Environmental Setup for Angular9

1) download and install NodeJS

- To install "Angular9" we need "npm".
- "npm" stands for node packaging manager.
- "npm" is the tool present in "NodeJS".

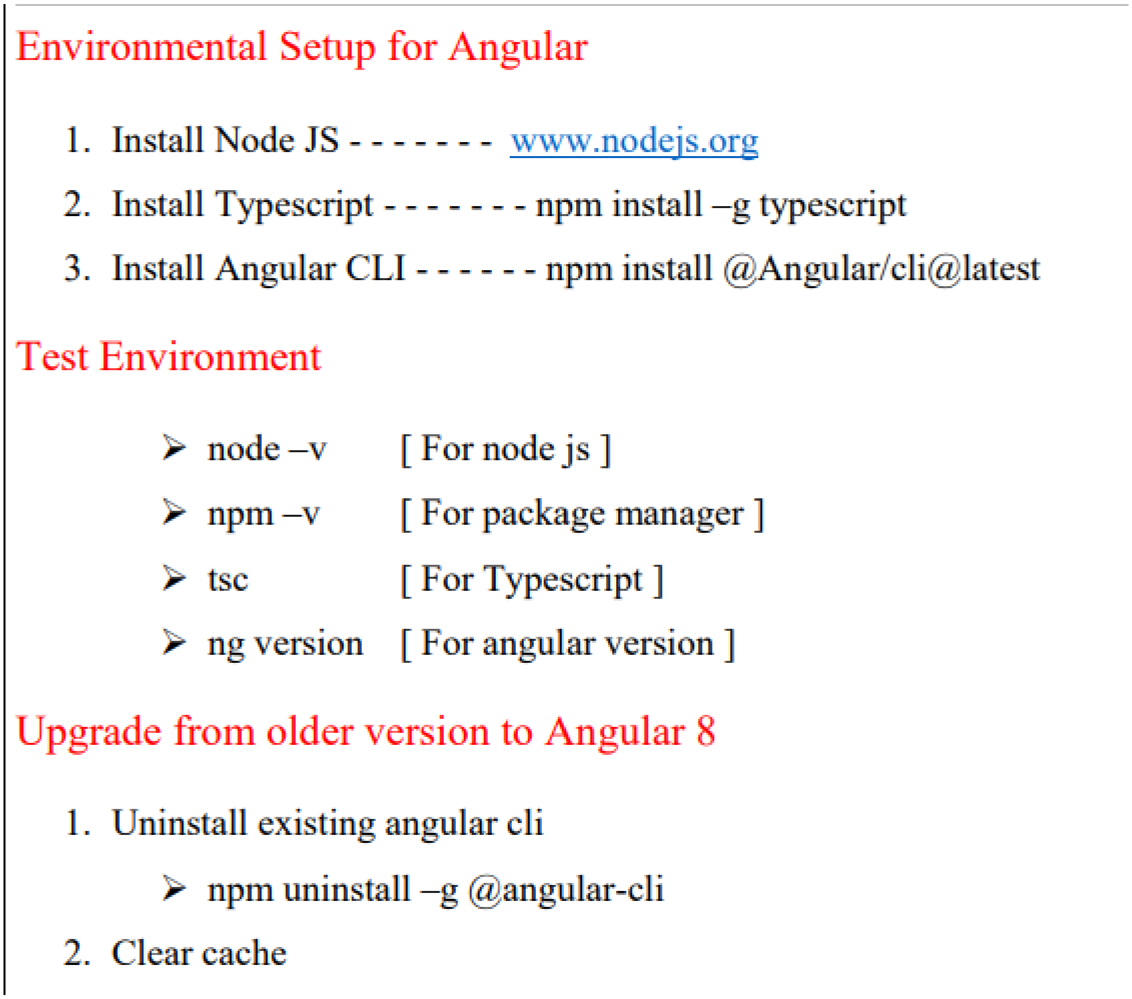
Website : https://nodejs.org/en/download/

file : node-v12.16.1-x64.msi

3) install Angular9

- we will install Angular9 by using following command.

  Command: `npm install -g @angular/cli@latest`
- "cli" stands for command line interface
- "cli" is the tool provided by google.
- "cli" tool used to build and execute the angular applications



Environmental Setup for Angular

1. Install Node JS - - - - - - - www.nodejs.org
2. Install Typescript - - - - - - - npm install –g typescript
3. Install Angular CLI - - - - - - npm install @Angular/cli@latest

Test Environment

- ➢ node –v [For node js]
- ➢ npm –v [For package manager]
- ➢ tsc [For Typescript]
- ➢ ng version [For angular version]

Upgrade from older version to Angular 8

1. Uninstall existing angular cli
   - ➢ npm uninstall –g @angular-cli
2. Clear cache

   - npm cache clear
   - npm cache verify
3. Install latest version
   - npm install –g @angular/cli@latest
4. Check the version of angular
   - ng version

Create a New Angular Project:-

1. Create a new Folder on your drive
   - “c:\angular8project”
2. Open folder in your VS code
3. Open terminal “ Ctrl+` ” (Ctrl + back tic)
4. Type the command
   - ng new shopping
5. Add routing model to project? : n
6. Which css format you want? : css

Directory Structure

2) node_modules:

- "node_modules" directory containes libraries.
- those libraries helps to execute the angular application.

3) src/app:

- this directory used to deploy the angular applications.

  Ex.

  Components , Directives , Services, Pipes

4) src/app/app.module.ts

- ✓ this file we can call registration file.
- ✓ this file also called as Module file.
- ✓ this is the Default Module in Angular Application.
- ✓ this file used to register the angular applications.
- ✓ once if we register, then only angular applications will
- ✓ be executed by angular framework

6) environments:

- in general we have 3 types of environments
    - development environment
    - production environment
    - testing environment
- what ever the required environment, we will configure
- in environments directory.

7) src/favicon.ico:

- this is the default logo of angular.

8) src/index.html:

- angular starts the execution from "index.html" file.
- "index.html" file is the landing template.
- "index.html" file is the main template in angular application.
- main template internally invokes the "main.ts" file.
- "main.ts" file internally invokes the "app.module.ts" file.
- "app.module.ts" file containes our applications registrations.
- based on registrations our applications will be executed by angular framework.

9) src/main.ts:

- this file acting as interface between main template to registration file.

(app.module.ts <==> index.html)

11) src/styles.css:

- we will define global styles here.
- what ever the styles we define here, automatically applicable to entire angular application.

13) editorconfig & .gitignore:

- these two files not related to angular applications.
- first file related to "VisualStudioCode" Configurations.
- second file related "Git" configurations.

14) angular.json:

- this file representing directory structure of angular application.
- we can customize directory structure based on application requirement by using angular.json file.
- this file used to configure the 3rd party technologies

  => jQuery

  => BootStrap

  => ReactJS

17) package.json:

- this file used to download the 3rd party libraries.
- all these libraries downloads to "node_modules" folder.

18) tsconfig.app.json:

- this file acting as controlling file for entire angular application.
- what ever the business logic written here, automatically applicable to entire angular application.

  Ex.

- removing the white spaces in entire angular applications

-overcome the data redundancy in entire angular applications.

19) tsconfig.json:

- it contain TypeScript Configurations

20) tsconfig.spec.json:

- this file is the controlling file for all unit test cases present in angular project.

Components:

- Angular is the Framework.
- Angular Framework follows the MVC Design Pattern.
    - M – Model
    - V – View
    - C – Component
- Simple TypeScript class behaves like Component.
- We Can Create more than one component in angular applications.
- Angular Applications are component based applications.
- Because of Components Code Reusability is high in Angular Compared to AngularJS.
- Component acting as Interface Between View and Service in MVC Architecture.

- we can establish the communication between server to database by using modules.
    - Ex.=> Mysql, mssql, mongodb,, firebase
- we can provide communication between service to server by using AJAX Calls (Observables).
- we can establish communication between component to service by using dependency injection.
- the communication between view to component called as two way data binding.

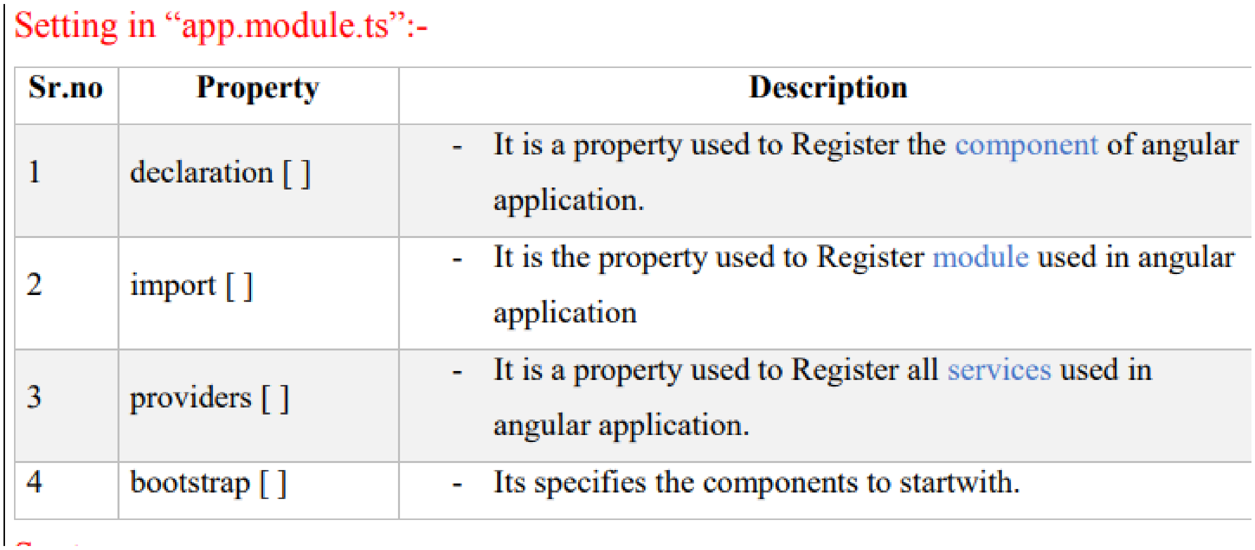
- "first.component.ts" file used to create the component.
- "first.component.html" file used to display the component output.
- "first.component.html" also called as external template of component.
- in general we will register our applications (component) in app.module.ts file.
- index.html file is the main template.

First.component.ts:

- Component is predefined class available in @angular/core package
- Component class used to convert the TypeScript Standards to HTML Standards

- we will use Component class by using "@"
- Using the predefined class by using "@" symbol called as Decorator.
- Decorators are used to define th METADATA
- Data About Particular Component Called as METADATA
- Component Class constructor takes the JSON Object as Argument.
- "selector" is the json key used to define the custom HTML Element.
- we will call custom HTML Element in "index.html" file.
- "templateUrl" is the json key used to define the external template to Component.
- in general we will use external templates to display components data.
- export is the keyword in TypeScript

- export keyword used to export the components,services,directives,pipes,....
- anyone can import the exported members in angular applications



Setting in “app.module.ts”:-

Sr.no	Property	Description
1	declaration [ ]	- It is a property used to Register the component of angular application.
2	import [ ]	- It is the property used to Register module used in angular application
3	providers [ ]	- It is a property used to Register all services used in angular application.
4	bootstrap [ ]	- Its specifies the components to startwith.