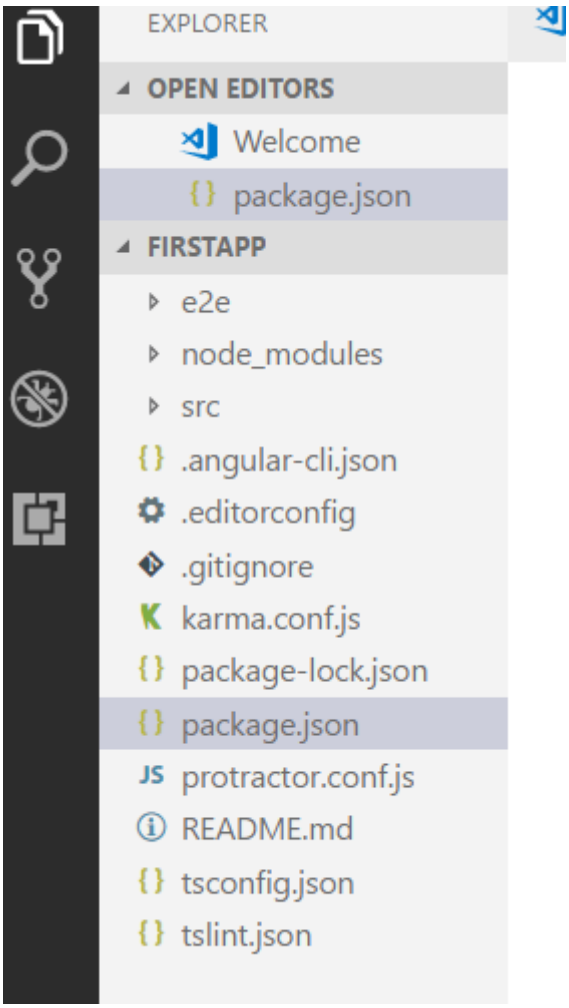
Angular App Folder Structure

When you create a application using Angular CLI you get a application structure as follows,



Let us see, What each file and folder in the folder structure is for

tslint.js

TSLint checks your TypeScript code for readability, maintainability, and functionality errors. tslint.json file is used to configure which rules should run.

README.md

Readme.md is a markdown file for this project. README.md is used to generate the html summary of our projects. Similar to what you see at the bottom of most gut hub projects, it contains information about various commands that can be used to build

run and test this project with Angular CLI. This might come in handy as a quick reference when you have just started with Angular CLI.

protractor.conf.js

Protractor is an end-to-end test framework for Angular applications. Protractor runs tests against your application running in a real browser, interacting with it as a user would. Protractor needs two files to run, a spec file and a configuration file. The protractor.conf.js file contains the configuration information to run the tests.

package.json

package.json file identifies npm package dependencies for the project. package.json file includes two sets of packages, dependencies and devDependencies. The dependencies are essential to running the application. devDependencies are only necessary to develop the application. You can exclude them from production installations.

karma.conf.js

Karma is a JavaScript test runner created by the Angular team. Karma needs to know about the project in order to test it and this is done via this karma configuration file.

angular-cli.json

angular-cli.json file contains configurations related to it's functioning. Like what is the root folder, where to save the out files, which is the default index file, and so on.

.gitignore

gitignore is useful if you are using git for version control. This file specifies intentionally untracked files that Git should ignore.

.editorconfig

EditorConfig file is used to define and maintain consistent coding styles.

The majority of our application is under src/app. And this is where we will be working most of the time. You might create multiple folders under the app folder to organise your application. Like a folder for Home, one for About and so on.

Angular CLI has created a root component for us. Which is app.component.ts. Each component may have its own html and css file and that is what they are.

app.component.spec.ts

The app.component.spec.ts is the spec file for testing your component.

app.module.ts

app.module.ts file is our root module. we will discuss more about what and why we need our root component and root modules in the later sections.

index.ts

index.ts is a barrel module. A barrel is a way to rollup exports from several modules into a single convenience module. So here, instead of importing both app.component and app.module, we can now just import this barrel.

assets

assets is where you may want to keep your application assets like images, fonts, and so on.

environment

The environment folder consists of two environment files one for dev and one for production. The list of which env maps to which file can be found in angular-cli.json.

main.ts

main.ts file is used tell Angular to start up our application. This code initializes the platform that our application runs in, browser in this case and then uses the platform to bootstrap our AppModule.

polyfill.ts

A polyfill is a browser fallback, made in JavaScript, that allows functionality you expect to work in modern browsers to work in older browsers, e.g., to support canvas (an HTML5 feature) in older browsers. polyfills.ts is used to list them.

style.css

styles.css is used to add global styles and also import other style files.

test.ts

test.ts is used to initialize the Angular testing environment.

tsconfig.json

tsconfig.json defines how the TypeScript compiler generates JavaScript from the project's files. This file contains options and flags that guide the compiler as it generates JavaScript files.

typings.d.ts

Typings is the simple way to manage and install TypeScript definitions. typings.d.ts is a Typings reference file.

node_modules

When we create an application using Angular CLI, it automatically downloads all the dependencies and saves them into this folder. It basically read's the "package.json" file in the current directory and installs all the package's dependencies into this folder.

e2e

Finally, we have a folder e2e that stands for end to end testing and this folder consists of the spec and configuration files related to end to end testing using protractor.

Now that we know what an angular 2 application is made of, let's jump into understanding the building blocks of Angular 2 applications.