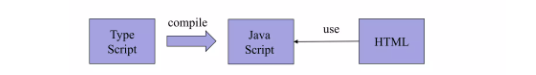
Contents

* Introduction & Overview
* Installation and Configuration
* Creating First Application
* Creating Components and Services
* Include Bootstrap
* Make a Build



**Introduction & Overview**

* **Angular History**
  1. Angular JS (version 1.0) the very first version was developed/Invented by two developers names Misko Hevery and Adam Abrons in 2009 as a side project.
  2. Google took this side project rename it Angular JS and built a team to create and maintain it within Google. It started to develop the entire framework afterward Google released the first version 1.0 of AngularJS in May 2011. From version 2.0 Google renames the project from Angular JS to Angular.
  3. First angular version is **2.0**. The initial release date is 14th Sept 2016.
  4. Stable release version **15.2.1**. The release date is 1st march 2023.
* **What is Angular?**
  1. Angular is open source.
  2. Angular is a JavaScript Framework written in TypeScript.
  3. Angular primary purpose is to develop single-page applications.
  4. TypeScript is a syntactic superset of JavaScript that adds **static typing**.



**Installation and Configuration**

* **Installation- configuration**
  1. As we know already, Angular is written in TypeScript. We need Node (JavaScript Runtime Environment) and NPM (Node Package Manager) to compile the files into JavaScript after that, we can deploy our application.
  2. We can check it using the below command:

**node --version**

You could see the version of the node. It is shown below.

**V14.20.0**

If **Node** is not installed, you can download and install it by visiting the following Refer to **<https://nodejs.org/en/>** to install Node.js.

* 1. NPM is used to install Angular CLI (Command line interface). Once Node.js is installed, NPM is also installed. If you want to verify it, type the below command

**npm –version**

You could see the version below:

**6.14.4**

Very useful in quickly creating an Angular project. Install the Angular CLI npm package globally using the following. Let’s install Angular CLI using npm as follows:

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

To verify Angular is properly installed on your machine, type the below command:

**ng --version**

You could see the following response:

**Node: 14.2.0**

**OS: win32 x64**

**Angular:**

**...**

**Package Version**

**------------------------------------------------------**

**@angular-devkit/architect 0.803.26**

**@angular-devkit/core 8.3.26**

**@angular-devkit/schematics 8.3.26**

**@schematics/angular 8.3.26**

**@schematics/update 0.803.26**

**Creating First Application**

* **How to create the angular application?**

1. Let us check whether the Angular Framework is installed in our system and the version of the installed Angular version using the below command:

**ng –version**

1. **ng** is the CLI application used to create, manage and run Angular Applications. It is written in JavaScript and runs in a NodeJS environment.
   * The result will show the details of the Angular version as specified below:

**Node: 14.2.0**

**OS: win32 x64**

**Angular:**

**...**

**Package Version**

**------------------------------------------------------**

**@angular-devkit/architect 0.803.26**

**@angular-devkit/core 8.3.26**

**@angular-devkit/schematics 8.3.26**

**@schematics/angular 8.3.26**

**@schematics/update 0.803.26**

1. **Angular CLI** helps in creating a project very easily. In order to create the project, use the following command.

**ng new hello-world-application**

* **new** is one of the commands of the **ng** CLI application. It will be used to create new applications. Regarding the routing question as mentioned below, specify **No**.

**? Would you like to add Angular routing? No**

* Select an option for a style like this,

**? Which stylesheet format would you like to use?**

* **CSS**
* Let us move into our newly created application folder name: **hello-world-application.**

**cd Hello-World-Application**

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

**| favicon.ico**

**| index.html**

**| main.ts**

**| polyfills.ts**

**| style.css**

**|**

**+--app**

**| app.component.css**

**| app.component.html**

**| app.component.spec.ts**

**| app.component.ts**

**| app.module.ts**

**|**

**+--assets**

**| .gitkeep**

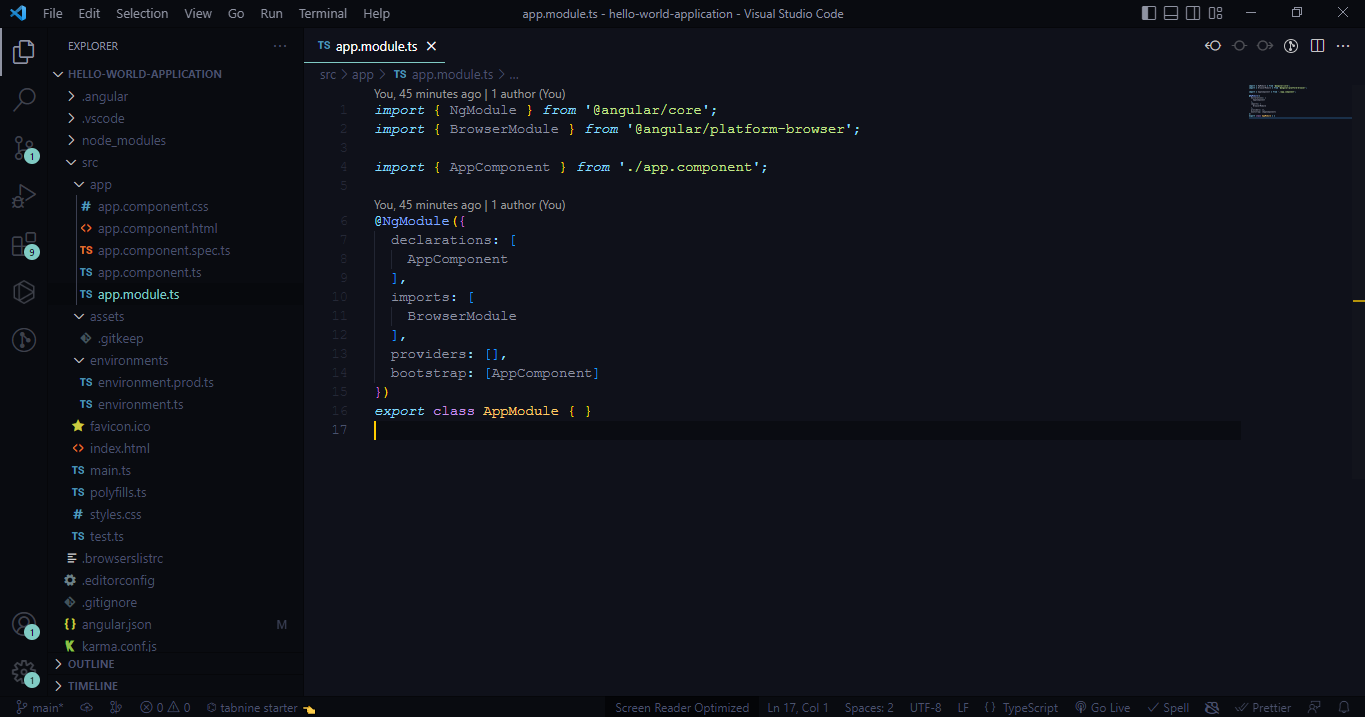
**|**

**+---environments**

**| environment.prod.ts**

**| environment.ts**

* + **favicon.ico** and **assets** are the application’s icon and the application’s root assets folder.
  + **polyfills.ts** contains standard code useful for browser compatibility.
  + **environments** folder will have the application’s settings. It includes production and development setup.
  + **main.ts** file contains the startup code.
  + **index.html** is the application base HTML code.
  + **style.css** is the base CSS code.
  + **app folder** contains the Angular application code.
    1. **app.component.html** is the main component of an angular app, and all other components are usually present within this component.
    2. **app.module.ts** isthe file where all the components, providers, and modules are defined. Without defining them here, they can’t be used elsewhere in the code **(METADATA)**.
       1. **declarations −** In declarations, the reference to the components is stored. The App component is the default component that is created whenever a new project is initiated.
       2. **imports −** This will have the modules imported as shown above. At present, Browser Module is part of the imports which is imported from @angular/platform-browser.
       3. **providers −** This will have reference to the services created. The service will be discussed in a subsequent chapter.
       4. **bootstrap −** This has reference to the default component created, i.e., AppComponent.



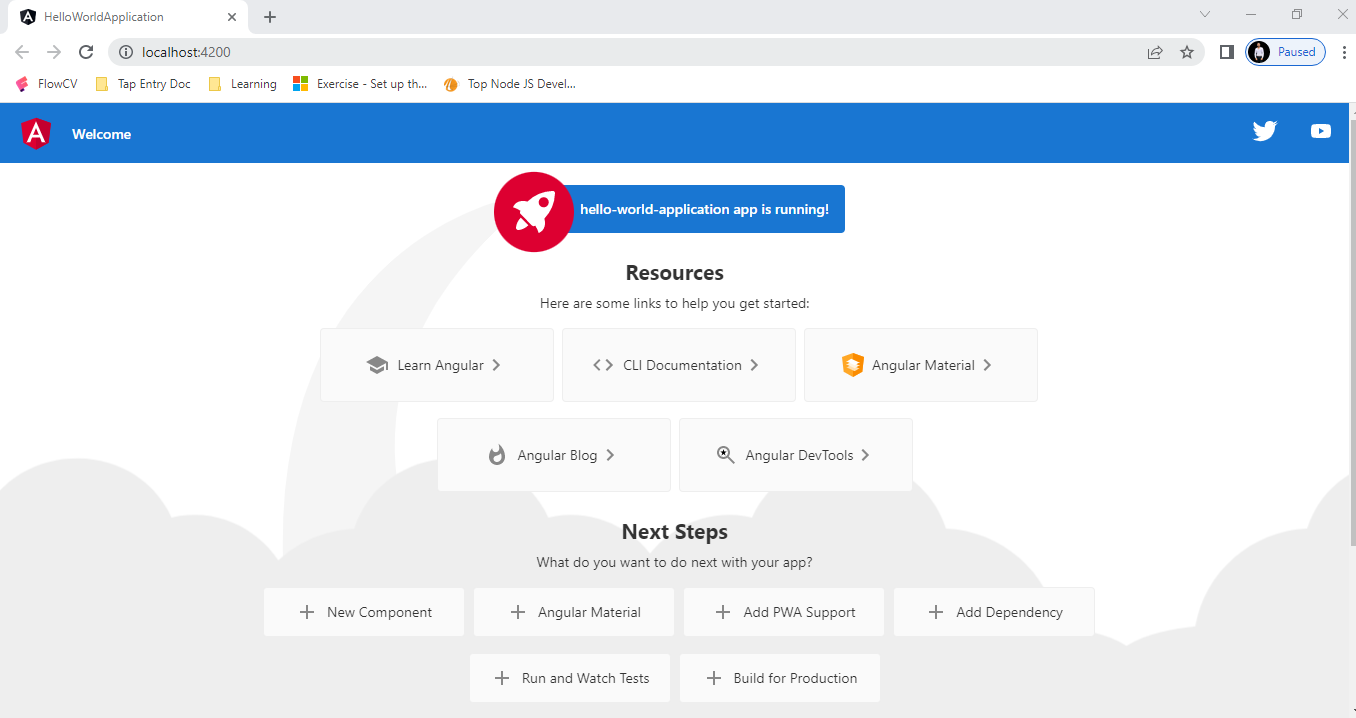
1. Let us start the application using below command:

**ng serve**

**OR**

**ng serve -o**

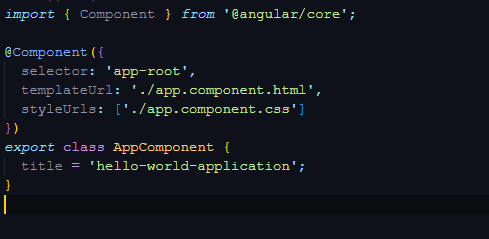
* + Here, **serve** is the sub-command used to compile and run the Angular application using a local development web server. **ng server** will start a development web server and serves the application under port, 4200.
  + Let us fire up a browser and open **<http://localhost:4200>**. The browser will show the application as shown below:



**Creating Components & Services**

* **Components**

1. The core of the Angular framework architecture is **Angular Component.** Angular Component is the building block of every Angular application. Every angular application is made up of one more **Angular Component.** It is basically a plain JavaScript / Typescript class along with an HTML template and an associated name.
   * The HTML template can access the data from its corresponding JavaScript / Typescript class. The component’s HTML template may include other components using its selector’s value (name). The Angular Component may have optional CSS Styles associated with it and the HTML template may access the CSS Styles as well.
2. Components accept the default three metadata objects.
   * **selector, templateUrl, styleUrls.**
   * In addition to like, animation, providers, changeDetection.



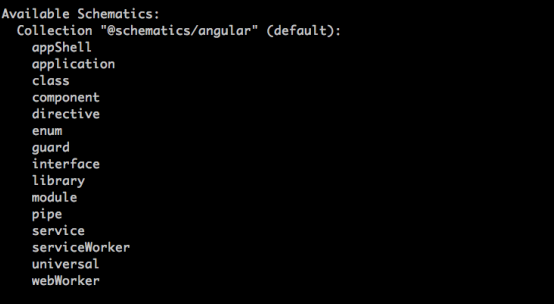
1. **Generate command** It is used to generate or modify files based on a schematic. Type the below command inside your angular project:

**ng generate**

* + Or, you can simply type generate as g. You can also use the below syntax:

**ng g**

* + **It will list out the available schematics:**



1. To create a component in angular use the below syntax:

**ng g c <component-name>**

1. For example, if the user wants to create a **my-profile** component then use the below code:

**ng g c my-profile**

1. After using this command, you could see the below response:

**CREATE src/app/my-profile/my-profile.component.css (0 bytes)**

**CREATE src/app/ my-profile /my-profile.component.css (22 bytes)**

**CREATE src/app/ my-profile /my-profile.component.css (635 bytes)**

**CREATE src/app/ my-profile /my-profile.component.css (274 bytes)**

**UPDATE src/app/app.module.ts (1201 bytes)**

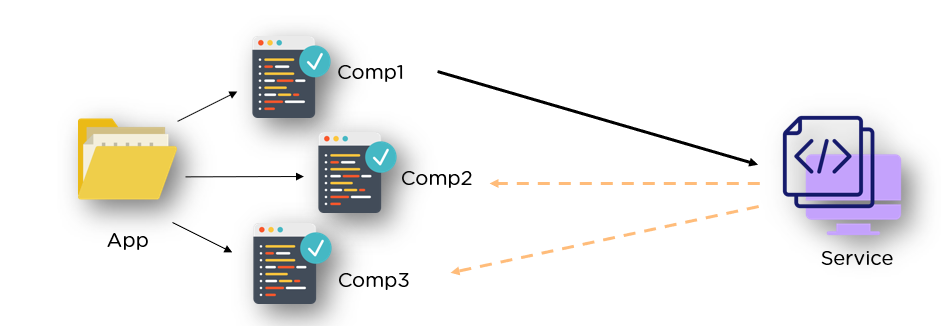
* **Services**

1. Service contains business logic and data, functions, shared by multiple Components.
2. In general, services communicate with Rest Web API and perform CRUD operations.
3. The main purpose of the service is **reusability**.
4. A service can be creating by following CLI command:

**ng generate c <service-name>**

**or**

**ng g c <service-name>**

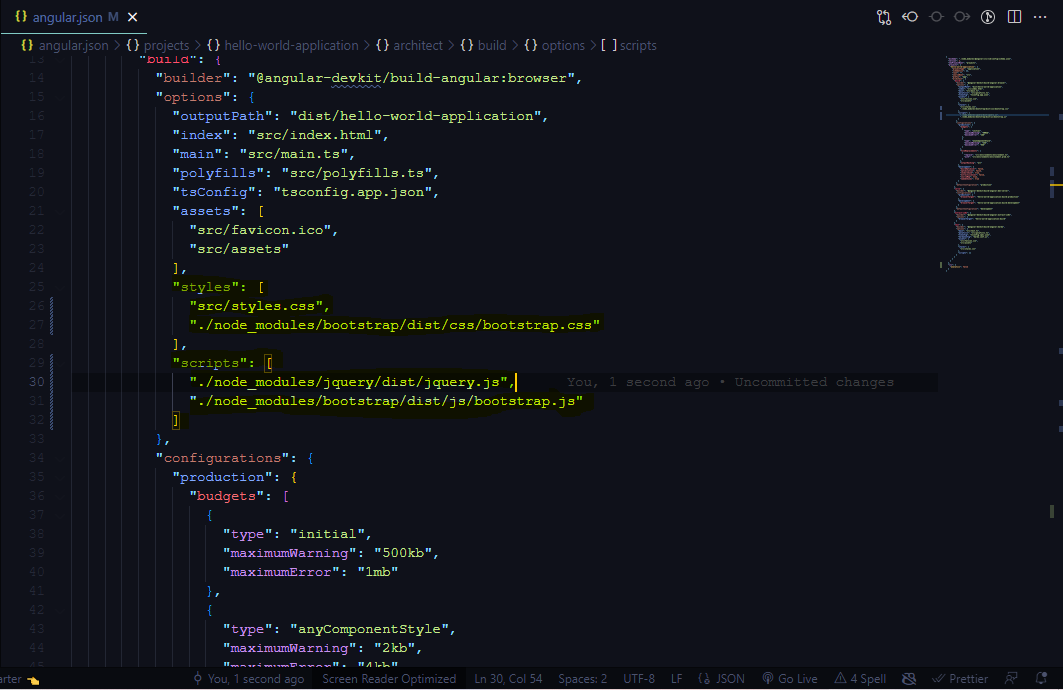


**Include bootstrap**

* **How to add bootstrap in angular?**
  1. It enables you to load the CSS and JavaScript files remotely from its servers.
  2. Navigate your application folder in Command Prompt then install **bootstrap** and **jQuery** library using below commands:

**npm install –save bootstrap jquery**

* 1. Option **angular. json** and set bootstrap and jQuery library path.



**Make a Build**

* **DIST**
  1. This folder is where the build files are present. TypeScript is basically converted to JavaScript and the resulting files are stored here after bundling and minification. It is therefore necessary to convert TypeScript to JavaScript before deploying the code. To see this folder, you can type the following in the command prompt.

**ng build**

* 1. If you want to create a build for particular environments for that following commands.

**ng build --dev | --stag | --prod**

