Monday: Angular File Review

We can now examine the files inside our goals app.

The root folder

```
Goals
|---e2e/
|   |---src/
|       |---app.e2e-spec.ts
|       |---app.po.ts
|   |---proctractor.conf.js
|   |---tsconfig.json
|---node_modules/...
|---src/...
|---.browserslistrc
|---.editorconfig
|---.gitignore
|---.angular.json
|---karma.conf.js
|---package-lock.json
|---package.json
|---README.md
|---ts.config.app.json
|---tsconfig.json
|---tsconfig.spec.json
|---tslint.json
```

e2e/

Inside `e2e` folder live the end-to-end tests. **End-to-end testing** is a methodology used to test whether the flow of an application is performing as designed from start to finish. Angular CLI builds an E2E test framework using Protractor and Jasmine. Testing is outside the scope of this course, hence, we will not be going deeper into it. For general understanding, **Protractor** (not the one used in Maths) and **Jasmine** are tools used for testing code. For those who are curious, check out **E2E testing with Jasmine (https://www.syncfusion.com/succinctly-free-ebooks/angular-testing-succinctly/e2e-testing-with-jasmine)**

node_modules/

Node.js creates this folder and puts all third-party modules listed in package.json inside of it.

.browserslistrc

This file contains configurations for sharing of target browsers and Node.js versions.

.angular.json

This file contains all the CLI configurations for the app including the configuration option for **build**, **serve,** and the test tool that the CLI uses. For more details, check out **Angular workspace configuration (https://angular.io/guide/workspace-config#angular-workspace-configuration)**

.editorconfig

Simple configuration for your editor to make sure everyone that uses your project has the same basic configuration. Most editors support an .editorconfig file. See **http://editorconfig.org (http://editorconfig.org/)** for more information.

.gitignore

Git configuration to make sure autogenerated files are not committed to source control.

karma.conf.js

Unit test configuration for the Karma test runner, used when running ng test.

package.json

npm configuration listing the third-party packages your project uses. You can also add your own custom scripts here.

README.md

Basic documentation for your project, pre-filled with CLI command information. Make sure to enhance it with project documentation so that anyone checking out the repo can build your app!

tsconfig.app.json

It contains Application-specific typescript configurations

tsconfig.spec.json

Contains typescript configuration for application tests.

tsconfig.json

TypeScript compiler configuration for your IDE to pick up and give you helpful tooling.

tslint.json

Linting configuration for TSLint together with Codelyzer, used when running ng lint. Linting helps keep your code style consistent.

src/

We will mostly work inside our *src/* folder:

```
src
|
|---app/
|      | ---app.component.css
|      | ---app-routing.module.ts
|      | ---app.component.html
|      | ---app.component.spec.ts
|      | ---app.component.ts
|      | ---app.module.ts
|---assets/
|      | ---.gitkeep
|---environments/
```

```
|      | ---environment.prod.ts
|      | ---environment.ts
|---favicon.ico
|---index.html
|---main.ts
|---polyfills.ts
|---styles.css
|---test.ts
```

The following are the files and folders found in the `src` folder where our app will live. Angular documentation defines them this way:

app/app.component.{ts,html,css,spec.ts}

Defines the AppComponent along with an HTML template, CSS stylesheet, and a unit test. It is the root component of what will become a tree of nested components as the application evolves.

app/app.module.ts

Defines AppModule, the root module that tells Angular how to assemble the application. Right now it declares only the AppComponent. Soon there will be more components to declare.

assets/*

A folder where you can put images and anything else to be copied wholesale when you build your application.

environments/*

This folder contains one file for each of your destination environments, each exporting simple configuration variables to use in your application. The files are replaced on the fly when you build your app. You might use a different API endpoint for development than you do for production or maybe different analytics tokens. You might even use some mock services. Either way, the CLI has you covered.

favicon.ico

Every site wants to look good on the bookmark bar. Get started with your very own Angular icon.

index.html

The main HTML page rendered when someone visits your site. Most of the time you'll never need to edit it. The CLI automatically adds all js and css files when building your app so you never need to add any <script> or <link> tags here manually.

main.ts

The main entry point for your app. Compiles the application with the Just-in-Time (JIT) compiler and bootstraps the application's root module (AppModule) to run in the browser. You can also use the Ahead-of-Time (AOT) compiler without changing any code by appending the–aot flag to the ng build and ng serve commands.

polyfills.ts

Different browsers have different levels of support of the web standards. Polyfills help normalize those differences. You should be pretty safe with core-js and zone.js, but be sure to check out the Browser Support guide for more information.

styles.css

Your global styles go here. Most of the time you'll want to have local styles in your components for easier maintenance, but styles that affect all of your app need to be in a central place.

test.ts

This is the main entry point for your unit tests. It has some custom configuration that might be unfamiliar, but it's not something you'll need to edit.