

LAB1 - Inspecting a first project

Prerequisite

As we know already, Angular is written in **TypeScript**. We need **Node** and **npm** to compile the files into **JavaScript** after that, we can deploy our application. For this purpose, **Node.js** must be installed in your system. Hopefully, you have installed **Node.js** on your machine.

1. Install Node.js Head over the Node.js official website and download the the LTS version: <https://nodejs.org/en/> After the installation is completed, you can check it using the below command :

```
node --version
```

npm is installed by default with Node.js:

```
npm -v
```

2. Install Angular CLI Once npm is installed, you can install the Angular CLI by running the command:

```
npm install -g @angular/cli
```

This will install the latest version of the CLI, you can check it by running:

```
ng version
```

Creating first application

Now, we use the Angular CLI to create our first application, it will a task management application, we will name is **TaskManager**, run the following command:

```
ng new TaskManager
```

ng new is one of the command of the **ng** CLI application. It will be used to create new application. It will ask some basic question in order to create new application. It is enough to let the application choose the default choices. Regarding routing question as mentioned below, specify **No**. We will see how to create routing later in the **Routing** chapter and for the stylesheet format select **CSS**.

Inspecting files

Let us check the structure of the application. The structure of the application is as follows :

```
| ---node-modules
+ ---src
  +---app
    | app.component.css
    | app.component.html
    | app.component.spec.ts
    | app.component.ts
    | app.module.ts
  + ---assets
  + ---environments
    | environment.prod.ts
    | environment.ts
  | favicon.ico
  | index.html
  | main.ts
  | polyfills.ts
  | styles.css
  | test.ts
| angular.json
| package.json
| tsconfig.json
```

- **angular.json** Angular project configuration
- **tsconfig.json** Typescript transpiler configuration
- **package.json** List of npm dependencies
- **main.ts** The entry point of the application, it bootstrap the root module which is by default *AppModule*
- **src/app/app.module.ts** Contains the *AppModule* configuration, the root module of the application it bootstraps the root component which is by default *AppComponent*
- **src/app/app.component.ts** Contains the Typescript class definition of the *AppComponent*

- **src/app/app.component.html** The HTML template of the *AppComponent*
- **src/app/app.component.css** The CSS stylesheet of the *AppComponent*
- ***.spec.ts** (Files used for unit testing)

Running the application

You can start the application by running this command in the project directory:

```
ng serve
```

Here, **serve** is the sub command used to compile and run the Angular application using a local development web server. **ng server** will start a development web server and serves the application under port by default : **4200**.

Generate code

Head over the **src/app** folder and run the following command to create a new component :

```
ng generate component first
```

A new component named *FirstComponent* will be created, notice the changes in the **src/app** folder

```
+---src
  +---app
    +---first
      |  first.component.css
      |  first.component.html
      |  first.component.spec.ts
      |  first.component.ts
```

Also, *FirstComponent* is automatically added to the *AppModule* declarations:

```
@NgModule({
  declarations: [
    AppComponent,
    FirstComponent
  ],
```

```
imports: [  
    BrowserModule  
],  
providers: [],  
bootstrap: [AppComponent]  
})
```