Angular2 file descriptions and Bootstrap Process

\*This is assuming the new project was created using the angular-CLI and the default files were created for you

tslint.json

First of all, we should explain what TSLint is. TSLint basically checks TypScript code quality. TypeScript is what Angular2 is written in before it is compiled to JavaScript. Hence the .ts files. While you type in your code editor, you could install a TSLint extension that will check you code to ensure it is correct by showing errors for incorrect syntax, declarations, imports, etc…

The tslint.json file can be thought of as the settings or rules that the TSLint extension uses to find these errors. “The configuration file specifies which rules are enabled and their options” (chrismbarr, 2017). More detailed information can be found here: <https://github.com/palantir/tslint#supported-rulesNow>. When the angular-CLI compiles a project for you, it seems to validate the typescript using a TSLint of its own, so it’s important to ensure that the code editor and typescript versions are the same.

main.ts

This is the entry point for the application. You could say that this is the transition barrier between out master (or root) module and the browser. Although it can import further things, the two main jobs this file performs are importing and using the “platformBrowserDynamic”, which is in turn used to load (bootstrap) the master module (ie: /src/app/app-module.ts). For a brief description, it contains client side code that processes templates and reflective dependency injection. Here is the main bootstrap code that is loaded:

*platformBrowserDynamic().bootstrapModule(AppModule);*

Basically, the bootstrapModule is a function of the platformBrowserDynamic class that is used to load the AppModule, which represents the root module of the Angular application.

References

chrismbarr. (2017). *An extensible linter for the TypeScript language.* Retrieved from https://github.com/palantir/tslint#supported-rules