**Chapter: 1.** **Angular 10 Framework Briefing…**

**Objective:** The angular 10 platforms can help you build web applications much more easily. It’s a complete framework. In this chapter we will study the:

* Introduction to Angular10
* Environment Setup
* Angular Version Enhancement "Angular-2" to "Angular-10"
* Angular CLI Command
* Creating First Angular Application
* The execution process
* Folder & File Structure of application.
* Building blocks of Angular

## Introduction to Angular 10

Angular 10 happens to be the latest version of the Angular framework. Angular has been developed by **Google** and is used for making **single page web applications**. Does not this single page application mean it has one page?

This is a new concept and you can take it as how good it will be if the whole code is downloaded on the user’s computer. That is what we know as a single page web application, and currently, this is possible through JavaScript frameworks like Angular, React and Vue.js, etc.

1. Google takes care of the Angular development and is supported by some individuals and corporations who come up with quite a major list of changes that they encounter during the single page application development.
2. The main objective of the Angular framework is to simplify the development-related and the testing of application through it for the client-side model view controller (**MVC**) as well as the **MVVM** model **View Model** architecture with the help of the components that are being used for the rich internet application.
3. Angular 2 came in 2016, and hence it is not that old. Angular 10 was released on June 24th, 2020.

### Prerequisites

For using the Angular framework, you need to be familiar with the below:

* Typescript
* HTML
* CSS

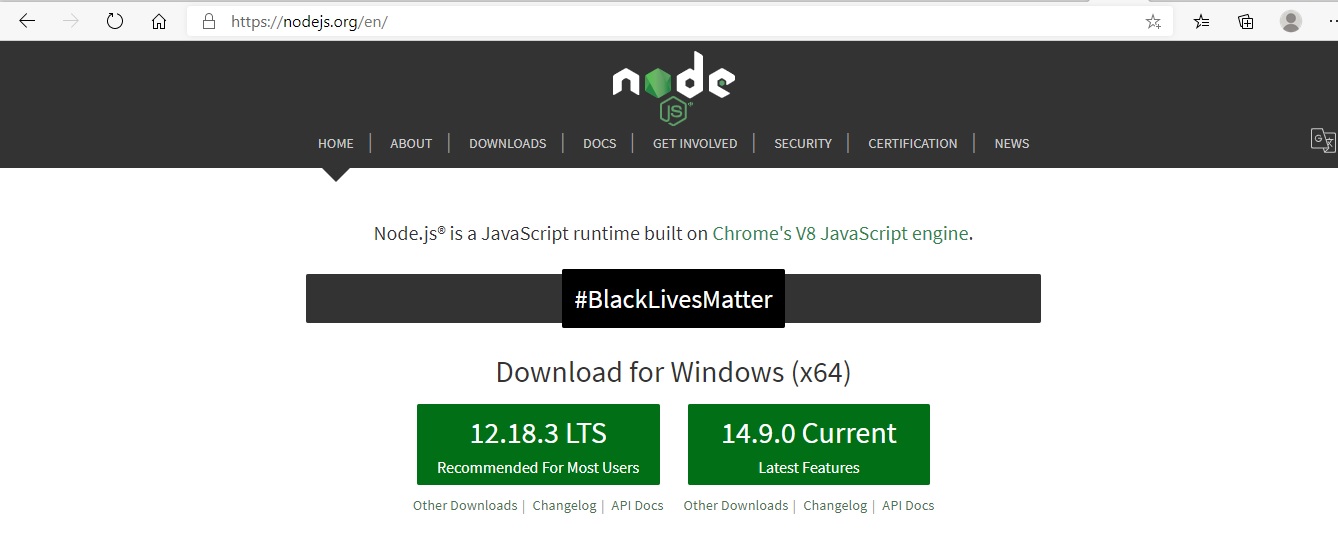
However, you can still make use of JavaScript, it is optional.

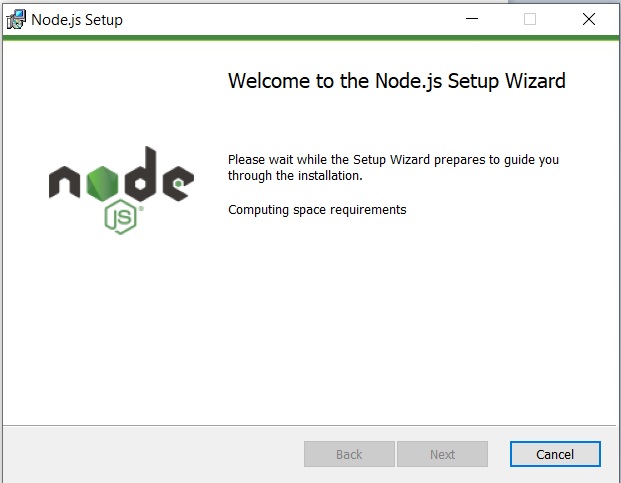
## Environment Setup

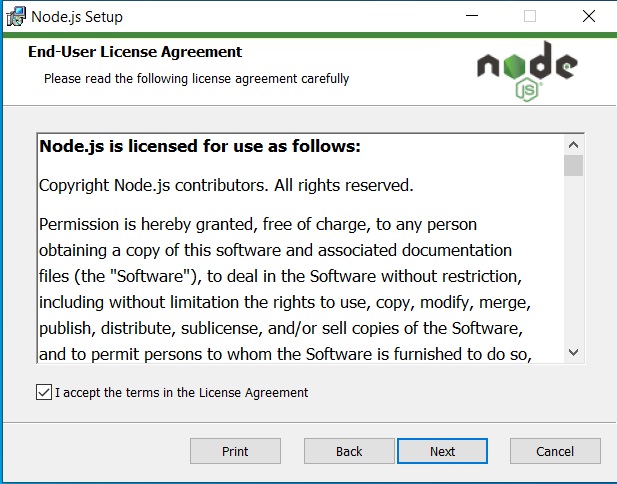
For installing the Angular locally, you need to follow the below steps:

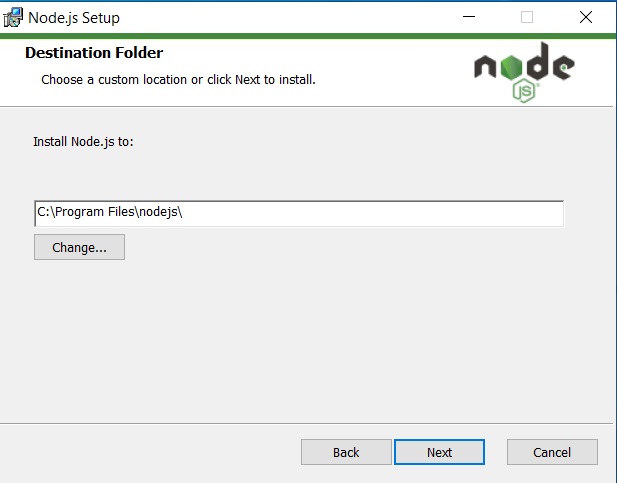
### Node.js

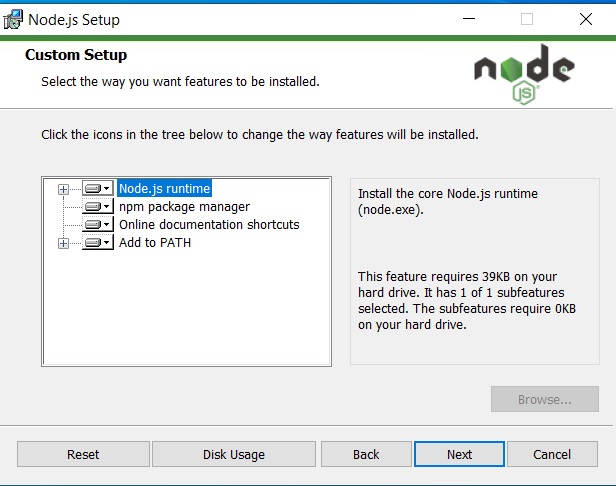
1. The Angular needs the present, active LTS, or rather maintenance LTS Node.js version.
2. For installing the node.js, you need to download the node.js from **https://**[**www.nodejs.org**](http://www.nodejs.org) and run the exe file and follow the instruction as will come in front of you. Go on clicking next, and you do not need to think of any other option.

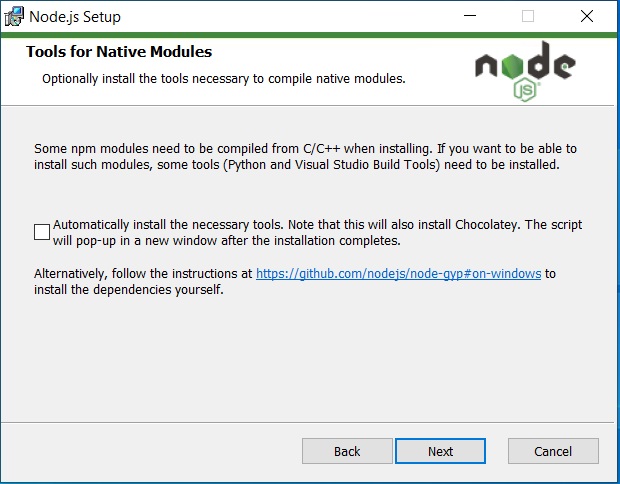
****

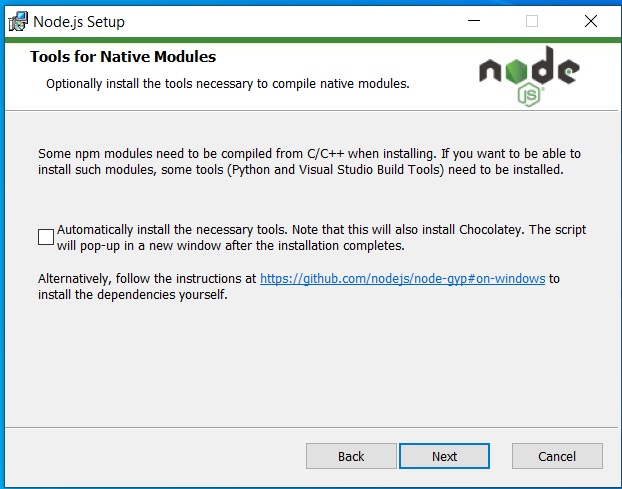
****

****

****

****

****

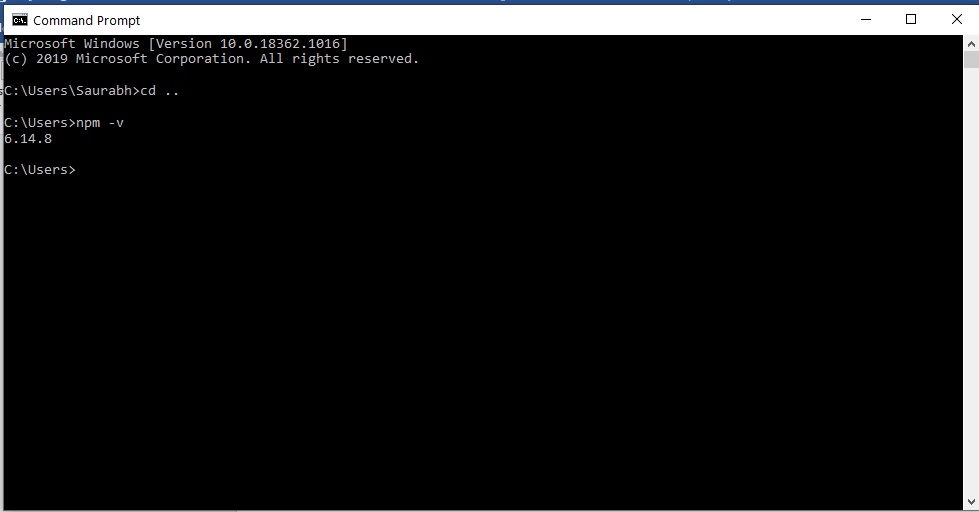
****

1. After installation to check the version we use “node -v” from the command prompt like.



### NPM:

1. The Angular framework, CLI, as well as the Angular apps they all depend upon the npm packages for numerous features and functions.
2. For downloading as well as installing the npm packages, you require the npm package manager. And for checking that the npm client is installed on your computer, you need to run “npm –v” from the terminal window as shown in the below.

****

### Installing Typescript: And a Brief Introduction:

Typescript is a typed version of JS6. It has been developed by Microsoft. Through it, we can catch almost all the errors at the development time itself. Hence there are fewer chances of errors if we use typescript. Hence a lot of time is saved. For installing Typescript use the following command:



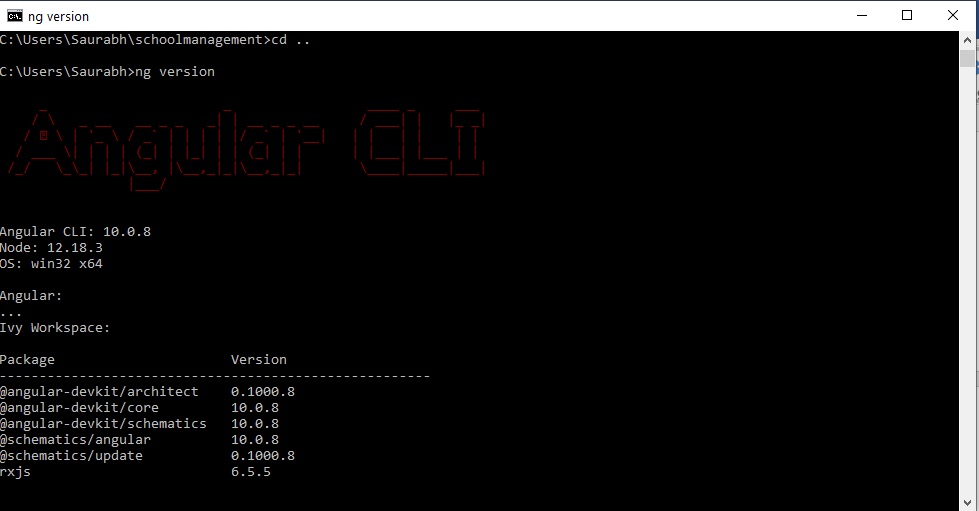
### Installing the Angular CLI

1. You need to make use of the Angular CLI for creating the projects, generating the apps and the library code, as well as performing a variety of development tasks like testing, deployment, and bundling.
2. For installing the Angular CLI, you need to open the terminal windows, deploying, and bundling.

For installing the Angular CLI, you just require to open the terminal window and finally run the below command:

****

**To check the version command**

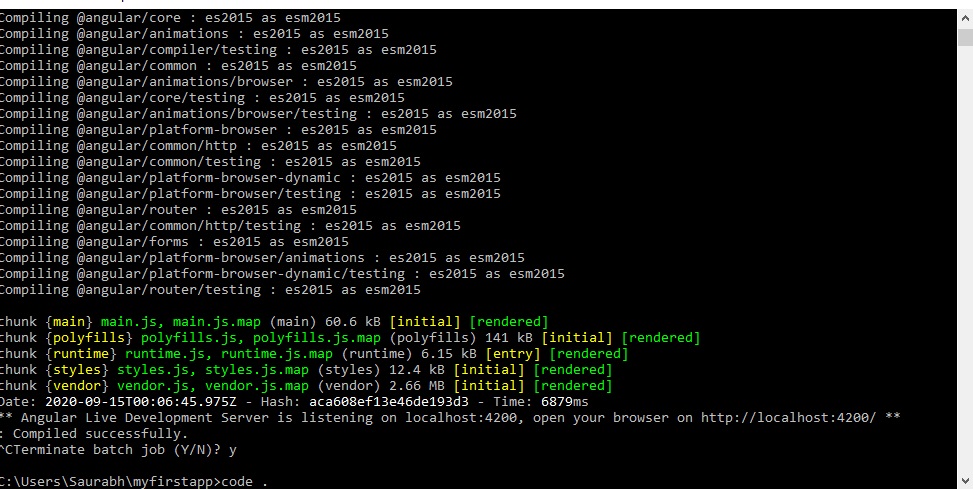


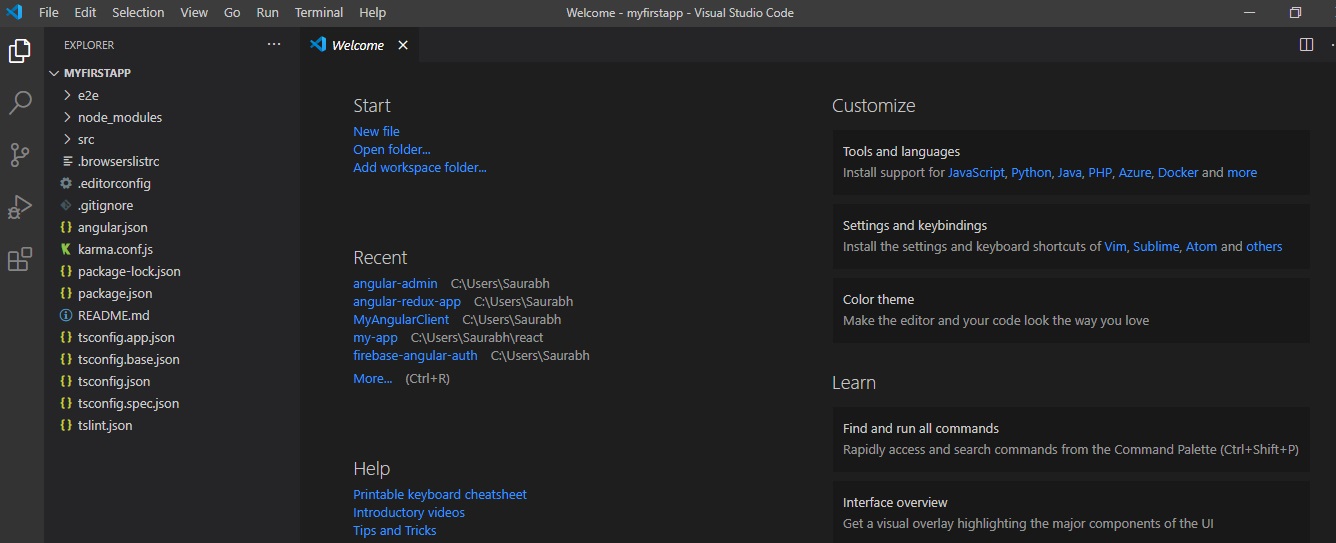
### **Visual Studio code**

The visual studio code happens to be the code editor which is supported by Microsoft and it's now more popular than any other editor. And we will be using this editor as the VS code supports the typescript perfectly with all the best features.

To enter vs code. From inside the application folder type **code.** We will see in a while how to create the app. First, create the app, and then move inside the app folder. Now type

****

****

****

## Angular Version Enhancement "Angular-2" to "Angular-10"

|  |  |  |
| --- | --- | --- |
| Angular version | Released Year | Enhancement |
| Angular 2 | September 14, 2016 | 1. This was the complete rewrite from the team that developed the AngularJS. And the team now made use of the TypeScript. 2. Angular 1.x didn’t support mobile, though the Angular 2 was fully mobile-oriented. |
| Angular 3 | Never | 1. And you will never find the Angular 3. 2. The main reason for this is the fact that there was a version mismatch between @angular/core, @angular/compiler, and the router library. 3. You will also find that in the Angular 2 version for the router as well as the core we have the below-mentioned details: 4. Core v2.3.0 5. Compiler-cli v2.3.0 6. Compiler v2.3.0 7. Router v2.3.0 8. HTTP v 3.3.0   There is a version mismatch between core and compiler above, and hence no version 3. |
| Angular 4 | March 23, 2017 | 1. The Angular 4 was released in 2017, and the month of March. 2. And this particular release happens to be backward compatible with the 2.x.x for most of the apps. And you will not find a big deal of changes in Angular 4 in comparison to Angular 2. 3. Though the Angular 4 is not the complete copy of the Angular 2.l However, the angular team emphasized in the development of the angular app at much better great speed, as well as it was more compact than the previous version. |
| Angular 5 | November 1, 2017 | 1. And this was released in 2017 in November. 2. Let’s now consider some of the most significant changes in version 5. 3. Build Optimizer: This was the production build that was being developed with the Angular CLI now applying by default the build optimizer 4. Universal state transfer API and the support for DOM 5. Improvement in compiler 6. Internationalized Date, Number, and currency pipes 7. HttpClient;@angular/Http was deprecated in this version. And it was being replaced by @angular/common/Http 8. Library: not was now brought into effect instead of map (res=>res.json ()) calls, which are now no longer required. And the angular CLI 1.5 is now used to generate the v5 projects by default. 9. And the Angular form was updated with Submit/blur. 10. The RxJS 5.5 was also being added. 11. Several new lifecycle events were added too like Guardschecked. |
| Angular 6 | May 4, 2018 | 1. Angular 6.0.0 came in May 2018. And this came with Angular CLI6 and Material 6. 2. Beginning from this version the angular team synchronized the framework packages (core, common, compiler,etc), 3. Added were two more angular cli commands: 4. ngupdate and ng add 5. Angular Elements: 6. Component Development Kit (CDK) 7. Angular Material starter components 8. CLI Workspaces 9. Schematics 10. Library support 11. Tree Shakable Improvements 12. RxJS v6 |
| Angular 7 | October 18, 2018 | 1. This was released in October in 2018, which was released at the same time as Angular CLI7 as well as Angular Material 7. 2. CLI Prompts: The CLI is now going to prompt the users while running the commands such as: 3. Ng new or ngadd@angular/material 4. And these were added to discover the built-in features that were added in Schematics such as routing or SCSS support CLI prompts, and the bundle budgets in the CLI. Also added was Angular Material & CDK, Vir 5. Angular Material & CDK: 6. Virtual Scrolling 7. Drag and Drop 8. The Angular Elements project came with Content Projection support 9. Updates for dependency: 10. TypeScript 3.1 11. RxJS 6.3  * Node 10 was now supported. |
| Angular 8 | May 28th, 2019 | 1. This came in May in 2019 and was released at the same time as Angular CLI 8 and Angular Material 8. 2. Important features that came with Angular 8, Angular Material 8, and Angular CLI8 were as below: 3. Differential Loading: 4. By default, differential loading meant the browser was free to select in between the legacy and the modern JavaScript based on the capabilities. 5. Route configuration now came up with the dynamic imports 6. Builder APIs in the CLI: It’s an awesome feature and with the help of this, we can customize angular CLI commands such as: 7. ng build, ng test, and ng run 8. The workspace APIs in the CLI 9. Support of Web Workers 10. With Angular 8.3.0 we were provided with new UX for the initial app that has been created with the help of ng new 11. Another command was: ng deploy, and this was added in Angular CLI 8.3.0 |
| Angular 9 | February 6, 2020 | 1. Now we have Ivy renderer by default, and not at the op-in level. If you will read history in more details you will find that for using it you need to move to the tsconfig.json file and now add below it the following line: 2. “angularCompilerOptions”: “enable”: true} 3. And in Angular 9 the default Angular compiler is Ivy renderer, and you are not required to accomplish anything new to the tsconfig file for taking advantage of the Ivy. 4. Also, the URL definition has become quite consistent now. The style URL is being examined in the same manner as the template URLs. |
| Angular 10 | June 24,2020 | 1. New Date Range Picker 2. Notices about CommonJS imports 3. Optional Stricter Sceneries 4. Charge Up to Date with the Network 5. New Default Browser Sketch 6. Deprecations and Eliminations |

## Angular Commands used in CLI

The CLI tools from Angular is commonly known as an interface tool for command-line and is being used for initializing, developing, scaffolding, and maintaining Angular apps. Also one can use the below-mentioned commands from the command prompt or via an interactive UI indirectly like the Angular Console.

|  |  |  |
| --- | --- | --- |
| S.No | Command | Description |
| 1 | ng add  ng a | It is being used for adding the support for certain external libraries to the project. |
| 2 | ng build  ng b | This compiles the angular app inside the output directory called dist/ at the provided path. And it must be executed from inside the workspace. |
| 3 | ng config  ng c | This retrieves or configures the Angular values inside the angular file for certain workspace. |
| 4 | ng doc  ng d | If you want to open the official Angular documentation inside the browser or search for a certain keyword, you can use the above-mentioned command. |
| 5 | ng e2e  ng e2e | This builds as well as serves an Angular app, and then it runs end to end test with the help of the Protractor. |
| 6 | ng generate  ng g | It modifies and generates the files based on the Schematic. |
| 7 | ng help  ng h | This provides the entire available command list and their short description. |
| 8 | ng lint  ng l | This is being used for running the linting tools on the Angular app code in a certain project folder. |
| 9 | ng new  ng n | This is going to generate a novel workspace and an initial app. |
| 10 | ng run  ng r | It runs an architect target with the defined for the project the optional custom builder. |
| 11 | ng serve  ng s | This builds and serves the app, and rebuilds with all the changes made in the file. |
| 12 | ng test  ng t | This runs the unit test on the project. |
| 13 | ng update  ng u | This updates the application and its dependencies. Have a look at the https://update.angular.io/ |
| 14 | ng version  ng v | It outputs the Angular CLI version. |
| 15 | ng xi18n | This extracts the i18n messages from the source code. |

## Creating first Angular Application:

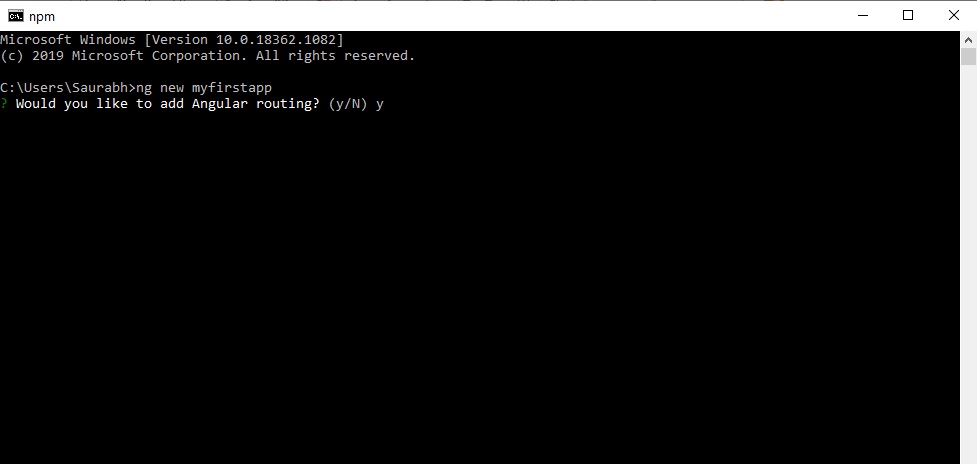
We are going to create an Angular App, explain each step, and examine the folder structure of the Angular app code.

**Step1:** Open the command prompt, and we have already installed angular cli and all prerequisites.

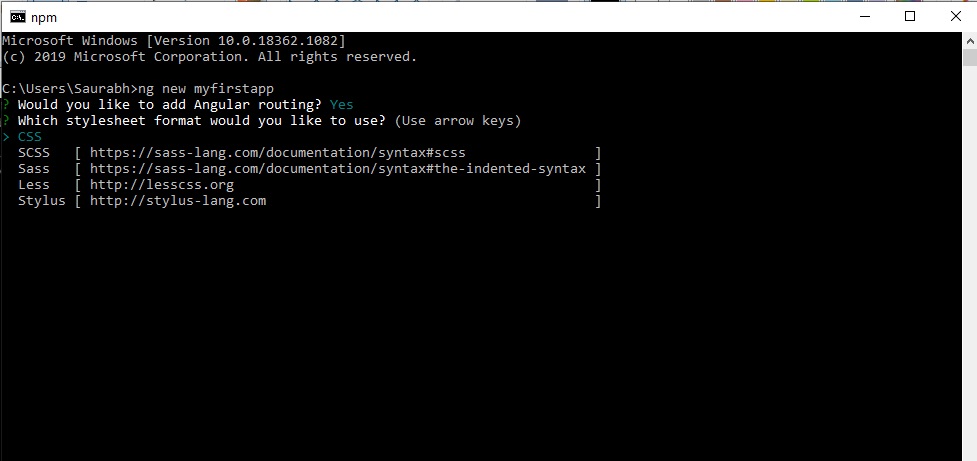
**Step 2:** type **ng new myfirstapp**



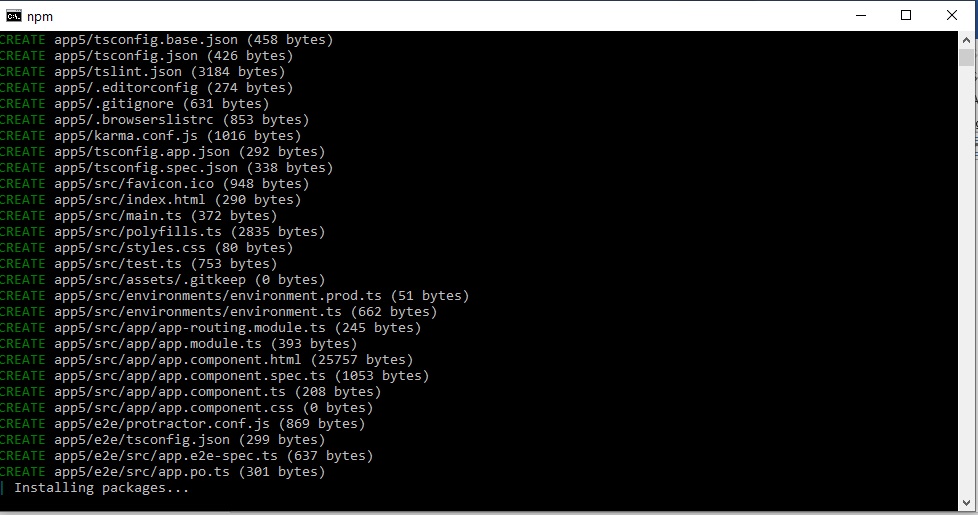
**Step 3:** Say yes to routing.

****

**Step 4:** Choose CSS for styling.

****

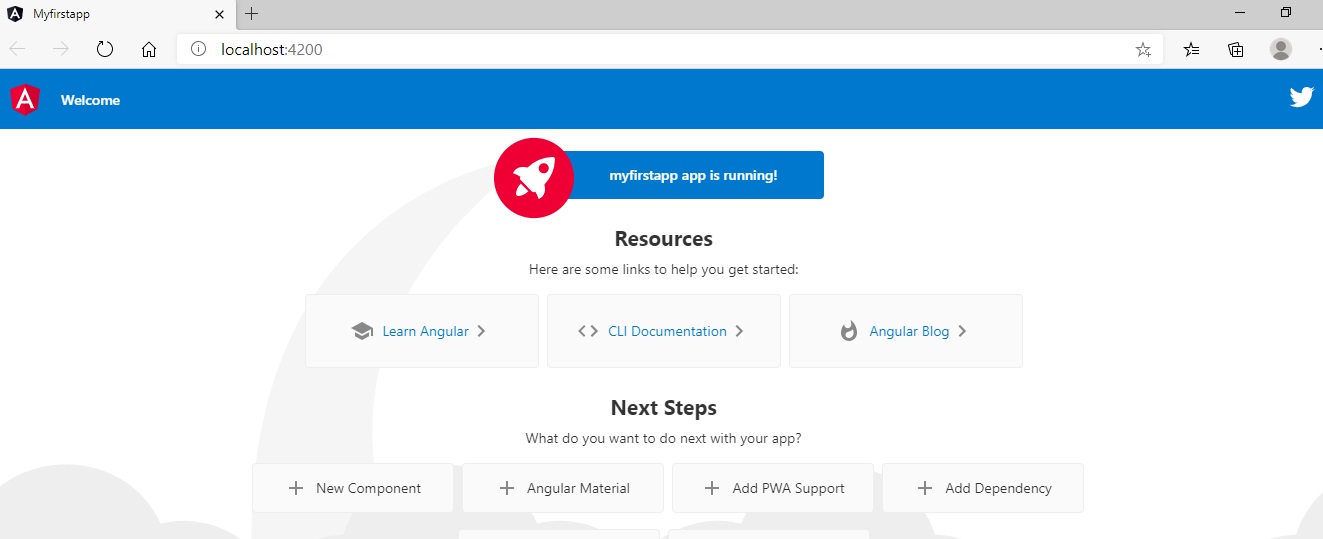
**Step 5:** Wait for all packages and hence app to install

****

**Step 6:** Go inside the app folder **cd myfirstapp**

**Step 7:** start app **ng serve- -o**

This will open the app in the defaultbrowser.

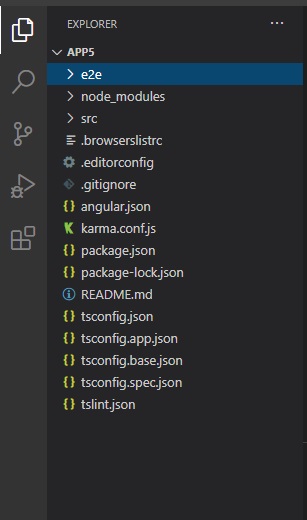
****

You have done it.

## Angular 10 app folder structure:

You as a programmer require to make some changes in the **app.component.ts.** Changes also need to be done in the **app.component.html.** And the port 4200 is being used by the Angular cli for compiling the app as it is the default port.

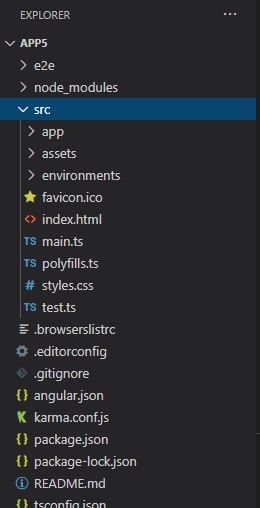
The angular10-app/ folder has the following file structure –



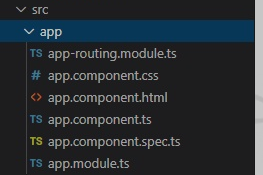
The angular 10-app/folder comes with the following folder structure—

|  |  |  |
| --- | --- | --- |
| S.No | Folder Name | Description |
| 1 | e2e/- | It’s the test folder and two e stand for end to end, which is meant for the integration testing and for ensuring that the app is running fine. |
| 2 | src/- | It’s this folder in which the real work is executed. Within this you will find the -app/folder which is up with below file structure- |
| 3 | Node\_modules/- | In this lies node package module. And you can watch the packages that are available for viewing within this folder. |
| 4 | Angular.json: | This has the project name, and the cli version, and so on. |
| 5 | .editorconfig – | is the editor config file. |
| 6 | angular.json − | It has the project name, version of cli, etc. |
| 7 | .editorconfig − | This is the config file for the editor. |
|  | .gitignore- | This file must be committed in the repository for ignoring the rules sharing with various other users cloning the repository. |
|  | Package.json- | This gives us the details that which of the libraries are going to be installed inside the node modules as you run the npm install. In the condition when you want to append more libraries you can add all of them here, and then you can run the npm install command to complete the proceeding. |
|  | tsconfig.json – | This contains the compiler options which are being needed during the compilation. |
|  | tslint.json – | this happens to be the config file and contains the rules which are being required while compiling. |
|  | Assets- | The images and the js files are being stored in this folder. |
|  | Environment- | This folder contains the details for the development of the production environment. And there are two files inside this. They are environment.prod.tsenvironment.ts |
|  | Favicon.ico- | This is the file that is generally found in the website’s root directory. |
|  | index.html- | This file is displayed in the browser. The body of the index, HTML has the <app-root><app-root>, and this happens to be the selector that is being used in the app.component.tsfile and is going to display the details that come from the app.component.html file. |
|  | main.ts- | The main.ts happens to be the file where we will be starting the project development work. It begins by importing the basic module that we require. And currently, if you find the angular/core.app.module. angular//’platform-browser-dynamic and the environment is being imported by default during the angular cli installation as well as the project setup. It comes with the parent module reference AppModule. Therefore, as you execute it in the browser, the file called is the index.html. And this internally refers to main.ts that call the parent module and that is the AppModule. And as the AppModule is being called, it sends a call for the app.module.ts that further calls the AppComponent based on the bootstrap as shown below: |
|  | Bootstrap(AppComponent)- | And in the app.component.ts you will find the selector app-root that is also mentioned in the index.html and this displays the app.component.html content. And the index.html will be displayed in the browser. |
|  | polyfill.ts- | for backward compatibility. |
|  | styles.css- | the style files |
|  | test.ts- | the unit test cases meant for project testing |
|  | tsconfig.app.json- | This is handy during the compilation; it contains the config details which are required for running the application. |
|  | tsconfig.spec.json- | This keeps track of the testing details. |
|  | typings.d.ts- | for Typescript definition management. |

The **src/folder** is the main folder, and it internally comes with a distinct file structure.



App: This has the files which are being listed here.



And these files are being installed with the help of the angular cli by default.

#### app.module.ts

The module is being created from the Angular CLI, and comprise of the following:

**@NgModule** is being imported from the **@angular/core**, and the object of it has the below properties and will learn about angular modules in upcoming chapters.

|  |  |  |
| --- | --- | --- |
| SNo. | Code Part | Description |
| 1. | Declarations- | In this, we have to the references to the components. And the App component happens to be the default component which is created at the time a new project is a setup. |
| 2. | Imports- | This contains the modules that are imported. Like the **Browser Module** is the part of the import that is imported from the @angular/platform-browser. Also, you will find a routing module known as **AppRoutingModule**. |
| 3. | **Providers**: - | This has references to the services which are being created. We will be discussing this later in detail. |
| 4. | Bootstrap- | This contains the reference to the component that is created by default. And that is the **AppComponent**. |

### App.Component:

**The component has the following files:**

|  |  |  |
| --- | --- | --- |
| S.no | File | Description |
| 1. | app.component.css. - | You can develop your CSS over here. |
| 2. | app.component.html | The HTML code is written in this file.  It’s the HTML code by default and this is made available at the time of project creation. |
| 3. | **app.component.spec.s** | These are being automatically generated and are meant for the unit testing for the source component. |
| 4. | **app.component.ts** | The component class is being defined here. And you can perform the processing of the HTML structure in the, ts file. This process is going to cover all like database connectivity, routing, interacting with various other components, and services, etc. |

### Bootstrapping Angular App, Component, and AppModule

**Execution Diagram**

Main.ts

App.module.ts

Route

Component1

Component n

Index.html

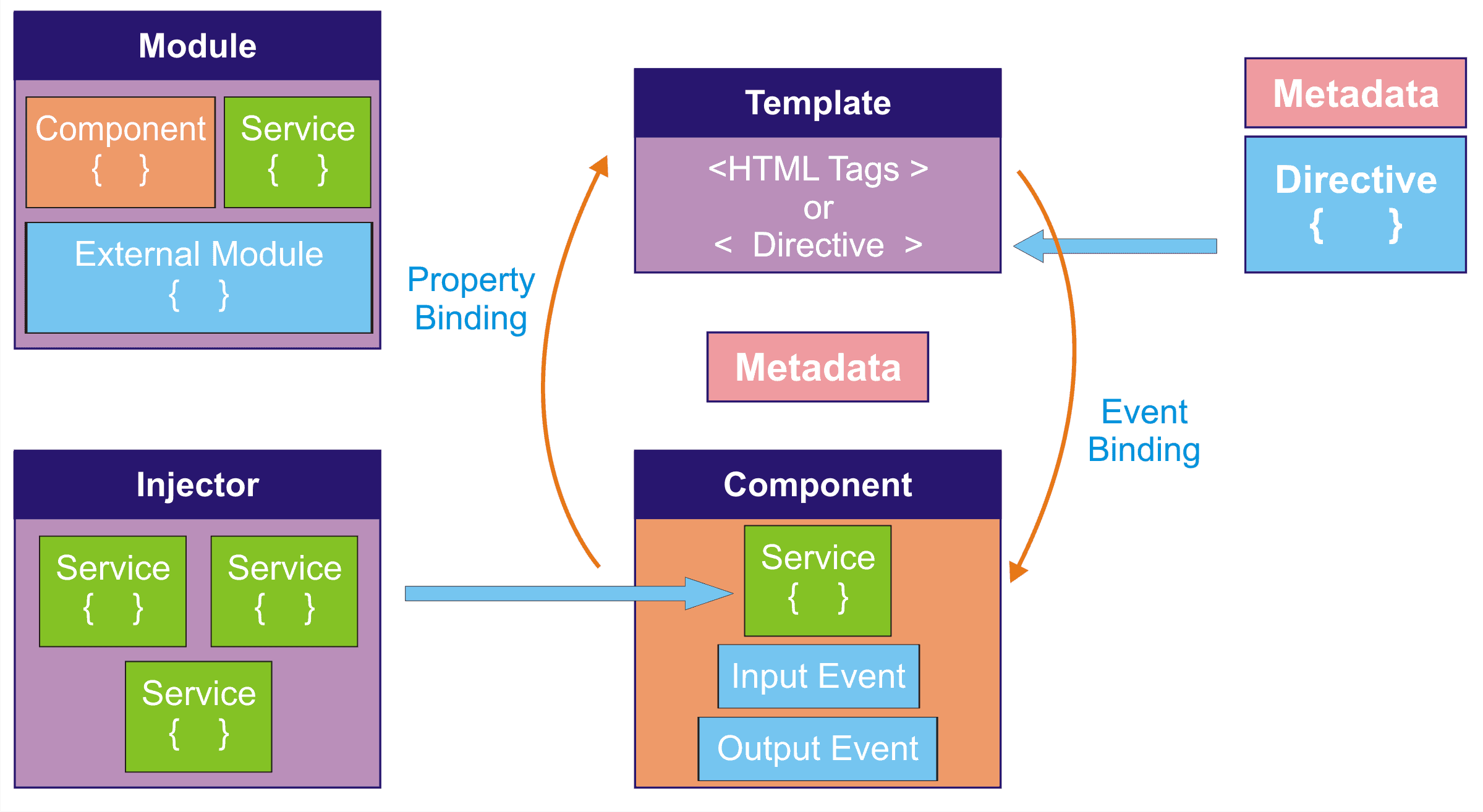
The bootstrapping happens to be a necessary process in Angular- and this is where the app loads and Angular becomes a living thing.

Bootstrapping the Angular application is undoubtedly quite not the same as Angular 1.x however; it is yet quite a clear-cut procedure, and as shown in the execution diagram above.

## **Building blocks of Angular**

Through the Angular framework and platform, you can build with ease single page client applications with the help of HTML and TypeScript. It is written in TypeScript. And it implements the core as well as the optional functionalities as a bundle of libraries in TypeScript which you can import in your apps.

Below Diagram explains it all together:



The following points are required to be noted:

1. The Angular Architecture is based on a list of fundamentals. And the basic building blocks are like **NgModules**, which caters to the compilation context for the components. You will always find one root module in an app that enables the bootstrapping, and in general, comes with a lot more feature modules.
2. The **components** define the view, which is a set of screen elements that the Angular can select from and formulate it as per the logic and the data.
3. The components make use of the services that relish explicit functionalities that are related to the views, however, not directly. The service providers can be injected as dependencies into the components and can make your code quite modular, efficient, and reusable.
4. The modules, services, and components are the classes that make use of the decorators. And these decorators dent its type and leverage the metadata which tells the Angular how the app needs to make use of them.
5. And the component class metadata relates it to the template which defines the view. And the template bundles the HTNL and the directives as well as the binding markups which allow the Angular to redefine the HTML before it renders that for the display.
6. And the service class provider metadata cater to the information which the Angular necessitate for making it ready for the component through the DI or the dependency injection.
7. Also, an app component typically defines a lot of views, which are arranged hierarchically. It caters to the Router service for helping you define the navigation paths amongst views. And the router caters sophistication in the browser capabilities for navigation.

### Modules

The module is the centralized location which stores components, directives, pipes, and services at one location in Angular. Modules are also categorized based on their definition, and we will study that in upcoming chapters.

1. And each of the Angular apps comes with a root module, which is known as **Appmodule**, and it caters to the bootstrap mechanism, which unveils the app. And al App typically has a lot of functional modules.
2. Organizing the code into the various functional modules that leverage the management of the development of the complex applications, as well as in designing for reusability. And together with that, this technique is going to take advantage of the lazy loading, by which we mean on-demand loading of the modules for minimizing the amount of code that is required for loading at the startup.

### Components

1. These are the fundamental UI building block in Angular. An angular app comprises of the tree of Angular component.
2. It’s a subset of directives and is always linked with a template.
3. Each angular application contains at least a component which is the root component that can links the component hierarchy with the page DOM.
4. And each of the components defines the class which has the application data and the logic and is associated with the HTML template which defines the view that is to be rendered for display in the target environment.

### Data binding

You will find two types of data binding.

1. The event binding helps your app to respond to specific user input in the target environment through the updating of the application data.
2. And the property binding helps you in interpolating the values which are being calculated from the application data inside the HTML.

And before the view is displayed, the Angular determines the directive as well as resolves the binding syntax in the template for modification of the HTML elements and DOM, as per the logic and the program data. The angular is up with two-way data binding, which means that changes in the DOM like user choices are also being reflected in the program data.

### Decorators

Decorators are the functions that allow the modification of the directives, services, or filters before their usage. They are the design patterns that separate the modification or decoration of a class without requiring modification in source code.

### Pipes

Your template can cater to the pipes for bettering the experience of the user through the transformation of the values for the display. As an example, the use of pipes for displaying the dates as well as currency values which are appropriate for the user’s locale. The Angular caters pre-defined pipes for the common transformation, as well as you can keep up with your pipes also.

### Directives

These are used to extend the power of HTML by leveraging it with a new syntax. Each directive has a name either one taken from Angular predetermined such as ng-if etc. The directive can determine whether it is to be used as an attribute, element, class, or comment.

### Services

For the logic or the data which is not linked with the specific view, and which you like to share with the various components, you can formulate the service class. It is a definition that is immediately heralded by the @Injectable () decorator. It caters to the metadata which helps other providers in being injected like dependencies to the other classes. As an example, we can have one service for HTTP, and one service for validation or one for Authentication.

### Dependency Injection

The dependency injection (DI) helps you to formulate the component classes as efficient and lean. And they never fetch the data that comes from the server, or validate the inputs from the users, or log to the console directly. Such tasks are being delegated by them to the services.

### Routing

### The NgModule Angular Router caters to the services that can help you while you are defining the navigation path amongst various application states and have a look over the hierarchies in the app. And it is being modeled on various well-known conventions related to the browser conventions.

## MetaData

The metadata used for decorating the class such as the expected behavior of the class is configured. And the decorators happen to be the most important concept when we do the Angular development for version 2 and above. The user can make use of the metadata to certain class for telling the Angular app that the component is the **AppComponent**. And the Metadata can be attached with the typescript with the help of the decorator.

# Multiple Choice Questions:

**1. What are the parts of the modules?**

a) Directives, Components, Pipes, Services

b) Ng\_modules, bootstrap, providers, import

c) Export, import

d) Formsmodule, coremodule, featuremodule,sharedmodule

**2. Which out of this is true?**

a) Components are the basis UI building blocks of angular

b) Modules are the basic UI building blocks of the angular

c) Directives and the basic UI building blocks of the Angular

d) Components are part of Modules

**3. Which prepares the app for loading?**

a) Boostrapping app b) ng serve

c) ng build d) ng new

**4. Which are the building blocks of Angular?**

a) Modules, components b) Directives, data binding

c) Bootstrapping, styles d) Pipes, services, dependency injection

**5. What is the full form of npm?**

a) Node packet manager a) Native packet manager

c) Node component manager d) Node module manager

**6. Which version was not released?**

a) Angular 2 b) Angular 3

c) Angular 4 d) Angular 5

**7. Which page first loads when the app is started?**

a) Index.html b) app component

c) last defined component d) AppModule

**8. Which out of the following is a complete framework?**

a) Angular b) React

c) Vue.js d) Bootstrap

**9. Which of these were developed after 2015?**

a) Angular 2 b) Angular 3

c) Angular 4 d) Angular JS

**10. Which of the following is/are true?**

a) The component is always attached to a template.

b) The module is always attached to a decorator.

c) The module is always attached to a template.

d) NgModule is part of each Module.

**11. Which of the following are true?**

a) All components must be listed in App.Module.ts before they are used

anywhere else.

b) The difference between the two imports is that one is for typescript and the

inside the class import is for angular.

c) Angular is a complete framework.

d) React and Angular both support component structure.

# 

# FAQs / LAB Assignments

1. What are parts of a Module? Can we have more than one module in an Angular App?
2. What is the importance of a decorator in Angular?
3. To what is a template bind within Angular?
4. Which tag should be placed in the index, HTML?
5. What is dependency Injection? Please just mention the definition.
6. Can you have multiple services in an Angular Application? Explain.
7. What are directives? Is the component a directive?
8. What are the pipes?
9. What is data binding?
10. What is the bootstrapping of the app?
11. Please mention five Angular cli commands with syntax.
12. Please mention five npm commands with syntax.
13. Create an App myfirstApp and explain the steps.
14. Which style formats are allowed in Angular? Please brief. Hint check the steps for creating an app on your computer, and the step where you select CSS.
15. What is typescript? Which is more suitable for Angular-TypeScript or JS6?
16. Explain Angular fundamental blocks in a nutshell.
17. Illustrate briefly Angular History starting from AngularJS till Angular10.
18. Explain with an example for each the Angular CLI commands.
19. Explain briefly with example NPM commands.
20. What do you understand by Bootstrapping of the App? Explain briefly with example?
21. Explain the concept of Metadata in angular.
22. Explain briefly components.
23. Briefly describe Data binding in Angular.
24. Briefly explain Modules in Angular.
25. Briefly explain Routing in Angular.
26. What are templates in Angular?
27. What are directives in Angular?
28. Explain briefly templates and directives in Angular.
29. What are the new features of Angular 10? Make a note of new features for each version of Angular. And brief its history.
30. Write down all Angular cli commands.
31. Write down all NPM commands.

# Summary

We discussed in chapter:

1. What is Angular?
2. What is the history of Angular?
3. What is Environment Prerequisite for Angular app development?
4. What is Angular CLI, and what are Angular CLI commands?
5. What is NPM and what are various NPM commands?
6. We created our first angular app and noted all the steps involved.
7. We saw the code in Visual Studio code, and noted down the file/folder structure and what each file and folders mean in the Angular application code.
8. We learned what is bootstrapping of the App, and saw through the execution diagram how an Angular App is executed.
9. We saw and learned about all the building blocks of the Angular framework.

And that concluded the first chapter.