1. **Introduction:**- Angular is UI Frame work for building the desktop and mobile web applications. We can use UI Frame work to develop client side applications using HTML,CSS, Type Script. It is open source and type script based UI FrameWork. Angular functionalities are available in the form of packages.Basic required packages are

core

common

compiler

platform-browser

platform-browser-dynamic

router.

* 1. Angular Versions:- Angular 2 is an AngularJS version built around the concept of the component, which allows building JavaScript classes. After releasing Angular JS, the Angular team released Angular 2, which is a complete rewrite of the original Angular 1 or AngularJS. It was rewritten from scratch by the Angular team using Typescript. It offers better performance to web developers.

Angular Version Date Description

Angular 2 14.09.2016 Initial Version of Angular

Angular 4 23.03.2017 Version 4

Angular 5 11.11.2017 Version 5

Angular 6 03-05-2018 Version 6

Angular 7 18-10-2018 Version 7

Angular 8 25-08-2019 Version 8

Angular 9 06-02-2020 Version 9

Angular 10 24-06-2020 Version 10

Angular 10.0.12 24-08-2020 Version 10.0.12

1.2)Angular Js/Angular:-

* Angular js is javascript based UI framework. Angular is typescript based UI framework.
* Angular-js does not support datatypes and oops concepts. Angular supports datatypes and oops.
* The application is written in Angular-js . This application code is visible in browser page.
* The application is written in Angular. This application code is not visible in browser page.

iv Angular js is modular programming. Angular is Component based programming.

1. **Install the Angular**:-

Step1:- install any IDE. For my convience , I will have installed ‘visual studio code’. We can down load it from following URL.

<https://code.visualstudio.com/>

step2:- Install the Node js.

Repository:- The repository is location/place. The packages of various technologies are placed in this location. There are two types of repositories.

a)Node Repositories.

b) Yarn Repositories.

To download and install required packages from node repository, we need one program. The program is ‘npm’. Npm stands for node package manager.

While installing the node js, the nodejs server and npm are also installed.

1. Download and install nodejs from following URL.

<https://nodejs.org/en/>

1. After installing , we can check the installed nodejs version and npm version.

i)Open command prompt

Type following command line to display node js version.

* Node –v

Type following command line to display npm version.

Npm –v

Example:-

C:\Users\Personal>node -v

v17.0.1

C:\Users\Personal>npm -v

8.1.0

C:\Users\Personal>

Step 3: Install the TypeScript.

i)open command prompt. Type the following command line

> npm install –g typescript

ii) To display type script version. Type the following command line.

>tsc –v

Example:- C:\Users\Personal>tsc -v

Version 4.4.4

Step4:- Install the Angular CLI.

* The Angular CLI stands for Angular Command Line Interface. We install angular using Angular CLI.
* It help us to quickly create Angular application with all the configuration file and package in one single command.
* It also helps us to add features (components,directives,services and etc) to existing angular application.

We use the npm command to install the Angular

-> npm install -g @angular/cli

We can find currenly installed Angular CLI version using command.

* Ng –version

Example:-

\_ \_ \_\_\_\_ \_ \_\_\_

/ \ \_ \_\_ \_\_ \_ \_ \_| | \_\_ \_ \_ \_\_ / \_\_\_| | |\_ \_|

/ △ \ | '\_ \ / \_` | | | | |/ \_` | '\_\_| | | | | | |

/ \_\_\_ \| | | | (\_| | |\_| | | (\_| | | | |\_\_\_| |\_\_\_ | |

/\_/ \\_\\_| |\_|\\_\_, |\\_\_,\_|\_|\\_\_,\_|\_| \\_\_\_\_|\_\_\_\_\_|\_\_\_|

|\_\_\_/

Angular CLI: 12.2.11

Node: 14.18.1

Package Manager: npm 6.14.15

OS: win32 x64

Angular:

...

Package Version

------------------------------------------------------

@angular-devkit/architect 0.1202.11 (cli-only)

@angular-devkit/core 12.2.11 (cli-only)

@angular-devkit/schematics 12.2.11 (cli-only)

@schematics/angular 12.2.11 (cli-only)

1. **Create new Angular Application:-**

3.1)Creating:-

The following command line has to be ran from command prompt.

Syntax: ng new “Application Name”

The Angular CLI Does the following activies while creating new application.

* Creates a new directory with specified name.
* Sets up the folder structure for the application
* Downloads and installs Angular libraries/packages and any other dependencies
* Installs and configures TypeScript
* Installs and configures **Karma & Protractor** for testing.

Example:- I created folder on desktop named sukumar. In that folder , I am going to create first angular project.

C:\Users\Personal\Desktop\sukumar>ng new first-app

? Would you like to add Angular routing? Yes

? Which stylesheet format would you like to use? CSS

CREATE first-app/angular.json (3057 bytes)

CREATE first-app/package.json (1073 bytes)

CREATE first-app/README.md (1055 bytes)

CREATE first-app/tsconfig.json (783 bytes)

CREATE first-app/.editorconfig (274 bytes)

CREATE first-app/.gitignore (604 bytes)

CREATE first-app/.browserslistrc (703 bytes)

CREATE first-app/karma.conf.js (1426 bytes)

CREATE first-app/tsconfig.app.json (287 bytes)

CREATE first-app/tsconfig.spec.json (333 bytes)

CREATE first-app/src/favicon.ico (948 bytes)

CREATE first-app/src/index.html (294 bytes)

CREATE first-app/src/main.ts (372 bytes)

CREATE first-app/src/polyfills.ts (2820 bytes)

CREATE first-app/src/styles.css (80 bytes)

CREATE first-app/src/test.ts (788 bytes)

CREATE first-app/src/assets/.gitkeep (0 bytes)

CREATE first-app/src/environments/environment.prod.ts (51 bytes)

CREATE first-app/src/environments/environment.ts (658 bytes)

CREATE first-app/src/app/app-routing.module.ts (245 bytes)

CREATE first-app/src/app/app.module.ts (393 bytes)

CREATE first-app/src/app/app.component.html (24617 bytes)

CREATE first-app/src/app/app.component.spec.ts (1082 bytes)

CREATE first-app/src/app/app.component.ts (213 bytes)

CREATE first-app/src/app/app.component.css (0 bytes)

√ Packages installed successfully.

'git' is not recognized as an internal or external command,

operable program or batch file.

3.2) Running the new Angular project:-

1.Go to the Application folder/directory.

2. To run our new application, we should run following command at command prompt.

Syntax: ng serve [--open]

The above command compiles the angular application and starts the “webpack development server”. The Webpack Development server listens on HTTP Port 4200. Hence open the browser and type http://localhost:4200/ and you will see new created app is running displayed on the browser.

C:\Users\Personal\Desktop\sukumar>cd first-app

C:\Users\Personal\Desktop\sukumar\first-app>ng serve --open

- Generating browser application bundles (phase: setup)...Compiling @angular/core : es2015 as esm2015

Compiling @angular/common : es2015 as esm2015

Compiling @angular/platform-browser : es2015 as esm2015

Compiling @angular/router : es2015 as esm2015

Compiling @angular/platform-browser-dynamic : es2015 as esm2015

√ Browser application bundle generation complete.

Initial Chunk Files | Names | Size

vendor.js | vendor | 2.39 MB

polyfills.js | polyfills | 510.58 kB

styles.css, styles.js | styles | 383.36 kB

main.js | main | 57.18 kB

runtime.js | runtime | 6.62 kB

| Initial Total | 3.33 MB

Build at: 2021-10-23T13:10:27.710Z - Hash: 8f06c35cd7c444a32eaa - Time: 61049ms

\*\* Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ \*\*

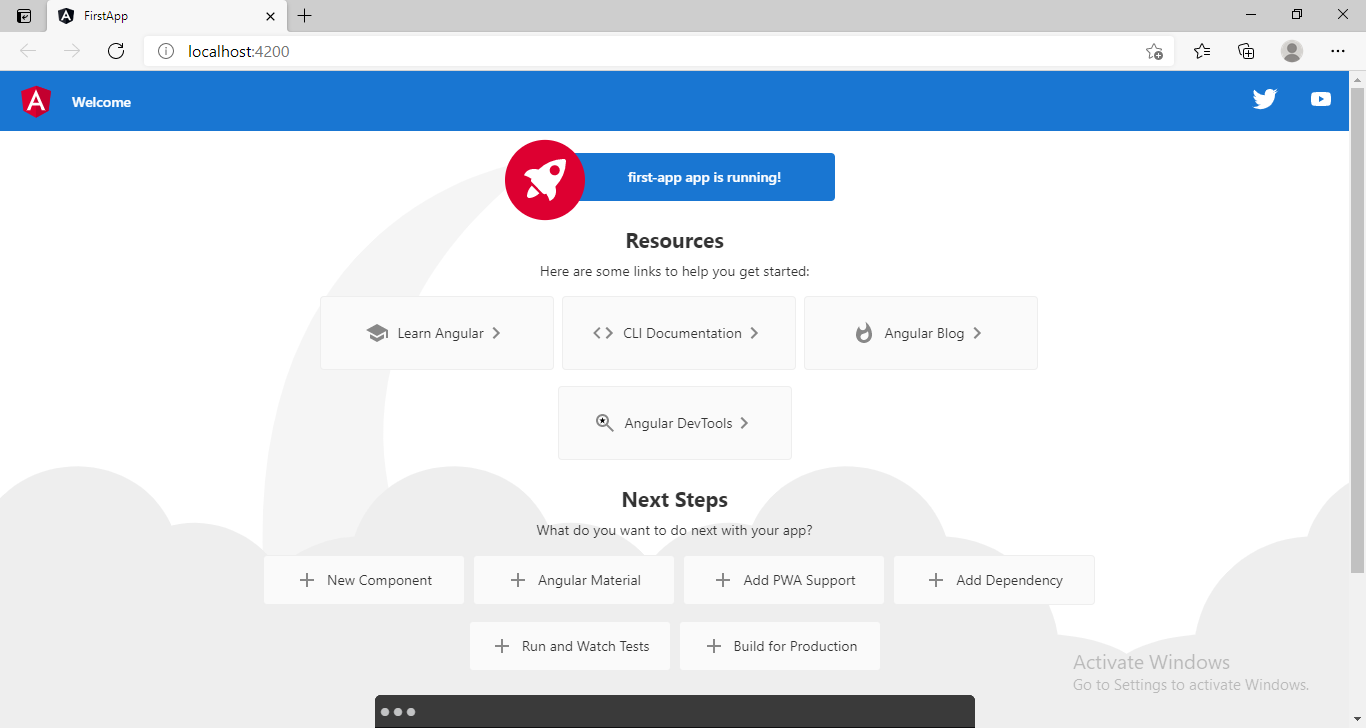
√ Compiled successfully.

√ Browser application bundle generation complete.

5 unchanged chunks

Build at: 2021-10-23T13:10:30.092Z - Hash: f8c23f68fb070054902e - Time: 799ms

√ Compiled successfully.

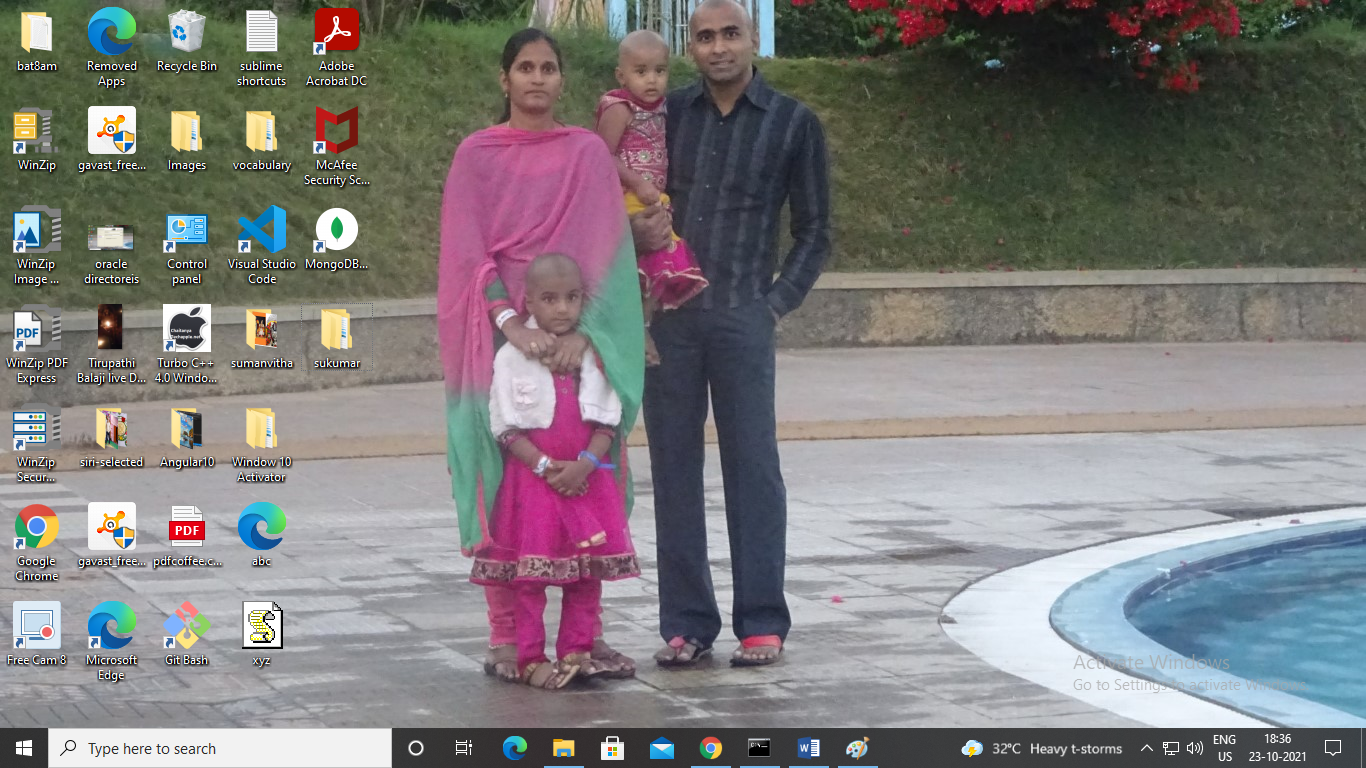


After pressing the ctrl+c , we see that

^CTerminate batch job (Y/N)? y

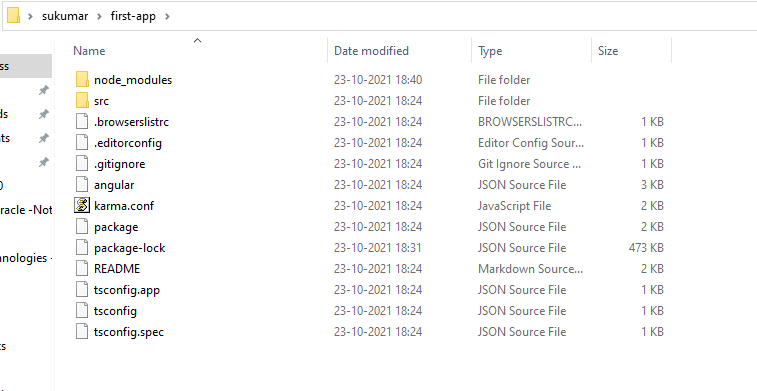
Now we will see the prompt symbol.

3.3)Folder Structure:



Step1:- Click on the sukumar folder.

Step2:- then after click the , first-app folder. Then we will see the following folders and files in that directory.



1.node\_modules:- This folder contains the Angular library/packages (folders of the packages) which are installed into your application when you created a project in angular. If you installed any third-party Libraries(BootStrap/Jquery) then also their folders are going to be stored within this node\_modules folder.  These libraries are useful for purely development.

you shouldn't include this folder during deploying your application to production or committing to the git repository. While moving your project to a new location you should skip this folder and run ‘npm install’ in a new location.

Example:

Step1: I created a new angular project(first-app) in folder named ‘sukumar’.

Step2: Now I wanted to move this project from sukumar to sukumar1 folder.

Step3: what I should do is copy all files and folder in first-app project from sukumar except node-module folder.

Paste all of them in first-app project in sukumar1 folder.

Step4: open command prompt .

Change directory to first-app in sukumar1 folder.

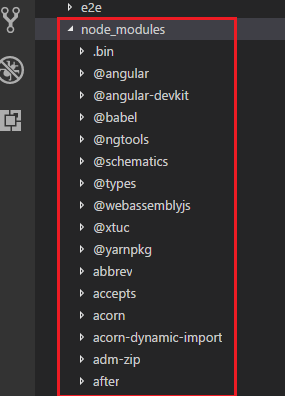
Step5: run the following command.

>Npm install

Npm uses information in package.json file and install node-module directory in new location.

Q) why did not copy the node-module in new location (sukumar1)?

a) The copying operation takes much time.



1. Angular.json:- It contains all the configuration of your Angular Project. It contains the configurations such as

* Project name
* Address of src folder.
* Address of index.html file.
* Address of main.ts file.

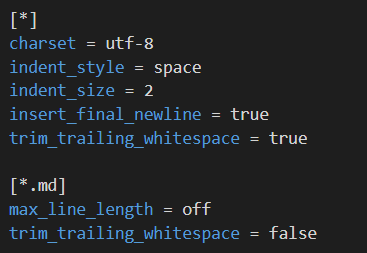
…etc.

1. Package.json:- This file contains

* Project name
* Application version
* Description
* Commands: These are used to run, test the application.(ng serve,ng build,ng test)
* Dependencies: The application requires some packages to work/run correctly. These packages are listed in Dependencies.
* devDependencies: we need some packages while developing application. These packages are not required in production environment. The packages are listed in devdependeices.

1. .gitIgnore file: The files which you want to ignore in the git repository, you need to put within this gitignore file.

5.editorConfig.file:- This file is basically used to set up the team environment. In real-time, many developers may work on a single project. And each developer may follow different coding standards to declare variables, classes, style, size of each character, length, etc. But in the end, we need to merge the code of each developer to produce the final product. At that time, it may produce some error or messy code as each developer having different coding standards. In order to solve this problem, the editor config file is used where the standard rules are defined which needs to be followed by the developers in teamwork. And moreover, the developers do not have access to this file and only the manager or the team lead who defines the rules can only have access to this file. Now, if you open the editor config file, you will find the following rules.



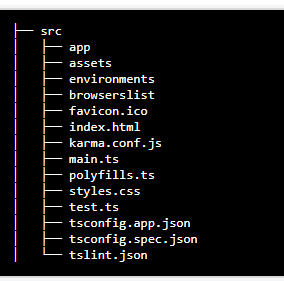
If you develop the component against the rule, then your component is not going to be compiled.

6.ReadMe.md:- File containing a description of our project. All the information which we would like to provide our users with before they start using our app .

7.tslint.json:- This is a tool useful for static analysis that checks our TypeScript code for readability, maintainability, and functionality errors.

8.ts.config.app.json: This file is used during the compilation of your application and it contains the configuration information about how your application should be compiled.

9.Src:- This folder contains actual source code .



1. App:- This is the application folder. Whenever you want to create any component, service, or module, you need to create it within this app folder. By default , every angular project has one module and component.

Defualt module is app.module.ts.

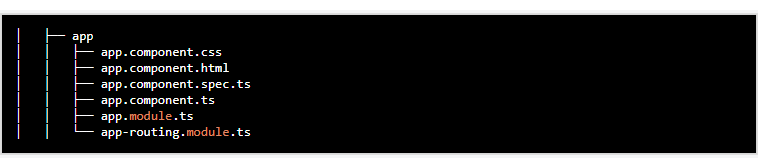
Default component is app.component.ts

The above file are in App folder. Along with them, This folder also contains following files:

App.component.html

App.component.css

App.component.spec.ts



1. Assets:- we usually places static assets like images,icons,..etc in this folder.

3.Environments:- This folder contains two files.

i)environment.ts:- This file is used for development.

ii)environment.prod.ts:-This file is used for production environment.

In this file, we store configuration settings for different environment (development,production) . configuration setting means database credentials and server addresses.

4.favicon.ico:- It is icon file which is displayed on browser.

1. Index.html:- This HTML file contains HTML code with the head and body section. It is the starting point of your application or you can say that this is where our angular app bootstraps. If you open it you will find that there are no references to any stylesheet (CSS) nor JS files.
2. Polyfills.ts:- This file is basically used for browser-related configuration. In angular, you write the code using typescript language. The Polyfills.ts file is used by the compiler to compile your Typescript code to a specific JavaScript method so that it can be parsed by different browsers. The polyfill.ts file imported the required script which is required for running an Angular application. This is because the angular framework uses the latest features of JavaScript which are not available in the current version of JavaScript supported by most of the browsers. So, basically, the polyfills.ts file fills the gap to provide the features of JavaScript that are needed by Angular and the feature supported by the latest browsers. It is mainly used for backward compatibility.

7.Main.ts:- It is typescript start up file. It is starting point to our application like main function in c language.

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)

.catch(err => console.error(err));

8.Style.css:- Global stylesheet file by default means it is where we can add global styles for our applications, including our project.

**4. Boot straping in Angular:**  Bootstrapping is a technique of initializing or loading our Angular application. The Angular takes the following steps to load our first view.

* Index.html loads
* Angular, Third-party libraries & Application loads
* Main.ts the application entry point
* Root Module
* Root Component
* Template.

4.1)index.html loading:- This file contains the following code.

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>GettingStarted</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>

This file does not have javascript and css files. The body tag has only <app-root> and </app-root> tag.

4.1.1)ng server:- This is command. This command build the our application but does not save the compiled application to the disk. It saves it in main memory and starts the development server.

4.1.2)ng build:- This will build and copy the output files to the dist folder. This command is used for debugging/testing.

Ng build –prod This command will build and distribute application for production .

Now open dist folder and index.html file. Now index.html file has following code.

<!DOCTYPE html><html lang="en"><head>

   <meta charset="utf-8">

   <title>SecondApp</title>

   <base href="/">

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

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

 <link rel="stylesheet" href="styles.31d6cfe0d16ae931b73c.css"></head>

 <body>

   <app-root></app-root>

 <script src="runtime.d0e6fb900f0a829d7eee.js" defer></script>

 <script src="polyfills.a7f69eba3eefbc1f2aeb.js" defer></script>

 <script src="main.77867874eaa54cf1be31.js" defer></script>

 </body>

 </html>

4.2) Angular, Third-party libraries & Application loads:- So when index.html is loaded, the Angular core libraries, third-party libraries are loaded.

4.3) Main.ts:- Angular need to identify entry point of application. The entry point of angular application is main.ts. Angular finds the address of main.ts from configuration file which is angular.json.

4.4)RootModule:- The root module is loaded by calling

platformBrowserDynamic().bootstrapModule(AppModule)

.catch(err => console.error(err));

The root module is ‘AppModule’.

4.5)RootComponent:- The Root component is ‘AppComponent’. It is loaded when root module is loaded by angular.

4.6)Template:- The AppComponent defines the template as app.component.html and the CSS Selector is app-root

Our index.html already have the app-root CSS selector defined

<body>

<app-root></app-root>

</body>

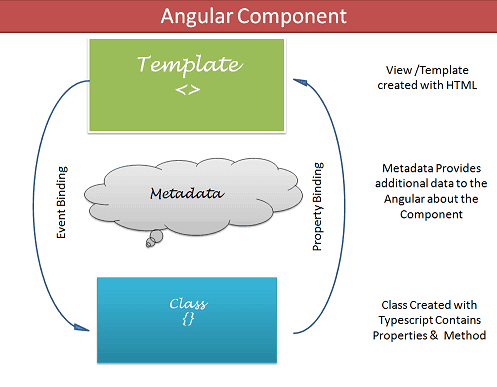
The Angular locates app-root in our index.html and renders our template between those tags.

**5. Component:** The component consists of 3 parts.

5.1. Template/view

5.2. class.

5.3. MetaData.



5.1. Template:- The Templates are nothing but HTML codes along with the Angular specific special HTML markups ( knows as the Angular Template Syntax). There are two ways you can specify the Template in Angular.

a. Inline Template

b. External Template

5.2. class:- we use TypeScript to create a class. This class contains data(properties) and logic(methods) to template/view.

Syntax:

Export class classname{

Properties;

Methods;

}

By convention we prefix the Component class with decorator(@Component) so as to easily identify them.

5.3.MetaData:-Metadata Provides additional information about the component to the Angular. We use the @Component decorator to provide the Metadata of the Component.

5.3.1) @copyDirector:- When Angular sees a class with @Component decorator, it treats the class as Component. A decorator is a function that adds metadata to class, its methods & to its properties. The Components are defined with a @Component class decorator.

Properties of @copyDirector:-

* selector:- Selector specifies the simple ‘CSS selector’. The Angular looks for the ‘CSS selector’ in the template and renders the component there.
* providers:- Providers are angular services, that our component will use them.
* directives:- The directives that our component will use them.
* Styles/styleUrls:-
  + - styles is used for inner-styles.
    - StyleUrls is used for external styles.
* template/templateUrl:- The templates can be inline (using a template) or we can use an external template (using a templateUrl). The Component can have only one template. You can either use inline template or external template and not both.

Example to component:

@Component({

  selector: '',

  templateUrl/template: '',

  styleUrls/styles: [''],

providers:’ ‘,

directives;’ ‘

})

export class AppComponent {

  properties;

methods;

}

5.4.Creating the component:- In two ways, we can create componets.

5.4.1) Manually:- The creation of the Angular component requires you to follow these steps

* Create the Component file
* Import the required external Classes/Functions
* Create the Component class and export it
* Add @Component decorator
* Add metadata to @Component decorator
* Create the Template
* Create the CSS Styles
* Register the Component in Angular Module

Drawback:- This is time taken process.

5.4.2) Using command:-

Syntax:- ng g c component\_name;

G means generate.

C means create.

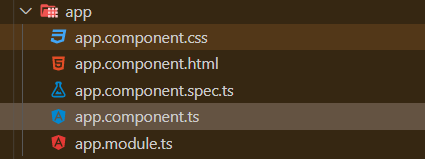
By default, this command creates the following:

* A folder named after the component
* A component file, <component-name>.component.ts
* A template file, <component-name>.component.html
* A CSS file, <component-name>.component.css
* A testing specification file, <component-name>.component.spec.ts.

After, we should modify contents in these files as per our requirements.

Example:

Before creating the component, app folder structure is



PS C:\Users\sukumar\Desktop\Ang\_projects\first-app> Get-ExecutionPolicy

PS C:\Users\sukumar\Desktop\Ang\_projects\first-app> ng g c sample

CREATE src/app/sample/sample.component.html (21 bytes)

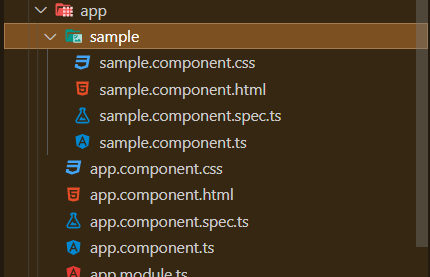
CREATE src/app/sample/sample.component.spec.ts (599 bytes)

CREATE src/app/sample/sample.component.ts (275 bytes)

CREATE src/app/sample/sample.component.css (0 bytes)

UPDATE src/app/app.module.ts (396 bytes)

PS C:\Users\sukumar\Desktop\Ang\_projects\first-app>



File name: Sample.component.ts

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

@Component({

  selector: 'app-sample',

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

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

})

export class SampleComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

Note:- **After creating component, Angular CLI assigns component name as value to declarations property in app.module.ts file automatically.**

Once component created, we have two ways to write the html and css code.

a. External Template:- Write a html code and css code in <component-name>. component.html. and css file <component-name>.component.css respectively.

Note:- Link these two file with <component-name>.compont.ts file by templateURL and styleURL properties. But this is done automatically by CLI.

b. Internal Template/Inline Template (or) styles:-

Step1:- Remove the <component-name>.component.html

Step2:- Remove the <component-name>.component.css

Step3:- open the <component-name>.component.ts. Modify the code in that file.

@Component({

  selector: '',

  template: '**Write Here HTML code**',

  styles: ['**Define css style sheets here**'],

providers:’ ‘,

directives;’ ‘

})

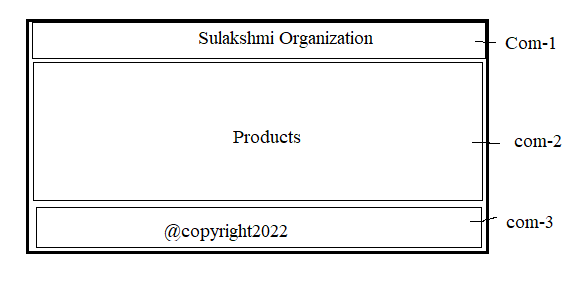
export class AppComponent {

  properties;

methods;

}

Example:- Create a below web page.



Stpe1: create a 3 components named header,body and footer.

* Ng g c header
* Ng g c body
* Ng g c footer

Step2: open the ’ header.component.html ‘file and write related html code.

<div>Sulakshmi Organization</div>

Opent the ‘header.component.css ‘ file and write related css code.

div{

    text-align: center;

    color:blue;

    background-color: orangered;

    border: 1px solid black;

    font-size: 25px;

    font-weight: bold;

    margin-bottom: 2px;

}

Step3: open the ’ body.component.html ‘file and write related html code.

<div>

     Products

</div>

Open ‘body.component.css’ file and write css code.

div{

    height:200px;

    border: 1px solid black;

    text-align: center;

    font-size: 40px;

    color:blue;

    background-color: red;

    margin-bottom: 2px;

}

Step4:

1. Remove the ‘footer.component.html’ file.
2. Remove the ‘footer.component.css’ file.
3. Open ‘footer.component.ts’ file and do modifications.

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

@Component({

  selector: 'app-footer',

  template: '<div>@copyright2022</div>',

  styles: ['div{ text-align:center;color:black;backgroud-color:white;border:2px solid black;}']

})

export class FooterComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

Step5: open the ‘app.component.html’ file and write selectors in that.

<app-header></app-header>

<app-body></app-body>

<app-footer></app-footer>

Step6: Run the project

* Go to project folder
* Ng server-> press enter.

