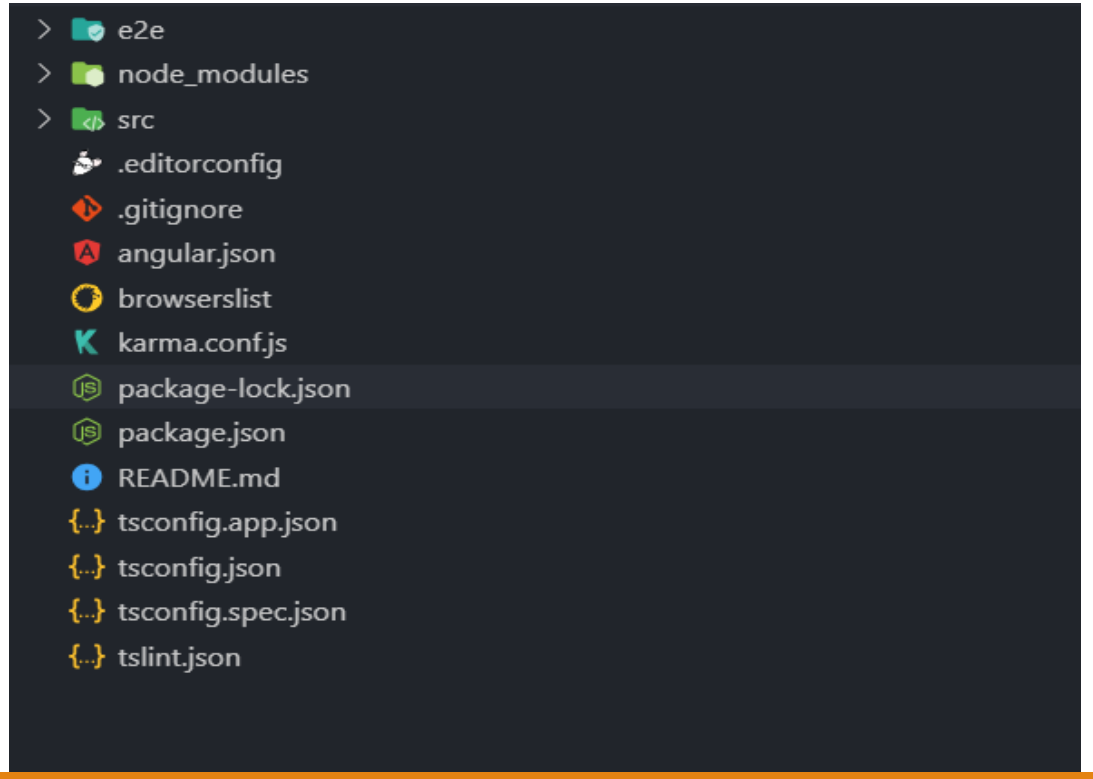
Mobile Application Development

LECTURE 4

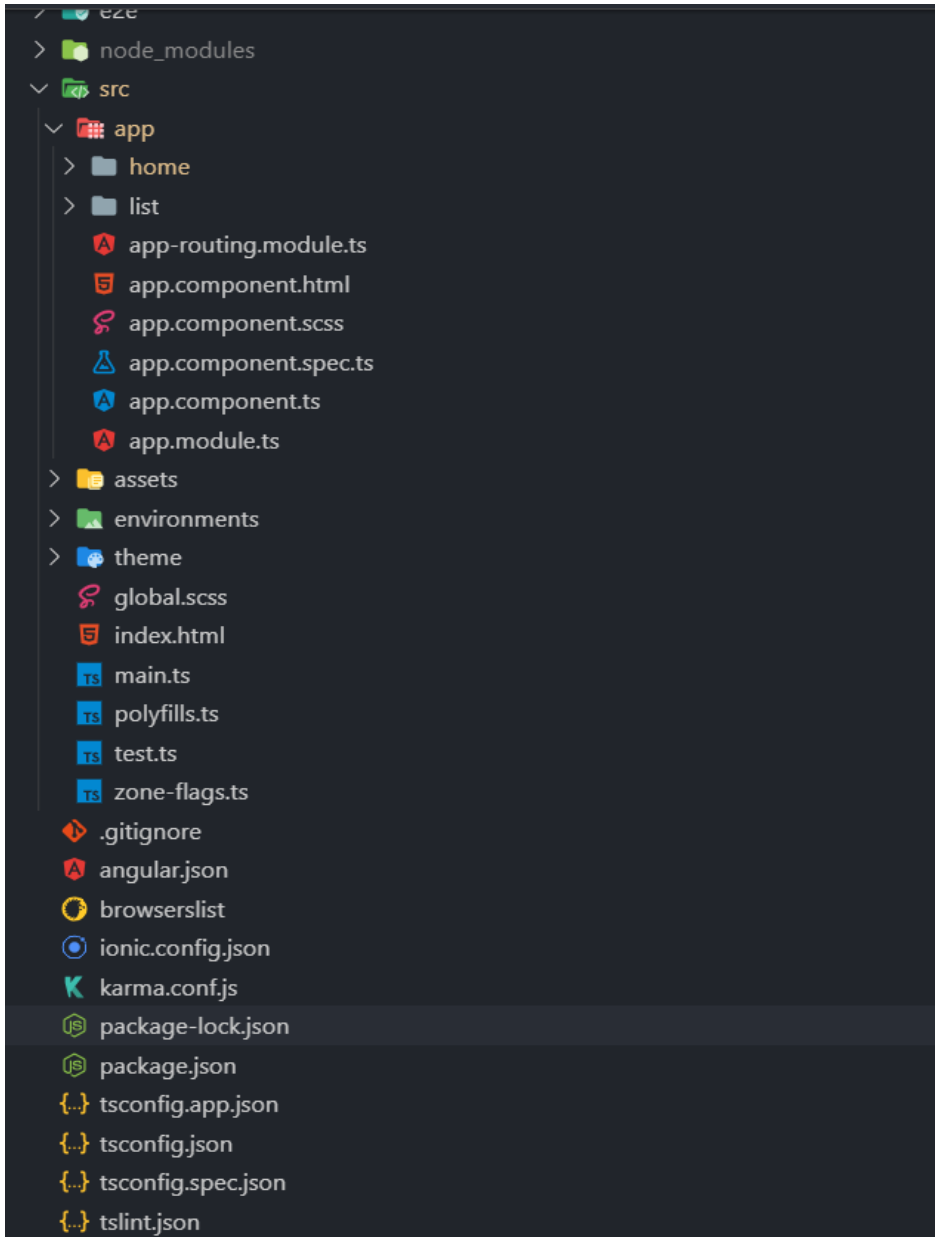
Angular and Ionic Folder Structure

Ionic

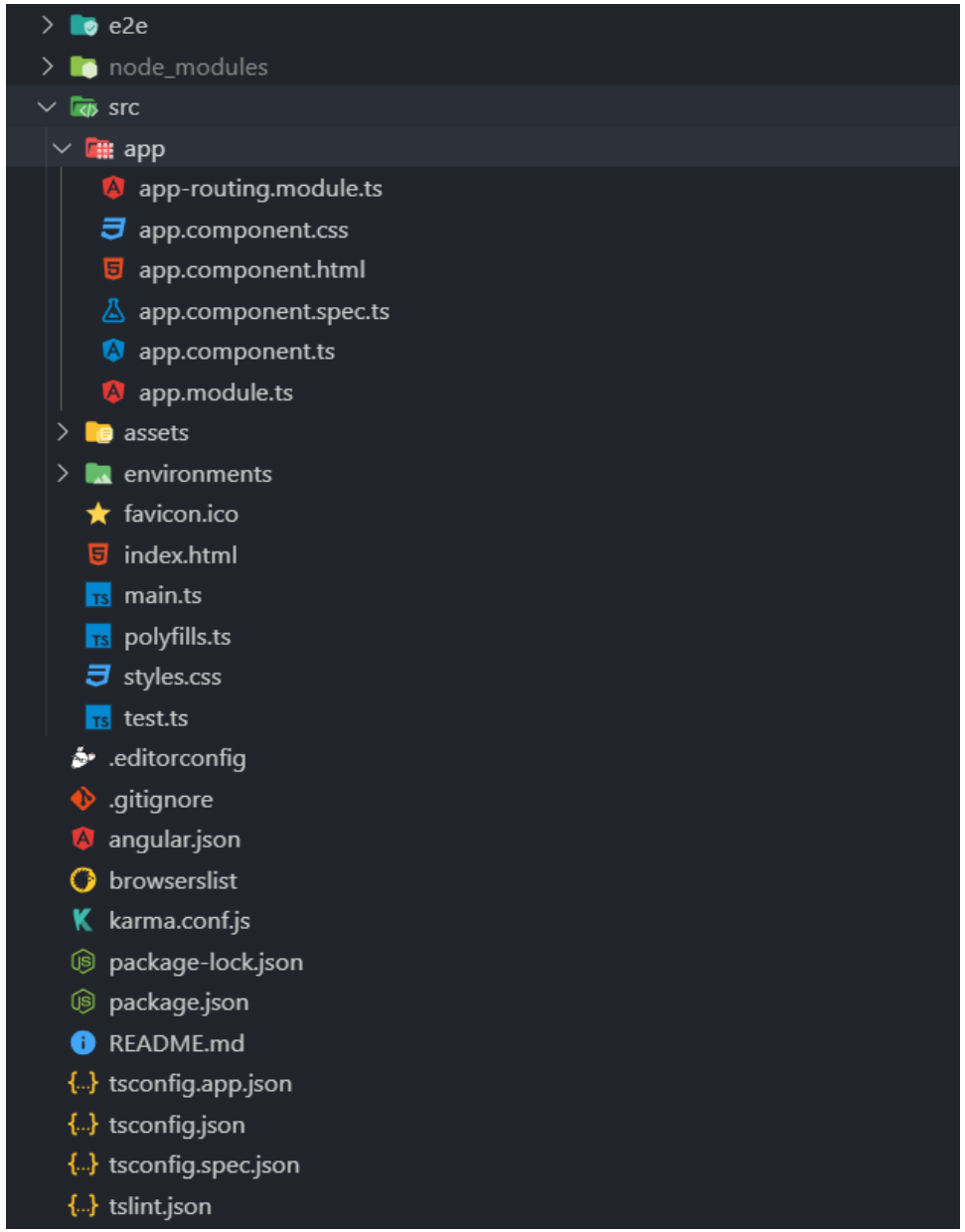
Angular



Ionic



Angular



Angular and Ionic Folder Structure

- Point is, there isn’t really much of a difference.
- Lets dive into each and every file/folder. We don’t need to understand how every is working right now but only understand what it’s doing.
- Complete detail here https://angular.io/guide/file-structure

Angular and Ionic Folder Structure

Package.json

- If you see a package.json file, there a few common things e.g name, version etc. However there's also few important properties which you need to know.

- Scripts, dependencies and devDependencies.

- **Scripts** include command line scripts which can be run to do various tasks. Scripts are basically shortcuts to common tasks e.g running a app, testing an app, deploying an app.

Angular and Ionic Folder Structure

Package.json

- Running a script is as simple as running ***npm* run `scriptname`**
- E.g "scripts": { "build": "node app.js", "test": "standard" }
- Now running **npm run build** inside the directory will actually run **node app.js** command.
- The **dependencies** property is used to define the dependencies that a module/app needs to run in production. These dependencies will install when running **npm install** or **yarn install**

Angular and Ionic Folder Structure

Package.json

- The **devDependencies** property is usually used to define the dependencies the module needs to run in development.**npm install –production** won't install devDependencies but **npm install** will.

Index.html

Basically where your app loads

.gitignore

Files/folders u dont need to be versioned controlled. Or you dont want Git to track. These files / folders won't be uplaoded to github. Includes node_modules and environment files.

Angular and Ionic Folder Structure

Package.json

- The **devDependencies** property is usually used to define the dependencies the module needs to run in development.

Index.html

Basically where your app loads

.gitignore

Files/folders u dont need to be versioned controlled. Or you dont want Git to track. These files / folders won't be uploaded to github. Mostly includes node_modules and environment files.

Angular and Ionic Folder Structure

Ionic.config.json

Configurations for your ionic app.

Karma.config

Karma is a test runner for JavaScript. It is used alongside Jasmine/Mocha for Unit Testing. This file has configuration for Karma test runner.

Angular.json

Configurations of our angular app. You can import your styles and JS files from here. In scripts and styles.

Angular and Ionic Folder Structure

Node_modules

Contains all the modules / packages your app needs to run. These packages are defined in package.json

Global.scss / style.css

Styles that are available throughout the app.

E2E (folder)

Contains files for End to End Testing. That is done using Protractor / Selenium.

Angular and Ionic Folder Structure

Environment folder

Multiple environment configurations for your angular app.

SRC

All your project code goes here.

Assets

All public icons, images go here.

Angular and Ionic Folder Structure

App

All Angular / Ionic apps have a base App folder. It will usually have 6 files. Every Module should have these 6 files as well (with name according to the module name).

App-routing.module.ts (all the routing for your app)

App.module.ts (module configurations)

App.component.ts (TypeScript code for your component)

App.component.html (View of your component)

App.component.scss (sass file for styling your component)

App.component.spec.ts (unit test file for your component)

“We live on an island surrounded by a sea of ignorance. As our island of knowledge grows. so does the shore of our ignorance. ~ John Archibald Wheeler

NPM (Node Package Manager)

- An online repository for the publishing/downloading of open-source **Node**.js projects.
- A command-line utility that aids in package installation, version management, and dependency management.
- Biggest package manager in the world. Most packages written in any programming language.
- For C#, we have Nuget, for Java we have Maven/Gradle. For JS, we have NPM

NPM local vs global packages

- Remember when we did **npm install –g ionic** ? That was a global package. Meaning you can access it anywhere in your system. You can run **ionic –v** and the cmd will pick it up.

- Local packages are the ones downloded in node_modules of the project repository and are available only inside that project. E.g **npm install font-awesome** or **npm i jquery –save.**

- If you want to install a specific version of a package instead of the latest one, you can do n**pm install** rxjs@6.3.3. this will install 6.3.3 version instead of the latest one. There also is another package manager (**Yarn**).