* To create a project in Angular 4, we will use the following command.

ng new projectname

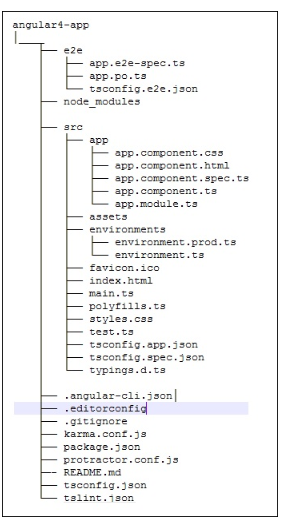
* To compile our project with the following command .

ng serve

* The web server starts on port 4200. Type the url **http://localhost:4200/** in the browser and see the output.
* You can change the port if you wish using the following command −

ng serve --host 0.0.0.0 –port 4205

The Angular 4 app folder has the following **folder 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 4.

**app** -It contains the files described below. These files are installed by angular-cli by default

* app.module.ts
* app.component.css
* app.component.html
* app.component.spec.ts
* app.component.ts

**Assets -** You can save your images, js files in this folder.

**Environment -**This folder has the details for the production or the dev environment. The folder contains two files.

* environment.prod.ts
* environment.ts

**favicon.ico-**This is a file that is usually found in the root directory of a website.

**main.ts -**main.ts is the file from where we start our project development. It starts with importing the basic module which we need.

**polyfill.ts-** This is mainly used for backward compatibility.

**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.

**package.json** − The package.json file tells which libraries will be installed into node\_modules when you run npm install.

**.angular-cli.json** − It basically holds the project name, version of cli, etc.

**.editorconfig** − This is the config file for the editor.

**.gitignore** − A .gitignore file should be committed into the repository, in order to share the ignore rules with any other users that clone the repository.

**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

# **Angular 4 - Components**

Components are basically classes that interact with the .html file of the component, which gets displayed on the browser.

The file structure has the app component and it consists of the following files −

* **app.component.css**
* **app.component.html**
* **app.component.spec.ts**
* **app.component.ts**
* **app.module.ts**

To create component run below command

ng g component new-cmp

* Index.html internally refers to main.ts which calls the parent module, i.e., AppModule
* When AppModule is called, it calls app.module.ts which further calls the AppComponent based on the boostrap
* In 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.

Flow :

**Index.html----🡪main.ts -----🡪app.module.ts-----🡪app.components.ts--🡪app.component.html**

**Index.html**

<!doctype html>

<html lang = "en">

<head>

<meta charset = "utf-8">

<title>Angular 4App</title>

<base href = "/">

<meta name="viewport" content="width = device-width, initial-scale = 1">

<link rel = "icon" type = "image/x-icon" href = "favicon.ico">

</head>

<body>

<app-root></app-root>

</body>

</html>

**main.ts**

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();

}

platformBrowserDynamic().bootstrapModule(AppModule);

**app.module.ts**

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 { }

**app.component.ts**

import { Component, OnInit } from '@angular/core'; // here angular/core is imported .

@Component({

// this is a declarator which starts with @ sign. The component word marked in bold needs to be the same.

selector: 'app-new-cmp', //

templateUrl: './new-cmp.component.html',

// reference to the html file created in the new component.

styleUrls: ['./new-cmp.component.css'] // reference to the style file.

})

export class NewCmpComponent implements OnInit {

constructor() { }

ngOnInit() {}

}

**app.component.html**

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

Welcome to {{title}}.

</h1>

</div>

# **Angular 4 - Module**

**Module** in Angular refers to a place where you can group the components, directives, pipes, and services, which are related to the application.

In case you are developing a website, the header, footer, left, center and the right section become part of a module.

To define module, we can use the **NgModule**.

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**

It is an array of components created. If any new component gets created, it will be imported first and the reference will be included in declarations as shown below −

declarations: [

AppComponent,

NewCmpComponent

]

### **Import**

It is an array of modules required to be used in the application. It can also be used by the components in the Declaration array. For example, right now in the @NgModule we see the Browser Module imported. In case your application needs forms, you can include the module as follows −

import { FormsModule } from '@angular/forms';

The import in the **@NgModule** will be like the following −

imports: [

BrowserModule,

FormsModule

]

### **Providers**

This will include the services created.

### **Bootstrap**

This includes the main app component for starting the execution.

# **Angular 4 - Data Binding**

Data Binding is available right from AngularJS, Angular 2 and is now available in Angular 4 as well. We use curly braces for data binding - {{}}; this process is called interpolation.

**app.component.ts**

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'Angular 4 Project!';

// declared array of months.

months = ["January", "Feburary", "March", "April", "May",

"June", "July", "August", "September",

"October", "November", "December"];

isavailable = true; //variable is set to true

}

The month’s array that is shown above is to be displayed in a dropdown in the browser. For this, we will use the following line of code −

<!--The content below is only a placeholder and can be replaced. -->

<div style="text-align:center">

<h1>

Welcome to {{title}}.

</h1>

</div>

<div> Months :

<select>

<option \*ngFor="let i of months">{{i}}</option>

</select>

</div>

<div>

<span \*ngIf = "isavailable">Condition is valid.</span>

//over here based on if condition the text condition is valid is displayed.

If the value of isavailable is set to false it will not display the text.

</div>

The syntax **for** in Angular is **\*ngFor = “let I of months”** and to get the value of months we are displaying it in {{i}}.