Angular 8

## What is Angular 8?

Angular 8 is a client-side TypeScript based framework which is used to create dynamic web applications. It is very similar to its previous versions except having some extensive features.

**These are the most prominent features of Angular 8:**

* Angular 8 supports TypeScript 3.4
* Angular 8 supports Web Workers
* The new compiler for Angular 8 is Ivy Rendering Engine
* Angular 8 provides dynamic imports for lazy-loaded modules.
* Improvement of ngUpgrade

# Angular 8 Introduction

Angular is the most popular JavaScript framework and platform for developing client-side (front-end) mobile and desktop web apps or single page applications (SPAs).

Angular community has released its latest version known as Angular 8. If you are familiar with previous version of Angular, it will not be difficult for you. You can easily upgrade your older version of Angular to latest version Angular 8.

## What is Angular 8?

Angular 8 is an open-source, client-side TypeScript based JavaScript framework. It is written in TypeScript and complied into JavaScript. Angular 8 is used to create **dynamic web applications**. It is very similar to its previous versions except having some extensive features.

### What is a dynamic web application?

A dynamic web application is simply a dynamic website. i.e. [www.gmail.com](https://www.gmail.com/), [www.facebook.com](https://www.facebook.com/), [www.yahoo.com](https://www.yahoo.com/) etc. which has a tendency to change data/information with respect to 3 parameters:

Hello Java Program for Beginners

* Time-to-time (eg. news update webs applications)
* Location-to-location (eg. Weather-report web applications)
* User-to-user (eg. Gmail, Facebook type applications)

## Angular Previous Versions

* **AngularJS (also known as Angular 1.0):** AngularJS is a JavaScript based open-source frontend web framework developed and maintained by Google. AngularJS can be added to an HTML page with a <script> tag. Because AngularJS was the first version of the Angular, so it is also known as Angular 1. AngularJS was first released on October 20, 2010 by a team of Google.
* **Angular 2:** Angular 2 was a complete rewrite of AngularJS. It was first released in May 2016 and the final version was released on September 14, 2016.
* **Angular 4:** Angular 4 was the updated version of Angular 2. The Google team skipped the Angular 3 to avoid the confusion due to the misalignment of the router package's version which was already distributed as v3.3.0.
* **Angular 5:** Angular 5 was the improved version of the previous one. It was released on November 1, 2017 and improved the support for progressive web apps.
* **Angular 6:** Angular 6 Angular 6 was released on May 4, 2018. It was a major released focused on ng update, ng add, Angular Elements, Angular Material + CDK Components, Angular Material Starter Components, CLI Workspaces, Library Support, Tree Shakable Providers, Animations Performance Improvements, and RxJS v6.
* **Angular 7:** Angular 7 was released on October 18, 2018. It was focused on Application Performance, Angular Material & CDK, Virtual Scrolling, Improved Accessibility etc.
* **Angular 8:** Angular 8 is the latest version running nowadays. Angular 8 is released on May 28, 2019. It is mainly focused on Differential loading, Dynamic imports for lazy routes, web workers and Angular Ivy as an opt-in support. It also supports TypeScript 3.4.

# Features and Advantages of Angular 8

The Angular community has released its latest version Angular 8 with an impressive list of changes and improvements including the much awaited Ivy compiler as an opt-in feature.

**Most prominent features of Angular 8:**

* Support TypeScript 3.4
* Supports Web Workers
* Preview of Ivy available
* Lazy loading
* Improvement of ngUpgrade

## TypeScript 3.4

Angular 8 supports TypeScript 3.4 and it is required to run your Angular 8 project. So, you have to upgrade your TypeScript version to 3.4. TypeScript 3.4 introduces a new flag called --incremental. The incremental tells TypeScript to save information about the project graph from the last compilation. Every time the TypeScript is invoked with --incremental, it will use that information to detect the least costly way to type-check and emit changes to your project.

## Web workers class

JavaScript is single threaded, so it is common for more critical tasks like data calls to take place asynchronously. Web Workers facilitates you to run the CPU intensive computations in the background thread, freeing the main thread to update the user interface.

Web workers can also be helpful, if your application is unresponsive while processing data.

If you want to outsource such a calculation to a background, we must first create the web worker using the Angular CLI.

## Preview of Ivy and Bazel available

After the release of Angular 8, a preview version of Ivy is now available for testing. Ivy is the new rendering engine that produces small bundle size and Bazel is the new build system. Both are ready for proper use with Angular 8. The preview of these two should be available shortly. Ivy is a new compiler/runtime of Angular and Angular 8 is a first release to offer a switch to opt-in into Ivy officially.

To use Ivy in your project, you can instruct the Angular CLI to enable Ivy in your project using the --enable-ivy switch:

1. $ ng new angular-project --enable-ivy

Ivy is supposed to be a by default rendering engine in Angular version 9.

Bazel provides one of the newest features of Angular 8 as a possibility to build your CLI application more quickly.

**The main advantages of Bazel are:**

* The incremental build and tests.
* It provides a chance to make your backends and frontends with a same tool.
* It has a possibility to have remote builds and cache on the build farm.
* Dynamic imports for lazy-loaded modules

## Lazy Loading

Angular 8 facilitates you to use standard dynamic import syntax instead of a custom string for lazy-loaded modules.

**It means lazy-loaded import that looked like this:**

1. { path: '/student', loadChildren: './student/student.module#StudentModule' }

**Will be looked like this:**

1. { path: '/student', loadChildren: () =**>** import('./student/student.module').then(s =**>** s.StudentModule) }

## Improvement of ngUpgrade

The Angular CLI is continuously improving. Now, the ng build, ng test and ng run are equipped by 3rd-party libraries and tool. For example, AngularFire already makes use of these new capabilities with a deploy command.

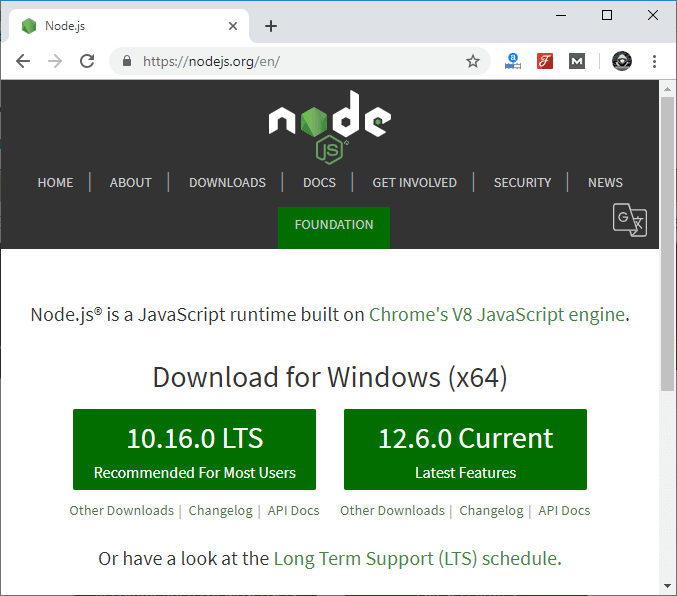
# Angular 8 Installation

(How to install Angular 8 or Angular 8 Environment setup)

Before to setup environment for Angular development using the Angular CLI tool, you must have installed Node.js on your system and set a development environment and npm package manager.

## Install Node.js

Angular requires Node.js version 10.9.0 or later. You can download it from <https://nodejs.org/en/>



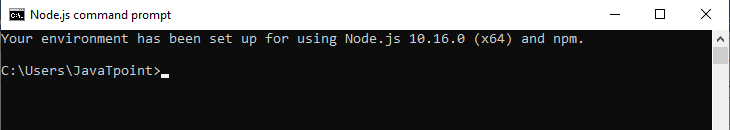
After downloading, you have to install it on your system.

HTML Tutorial

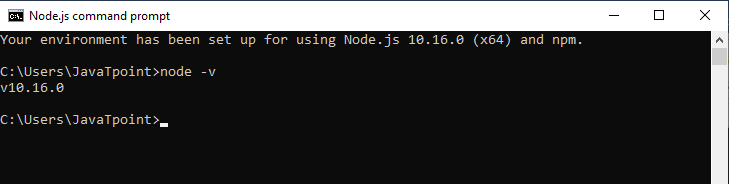
See how to install Node.js on Windows: [**Click Here**](https://www.javatpoint.com/install-nodejs)

See how to install Node.js on Linux/ Ubuntu/ CentOS: [**Click Here**](https://www.javatpoint.com/install-nodejs-on-linux-ubuntu-centos)

Once you have installed Node.js on your system, open node.js command prompt.



* To check your version, run **node -v** in a terminal/console window.



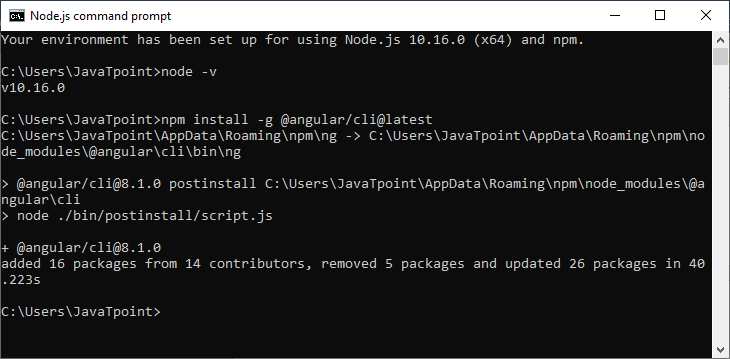
## Use npm to Install Angular CLI

Use the following command to install Angular CLI

1. npm install -g @angular/cli

Or

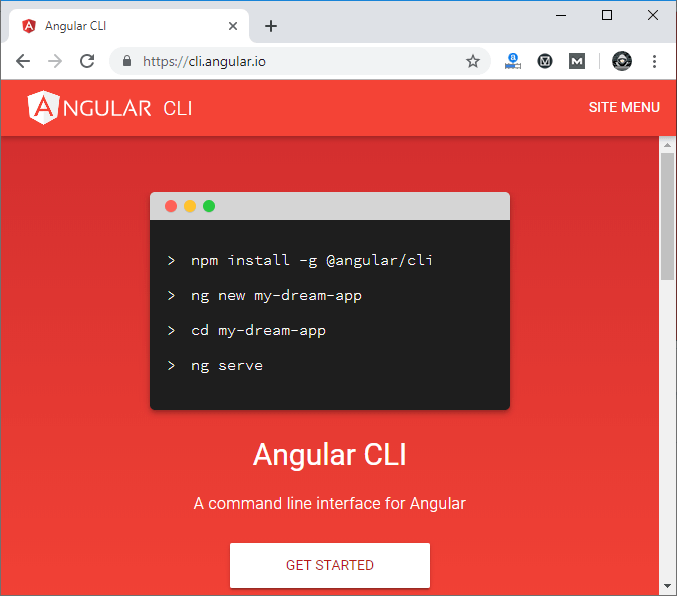
1. npm install -g @angular/cli@latest



Or

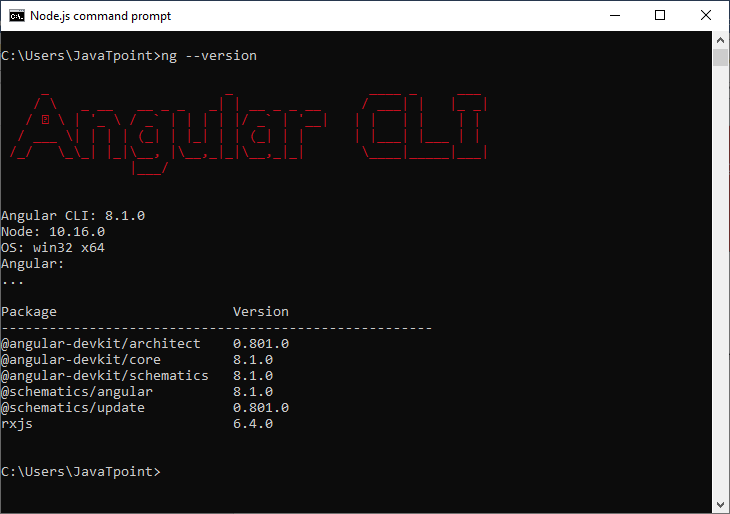
Just go to Angular CLI official website <https://cli.angular.io/>

You will see the whole cli command to create an Angular app. You need to run the first command to install Angular CLI. These steps are same for Windows and Mac.



## Check your Installed versions

* To check Node and Angular CLI version, use ng --version command.



Now, Angular 8 is installed on your system.

# Angular 8 First App

Let's see how to create an Angular 8 application.

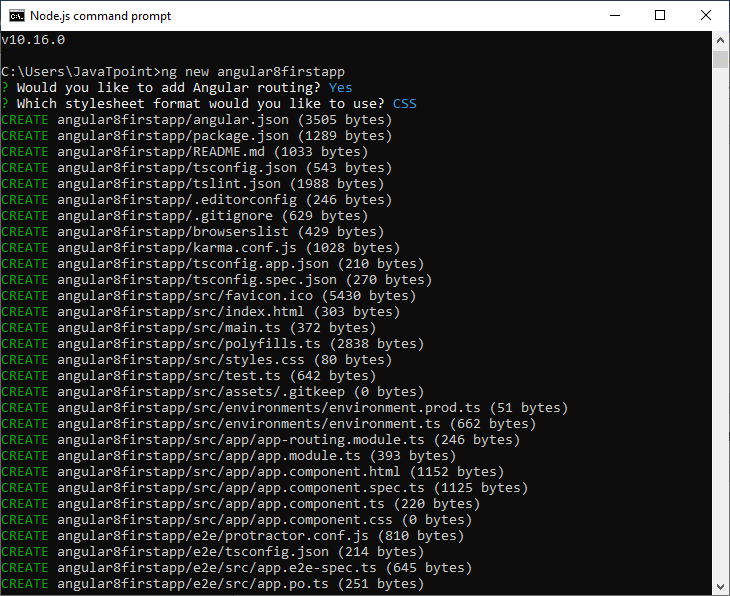
### To create an app

**Syntax:**

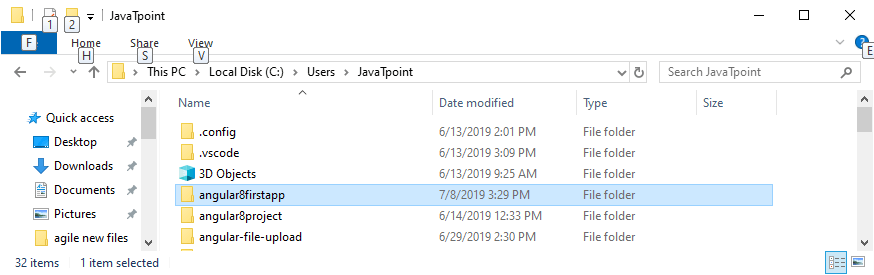
1. ng new app\_name

**For example:** Here, we are going to create an app named "angular8firstapp"

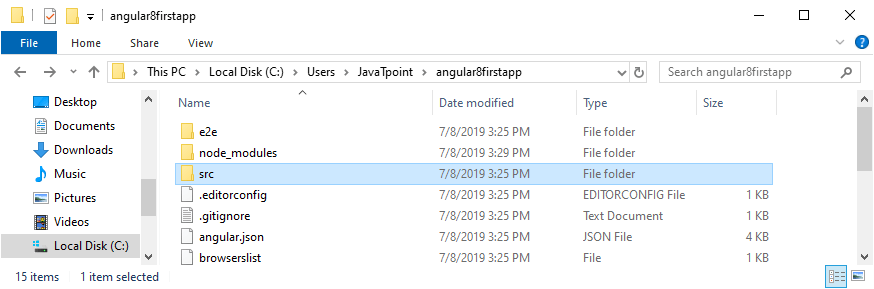
1. ng new angular8firstapp



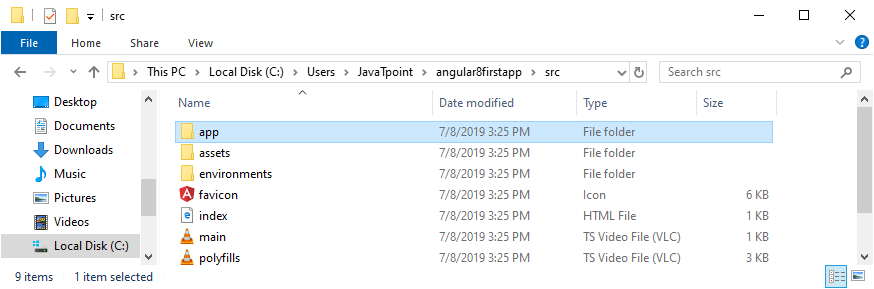
You can see that a folder is created. This is your first created app of Angular 8.



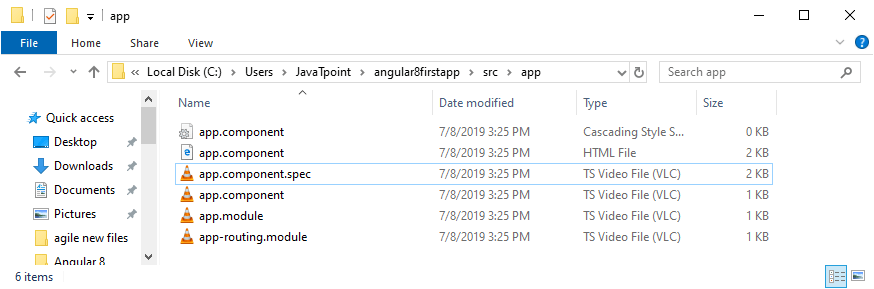
Open this folder and you will see the some subfolders.



Here, **src** is the main folder of your project. Open the src folder and you will see some other subfolders.



This app folder is the root of your Angular 8 app. Open this folder and you will see some .ts, html and css files.



## Files used in Angular 7 App folder

Angular 7 App files which are mainly used in your project are given below:

* **src folder:** This is the folder which contains the main code files related to your angular application.
* **app folder:** The app folder contains the files, you have created for app components.
* **app.component.css:** This file contains the cascading style sheets code for your app component.
* **app.component.html:** This file contains the html file related to app component. This is the template file which is used by angular to do the data binding.
* **app.component.spec.ts:** This file is a unit testing file related to app component. This file is used along with other unit tests. It is run from Angular CLI by the command ng test.
* **app.component.ts:** This is the most important typescript file which includes the view logic behind the component.
* **app.module.ts:** This is also a typescript file which includes all the dependencies for the website. This file is used to define the needed modules to be imported, the components to be declared and the main component to be bootstrapped.

## Install Visual Studio Code IDE or JetBrains WebStorm

You must have an IDE like Visual Studio Code IDE or JetBrains WebStorm to run your Angular 7 app.

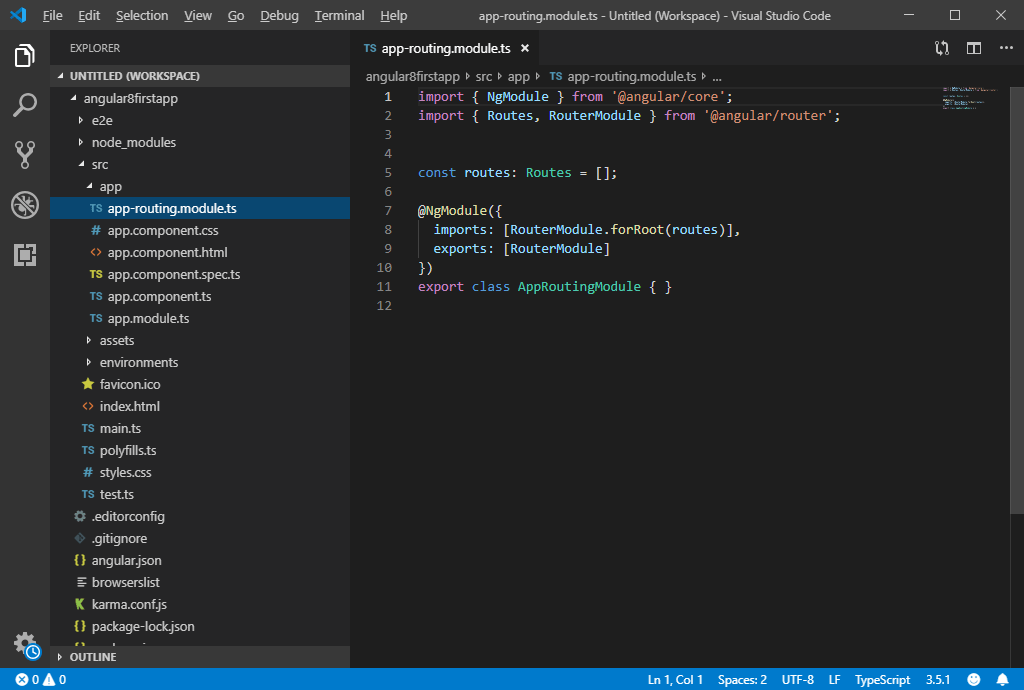
**VS Code** is light and easy to setup, it has a great range of built-in code editing, formatting, and refactoring features. It is free to use. It also provides a huge number of extensions that will significantly increase your productivity.

You can download VS Code from here: [https://code.visualstudio.com](https://code.visualstudio.com/)

**JetBrains WebStorm** is also a great IDE to develop Angular 7 apps. It is fast, attractive, and very easy to use software but, it is not free to use. You have to purchase it later, it only provides a trial period of 30 days for free.

You can download Jetbrains Webstorm from here: [https://www.jetbrains.com/webstorm/download/#section=windows](https://www.jetbrains.com/webstorm/download/#section=windows/)

Here, we are using VS Code IDE:

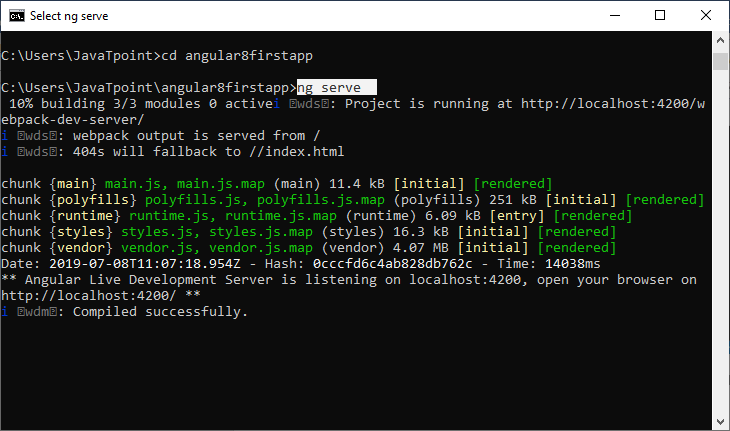


You can see that your project is open in the VS Code IDE. You can also make changes in .ts and html files to change your output accordingly.

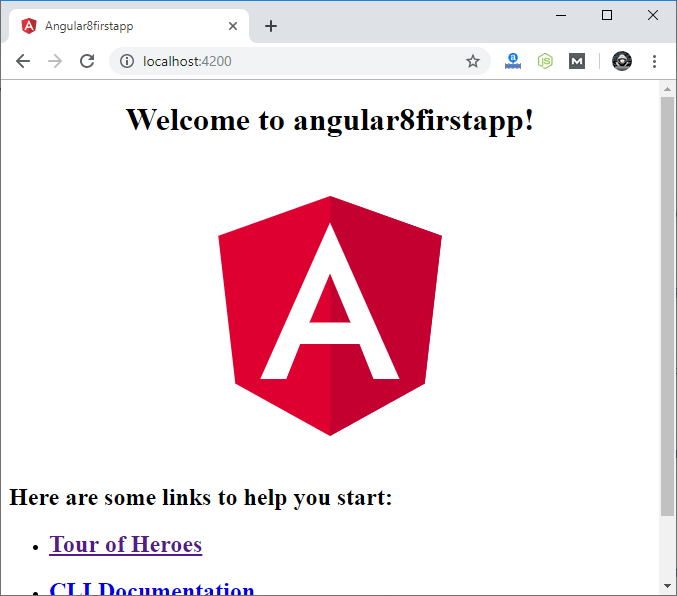
## Run your app

Open your node.js command prompt and go to your project by using cd command and then run the **ng serve** command to compile and run your app.

1. ng serve



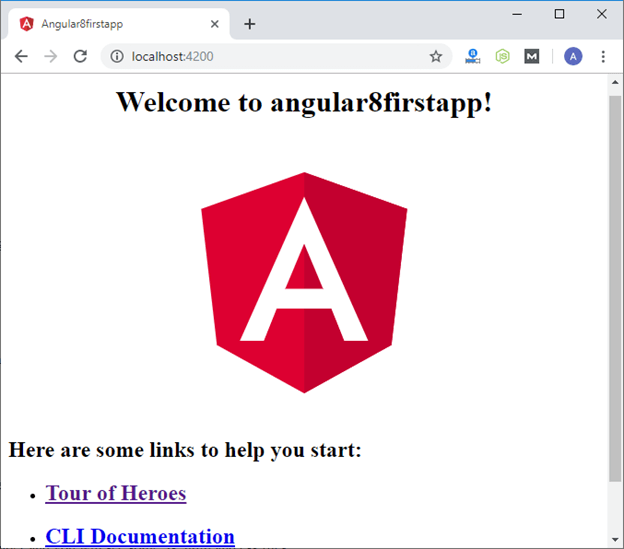
Open your browser and go to local host at: <http://localhost:4200/>



Now, you can see that your app is running now.

How an Angular's app get loaded and started

When you create an Angular app and run it by using *ng serve* command, it looks like the following image.



It is a simple Angular app created by using ng new app\_name command and nothing is edited in the app. The name of the app is angular8firstapp.

Now, we will learn how the Angular's app is loaded and started.

Let's remove all the code from the app.compoment.html file and write some basic HTML code. For example:

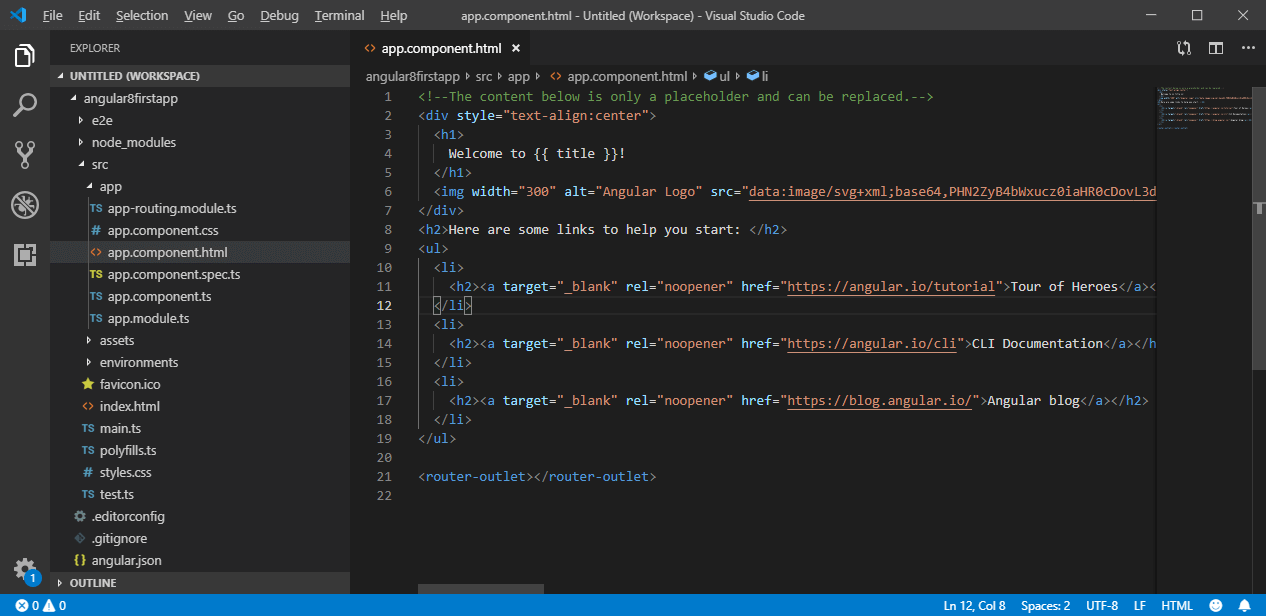
25.2M

584

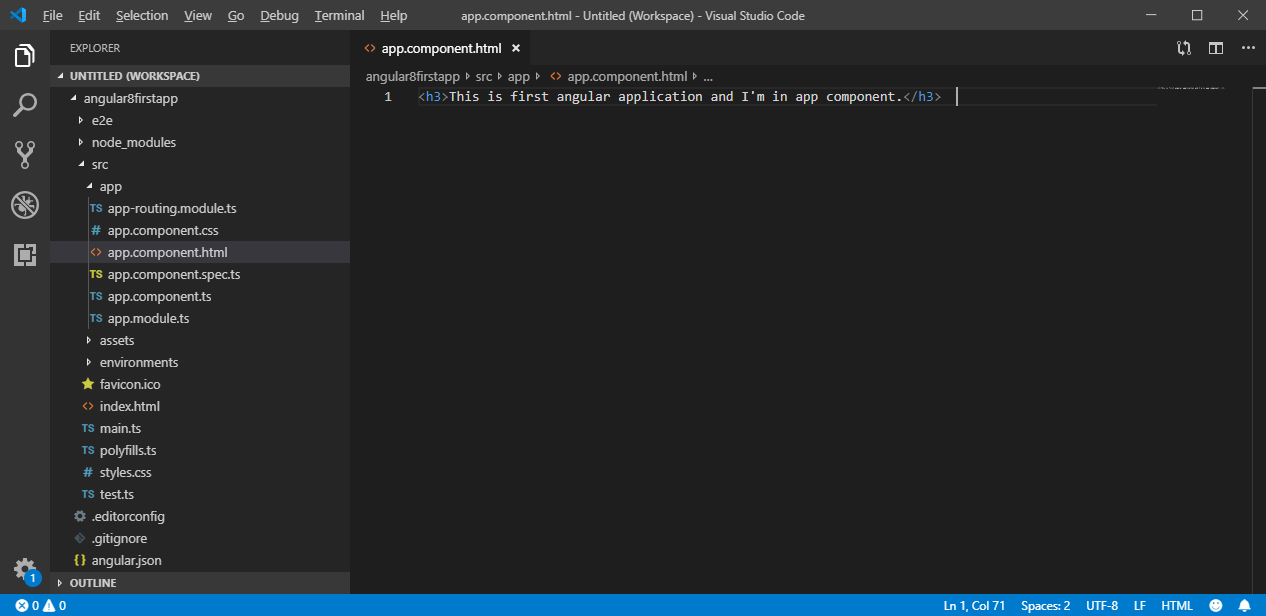
Features of Java - Javatpoint

1. **<h3>**This is first angular application and I'm in app component.**</h3>**

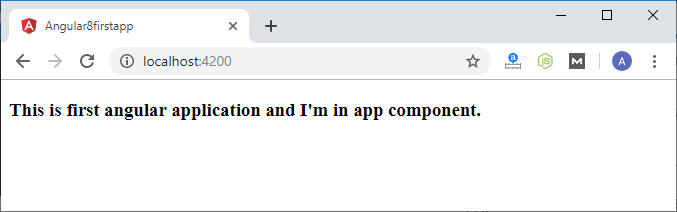
This is the original code in the app.compoment.html file



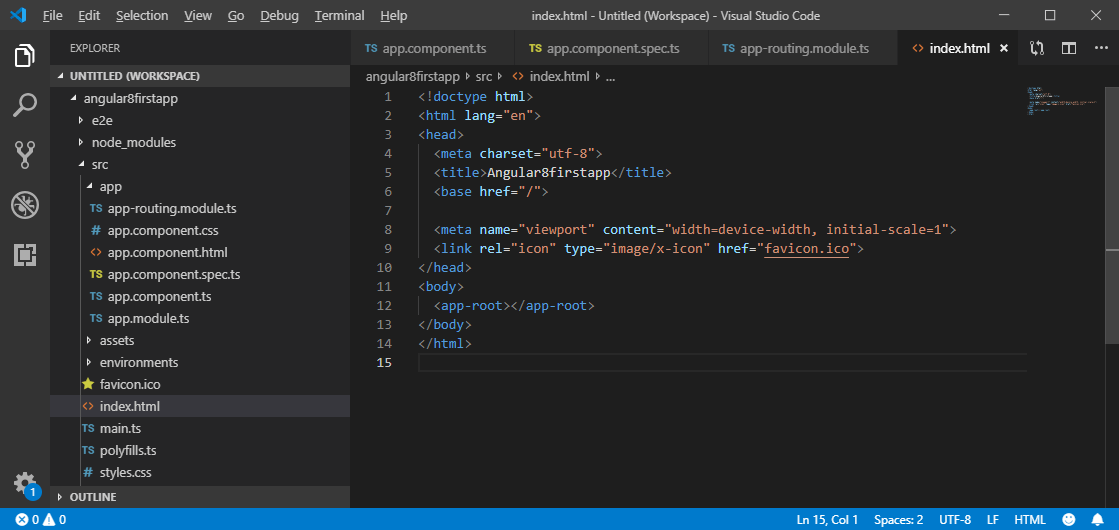
Now, it is replaced and looked like this:



You can also see it in browser:



Here, the above file is not served by the server. The server served an **index.html** file.



Angular is a framework which allows us to create "Single Page Applications", and here the **index.html** is the single page which was provided by the server.

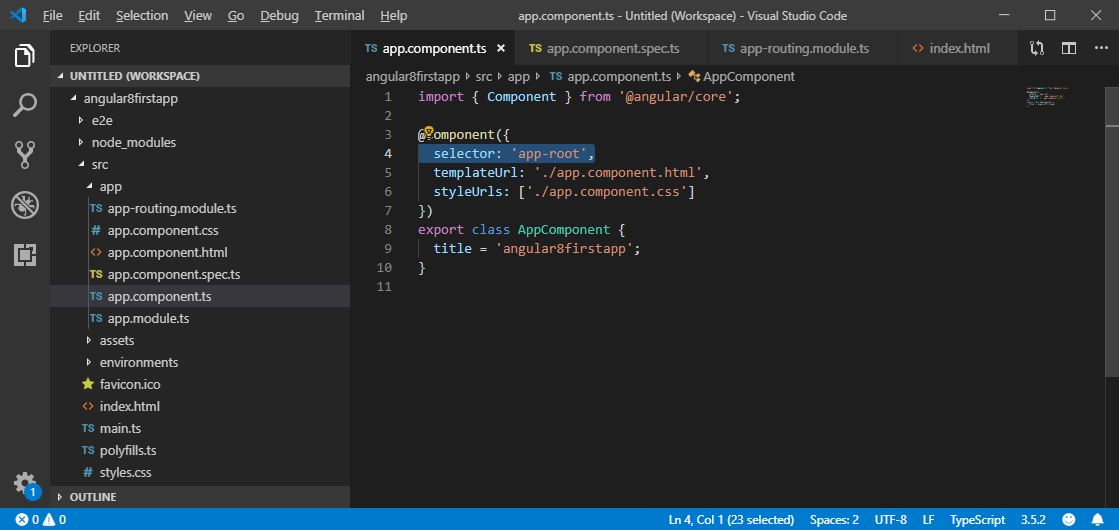
**Index.html:**

1. <!doctype html>
2. <html lang="en">
3. <head>
4. <meta charset="utf-8">
5. <title>Angular8firstapp</title>
6. <base href="/">
8. <meta name="viewport" content="width=device-width, initial-scale=1">
9. <link rel="icon" type="image/x-icon" href="favicon.ico">
10. </head>
11. <body>
12. <app-root></app-root>
13. </body>
14. </html>

The above code looks like a normal HTML code and here the <title> tag shows the same title in the browser as the app's title. But the <body> code is different from normal HTML code. Here, you see "<app-root>" tag which is provided by the CLI. We can say that, whenever we create a project from CLI, by default, one component is created, i.e., "app component".

Now, see the "**app.component.ts**" file. It is a TypeScript file. Here, you see the "selector" property.

1. **import** { Component } from '@angular/core';
2. @Component({
3. selector: 'app-root',
4. templateUrl: './app.component.html',
5. styleUrls: ['./app.component.css']
6. })
7. export **class** AppComponent {
8. title = 'angular8firstapp';
9. }



You can see that the selector property contains the string as index.html file. This information is required the Angular to place this part into an index.html file with the template of the component.

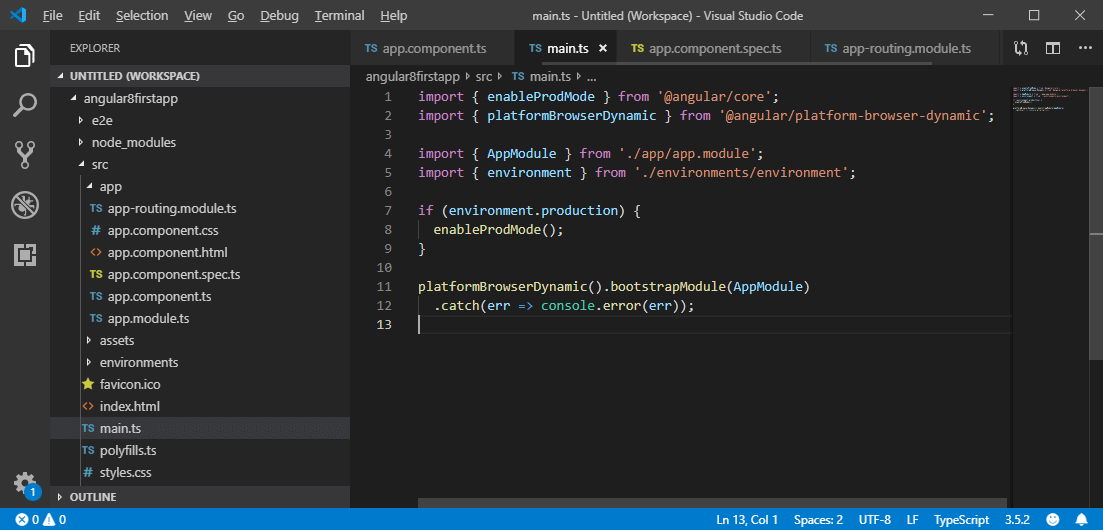
The template of the component is "./app.component.html", so, Angular includes this part into the body of the index.html file.

Now, you see how an "app-root" is included in index.html file. Now, let's see "How does Angular trigger?"

Whenever ng-serve builds the application, it creates "bundles" and automatically adds these to index.html file at runtime. So, from these bundles, the first code has to be executed from "main.ts" file, i.e., "main.ts" file is the main file from where the execution of an Angular application starts.

**Main.ts file:**

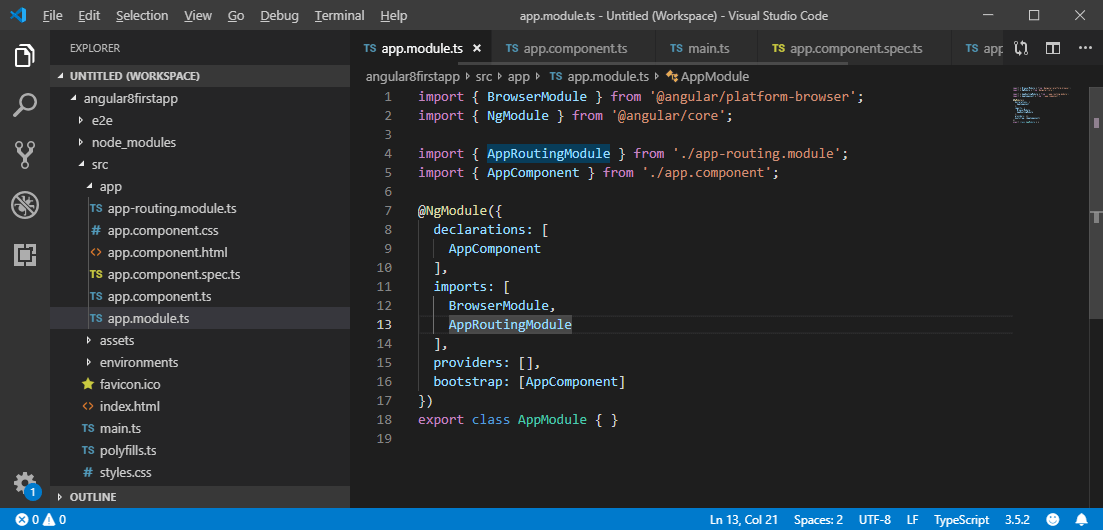
1. **import** { enableProdMode } from '@angular/core';
2. **import** { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3. **import** { AppModule } from './app/app.module';
4. **import** { environment } from './environments/environment';
5. **if** (environment.production) {
6. enableProdMode();
7. }
8. platformBrowserDynamic().bootstrapModule(AppModule)
9. .**catch**(err => console.error(err));



Here, the bootstrap method starts the Angular application. It refers to AppModule, which looks into the app folders. You can see in the "app.module" file that a bootstrap array which is basically a list of all the components analyzes the index.html file.

**See app.module.ts file:**

1. **import** { BrowserModule } from '@angular/platform-browser';
2. **import** { NgModule } from '@angular/core';
3. **import** { AppRoutingModule } from './app-routing.module';
4. **import** { AppComponent } from './app.component';
5. @NgModule({
6. declarations: [
7. AppComponent
8. ],
9. imports: [
10. BrowserModule,
11. AppRoutingModule
12. ],
13. providers: [],
14. bootstrap: [AppComponent]
15. })
16. export **class** AppModule { }



Now, you can see that the Angular application gets loaded as:

1. main.ts  **>>**   app.Module.ts  **>>**  app.component.ts  **>>**  index.html  **>>**  app.component.html

* main focus should be inside src ->app folder by default all files will be created in App folder.
* Every single page application will have index.html
* Inside index.html it has app-root its called a selector
* If u go to the app.component.ts it has selector app-root and app name.
* Go to the app-module.ts in angular application will have atleast one @ngmodule and it has to bootstrap atlease one component our case appcomponent and just double click on app component.
* Component got created in app.component.ts and got bootstrapped in app.module.ts
* @ngmodule will have all the components u created or some one created
* In imports we have to declare other angularmaterial and other imports
* First will go to index.html will have app-root so it will go to the app.component.ts
* In package.json will have all dependencies.