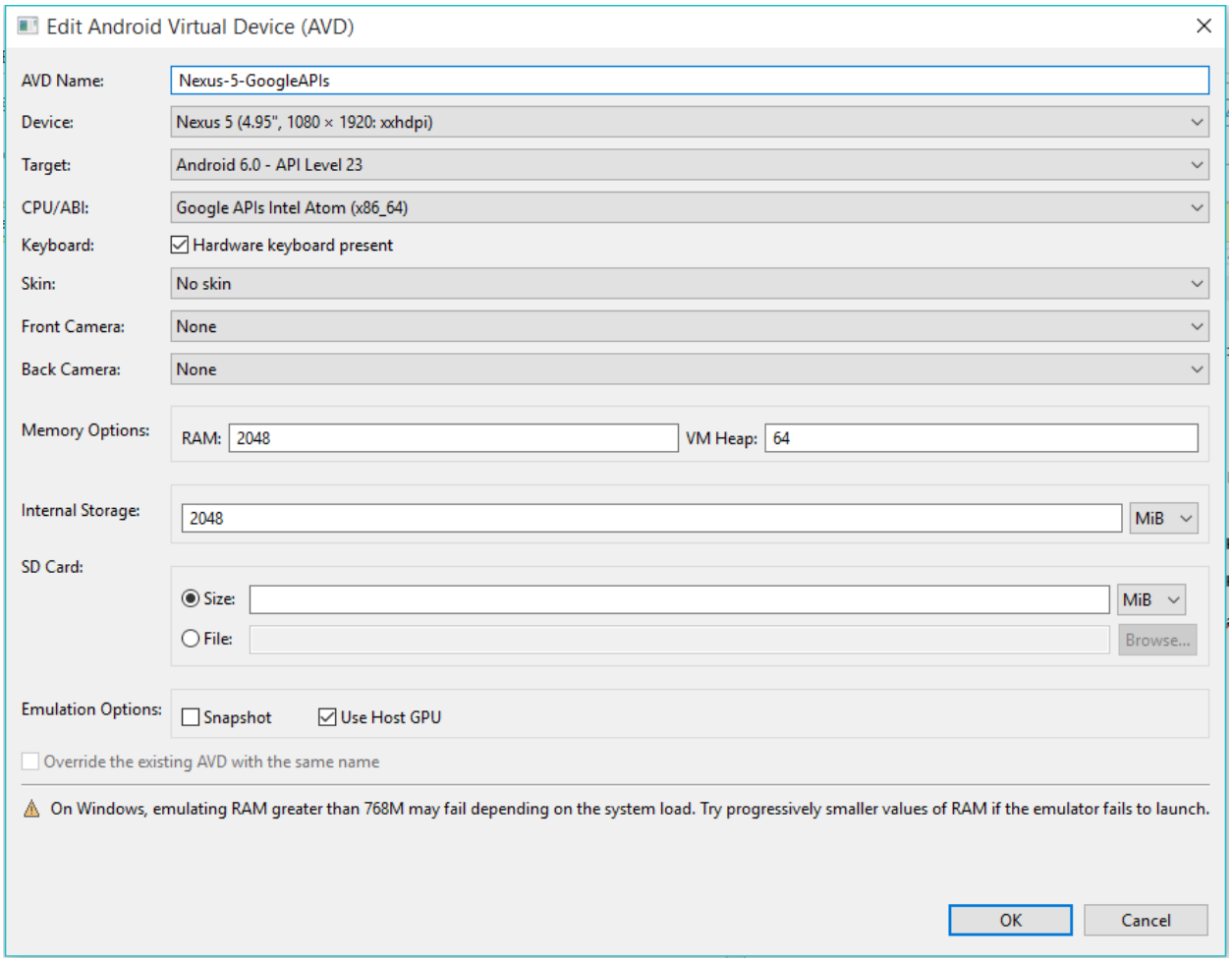
‘

3. Next, you should locate the following files which were installed with Android Studio. These files let you see the emulator options and launch emulators for testing our React-Native Code. Open these executables to confirm they work.
    a. AVD Manager.exe
    b. SDK Manager.exe

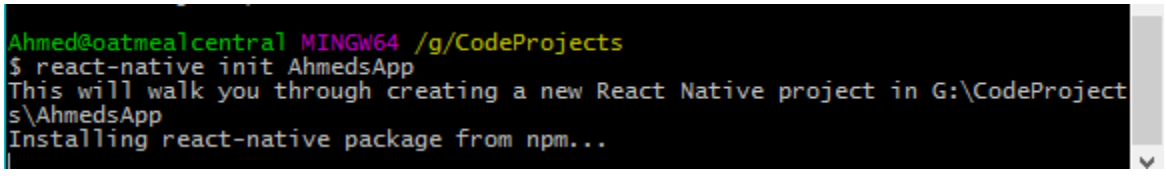




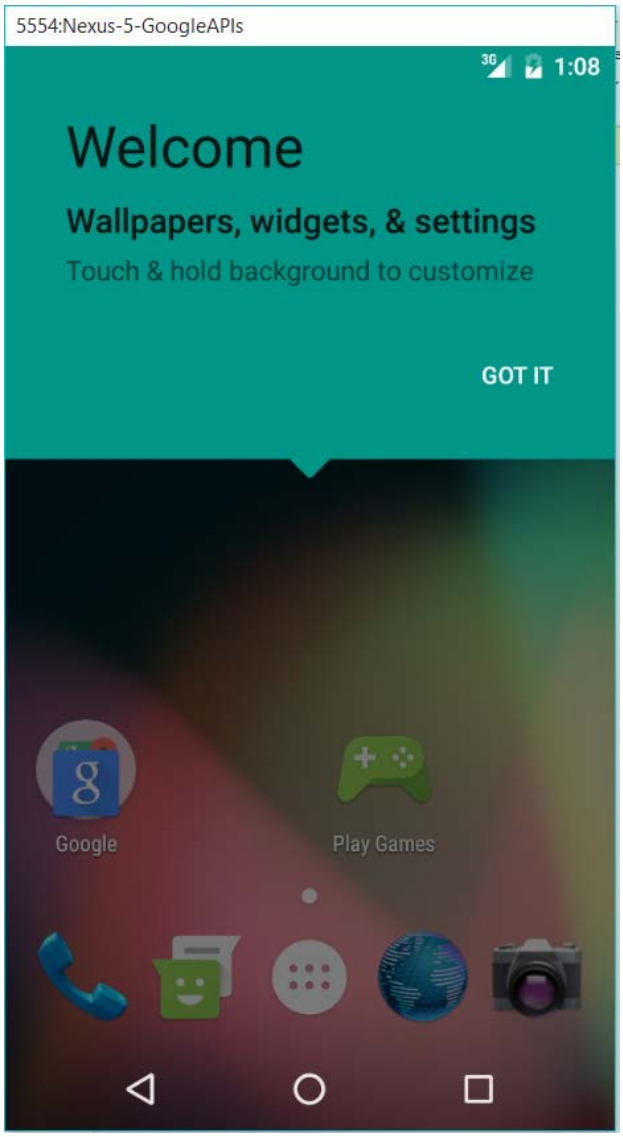
4. Now, create a new AVD in the AVD Manager. Use the settings shown below. Then hit okay and keep this AVD handy. We'll be launching it later in this attempt.



5. Now open up a Bash terminal and navigate to a folder close to your C or D Drive. Then run the command "react-native init <APPNAME>". Replace <APPNAME> with the name you'd like to call your react application. Be expecting to wait a long while.



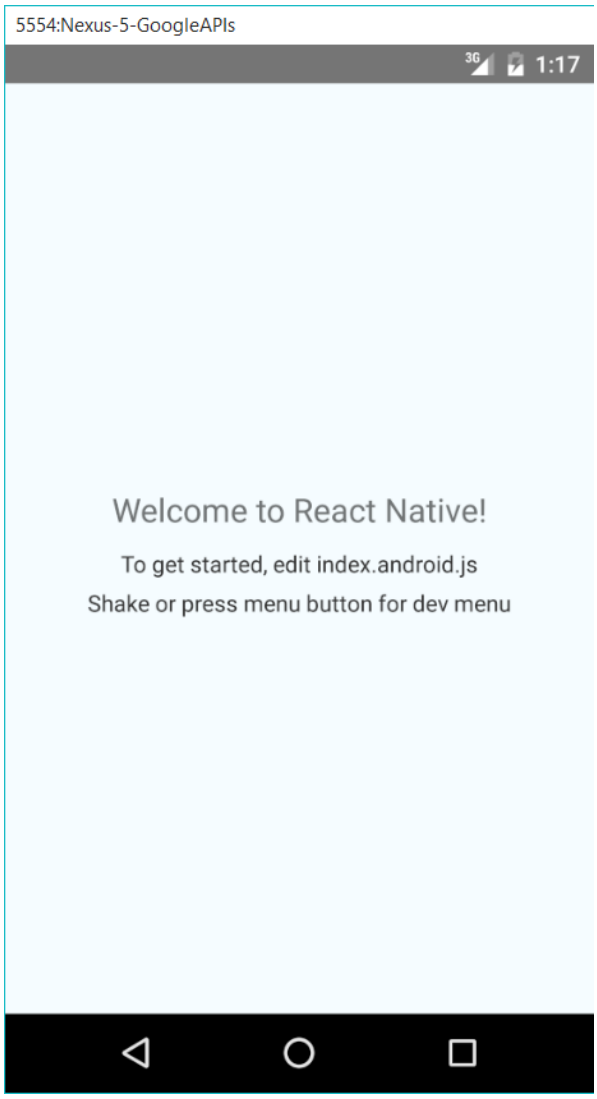
6. Once it’s complete navigate into the folder. Keep this bash window handy. Then return to your AVD Manager and click “Start” to launch the Android Emulator you crated earlier. This will literally create an emulated version of a smartphone on your computer. Let the emulator load until reaches the Android Home screen.



7. Now return to your bash window and navigate into your react-native app's directory. Then run the command "react-native run-android". This process will take some time.

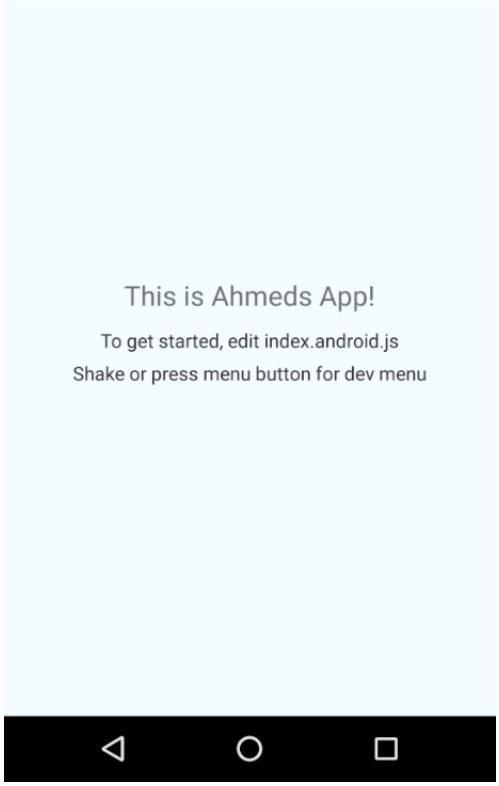
```
Ahmed@oatmealcentral MINGW64 /g/CodeProjects/AhmedsApp
$ react-native run-android
Starting JS server...
Building and installing the app on the device (cd android && gradlew.bat install
Debug...
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:checkDebugManifest
:app:preReleaseBuild UP-TO-DATE
:app:prepareComAndroidSupportAppcompatV72301Library UP-TO-DATE
:app:prepareComAndroidSupportRecyclerviewV72301Library UP-TO-DATE
```

8. Once the process is complete, the output should render to the emulator.



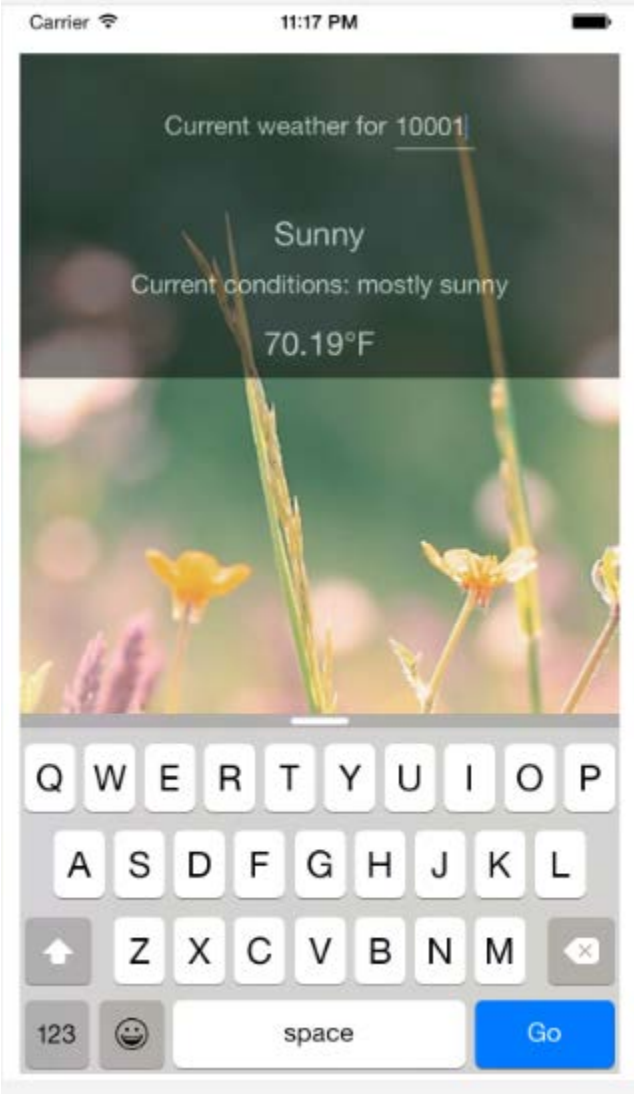
9. Once you get to this point, begin to investigate how you can change up the code in the index.android.js file. Then re-run the command “react-native run-android”. This will re-compile your code and refresh the emulator.

```
class AhmedsApp extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.welcome}>
          This is Ahmeds App!
        </Text>
        <Text style={styles.instructions}>
          To get started, edit index.android.js
        </Text>
        <Text style={styles.instructions}>
          Shake or press menu button for dev menu
        </Text>
      </View>
    );
  }
}
```



10. Once you get the basic changes working, begin looking into Core Components on the React Native documentation (https://facebook.github.io/react-native/docs/tutorial-core-components.html). Try to incorporate them into your application.

11. Your goal for the next two days is to begin aiming to build the following application. It will require a good bit of research (for which the instructor will be doing so along-side), but take your best stab!



Note: As a reference, here is a project with roughly the same goal.

https://github.com/bonniee/learning-react-native/tree/master/WeatherProject

The challenge here is that the code in this example does not work with the latest version of React and it also uses ES5 instead of ES6. Work with one another to dissect the code. As a suggestion, break the group into a team of “front-end” developers and “back-end” developers. Try to have some research how to get the form input, some to look into flexbox and backgrounds, some to look into API calls with fetch, etc.

Good luck!