**AngularJS 1.6**

**Install Nodejs and Angular CLI(Command Line Interface):**

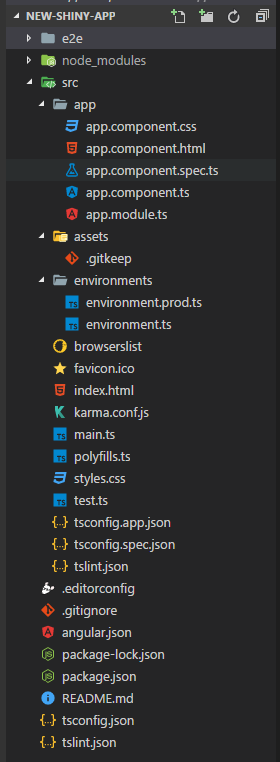
Download and install NodeJS, then from the command prompt type,

npm install –g @angular/cli@version

ng –v -Check the angular CLI version

ng –new projectName - It will create a projectName directory and import all the dependent files.

**Project Structure:**



* **e2e** − end to end test folder. Mainly e2e is used for integration testing and helps ensure the application works fine.
* **node\_modules** − The npm package installed is node\_modules. You can open the folder and see the packages available.
* **src** − This folder is where we will work on the project using Angular 6.
* **angular.json -** configuration schema for the whole project and manipulated by the CLI - including managing of different environments, testing, proxy, third-party resources.
* **karma.conf.js** − This is used for unit testing via the protractor. All the information required for the project is provided in karma.conf.js file.
* **package.json**− The package.json file tells which libraries will be installed into node\_modules when you run npm install.
* **protractor.conf.js** − This is the testing configuration required for the application.
* **tsconfig.json**− This basically contains the compiler options required during compilation.
* **tslint.json** − This is the config file with rules to be considered while compiling.
* **app.module.ts** − If you open the file, you will see that the code has reference to different libraries, which are imported. Angular-cli has used these default libraries for the import - angular/core, platform-browser. The names itself explain the usage of the libraries.
* They are imported and saved into variables such as declarations, imports, providers, and bootstrap.

**import { BrowserModule } from '@angular/platform-browser';**

**import { NgModule } from '@angular/core';**

**import { AppComponent } from './app.component';**

**@NgModule({**

**declarations: [**

**AppComponent**

**],**

**imports: [**

**BrowserModule**

**],**

**providers: [],**

**bootstrap: [AppComponent]**

**})**

**export class AppModule { }**

* **declarations** − In declarations, the reference to the components is stored. The Appcomponent is the default component that is created whenever a new project is initiated. We will learn about creating new components in a different section.
* **imports** − This will have the modules imported as shown above. At present, BrowserModule is part of the imports which is imported from @angular/platform-browser.
* **providers** − This will have reference to the services created. The service will be discussed in a subsequent chapter.
* **bootstrap**− This has reference to the default component created, i.e., AppComponent.
* **app.component.css** − You can write your css structure over here. Right now, we have added the background color to the div as shown below.
* **app.component.spec.ts** − These are automatically generated files which contain unit tests for source component.
* **app.component.ts** − The class for the component is defined over here. You can do the processing of the html structure in the .ts file. The processing will include activities such as connecting to the database, interacting with other components, routing, services, etc.

The structure of the file is as follows −

**import { Component } from '@angular/core';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']**

**})**

**export class AppComponent {**

**title = 'app';**

**}**

* **Assets** -You can save your images, js files in this folder.
* **environment.prod.ts, environment.ts** -Both the files have details of whether the final file should be compiled in the production environment or the dev environment.
* **index.html**

<body>

**<app-root></app-root>**

</body>

* The body has <app-root></app-root>. This is the selector which is used in app.component.ts file and will display the details from app.component.html file.
* **main.ts** - is the file from where we start our project development. It starts with importing the basic module which we need. Right now if you see angular/core, angular/platform-browser-dynamic, app.module and environment is imported by default during angular-cli installation and project setup.

**import { enableProdMode } from '@angular/core';**

**import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';**

**import { AppModule } from './app/app.module';**

**import { environment } from './environments/environment';**

**if (environment.production) {**

**enableProdMode();**

**}**

* When AppModule is called, it calls app.module.ts which further calls the AppComponent based on the boostrap as follows,

bootstrap: [AppComponent]

* **app.component.ts -** there is a selector: app-root which is used in the index.html file. This will display the contents present in app.component.html.
* **polyfill.ts** - This is mainly used for backward compatibility.
* **styles.css** - This is the style file required for the project.
* **test.ts** - unit test cases for testing the project will be handled.
* **tsconfig.app.json** - This is used during compilation, it has the config details that need to be used to run the application.
* **tsconfig.spec.json** - This helps maintain the details for testing.
* **typings.d.ts** - It is used to manage the TypeScript definition.

**Flow of execution:**

angular-cli.json -> main.ts(load Env,. and app module)->app.module.ts (all components and bootstrap)->app.component.ts ->render the selector with HTML template and css.