Angular 4 is a JavaScript framework for building web applications and apps in JavaScript, html, and TypeScript, which is a superset of JavaScript. Angular provides built-in features for animation, http service, and materials which in turn has features such as auto-complete, navigation, toolbar, menus, etc. The code is written in TypeScript, which compiles to JavaScript and displays the same in the browser.

# Why does Angular need Node.js?

It is not mandatory to use node.js for developing angular application. You can very well go ahead without node.js for developing angular application but it would not be wise to do so. Let me explain you some of the reasons how node.js makes angular app development process easier for us:

* Node allows you to spin up a **lightweight web server** to host your application locally in your system.
* **NPM** (Node Package Manager) comes with node.js by default. NPM allows you to manage your dependencies. So, you don’t have to worry for operations like adding a dependency, removing some, updating your package.json.
* Third and the most important, npm gives you **angular cli** or **ng cli**(angular command line interface) . Angular CLI is a great tool for scaffolding your application. So, you don’t need to write boilerplates manually.
* Angular recommends the use of TypeScript. Now, your browser does not understand TypeScript. It needs to be transpiled to JavaScript. Also, you need to bundle your js files and stylesheets together with the html doc so as to get the web app CLI which is ready to be hosted. Angular CLI helps you to do all these behind the scene. By default, ng cli uses **webpack** for bundling your application and is very helpful for beginners who have just jumped into web development with angular as it abstracts such complexities.

To install Angular 4, we require the following:-

* Nodejs
* Npm (Node Package Manager)
* Angular CLI
* IDE for writing your code.

What is Webpack?

Answer: - Webpack handles the importing of 3rd party libraries. For instance, if you are using NPM to install libraries to your project, Webpack will handle the import process to your Angular project and turn them into modules for the developer automatically. Webpack also supports BrowserLink, which recognizes file changes when you change your Angular code, and automatically refreshes the browser that is listening to the app. Additionally, Webpack will handle a process called, “Code Spitting,” which bundles files together for Lazy Loading and better data caching while the project is running. Finally, it also minifies your JavaScript files, which speeds up your program and readies your code for a production environment. Essentially, Webpack works like task runner on windows, and provides developers with a large amount of functionality pre-built in a fresh project.

What is npm?

Answer: - npm, short for Node Package Manager, is two things: first and foremost, it is an online repository for the publishing of open-source Node.js projects; second, it is a command-line utility for interacting with said repository that aids in package installation, version management, and dependency management. A plethora of node.js libraries and applications are published on npm, and many more are added every day. These applications can be searched for on <http://search.npmjs.org/>. Once you have a package you want to install, it can be installed with a single commmand-line command.

Let's say you're hard at work one day, developing the Next Great Application. You come across a problem, and you decide that it's time to use that cool library you keep hearing about - let's use Caolan McMahon's async as an example. Thankfully, npm is very simple to use: you only have to run npm install async, and the specified module will be installed in the current directory under ./node\_modules/. Once installed to your node\_modules folder, you'll be able to use require() on them just like they were built-ins.

Let's look at an example of a global install - let's say coffee-script. The npm command is simple: npm install coffee-script -g. This will typically install the program and put a symlink to it in /usr/local/bin/. This will then allow you to run the program from the console just like any other CLI tool. In this case, running coffee will now allow you to use the coffee-script REPL.

Another important use for npm is dependency management. When you have a node project with a package.json file, you can run npm install from the project root and npm will install all the dependencies listed in the package.json. This makes installing a Node project from a git repo much easier! For example, vows, one of Node's testing frameworks, can be installed from git, and its single dependency, eyes, can be automatically handled.

What is Angular CLI (COMMAND LINE INTERFACE FOR ANGULAR APPS)?

Answer: - Angular cli is a command line interface to scaffold and build angular apps using nodejs style (commonJs) modules. Not only it provides you scalable project structure, instead it handles all common tedious tasks for you out of the box.

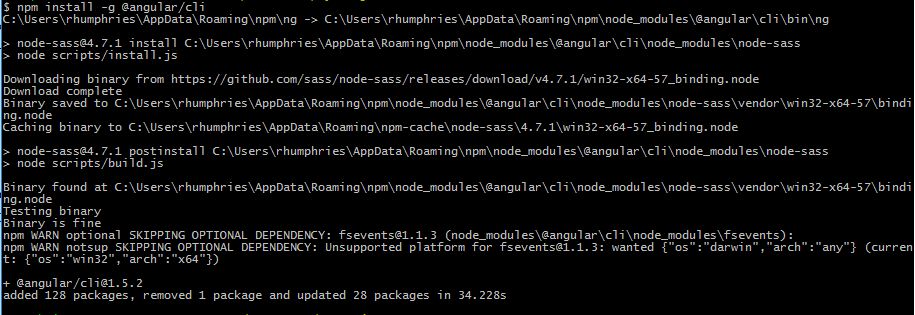
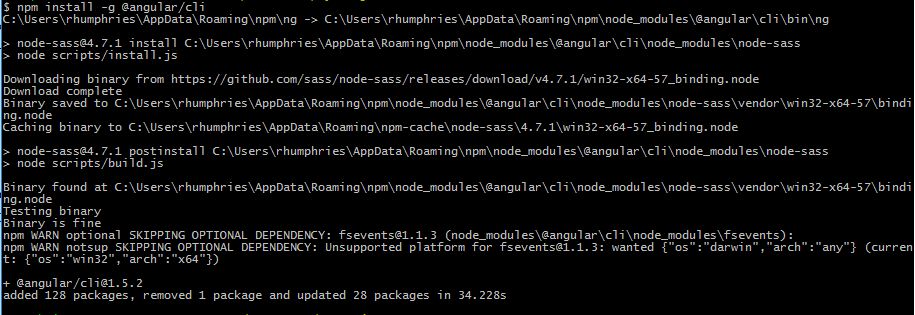
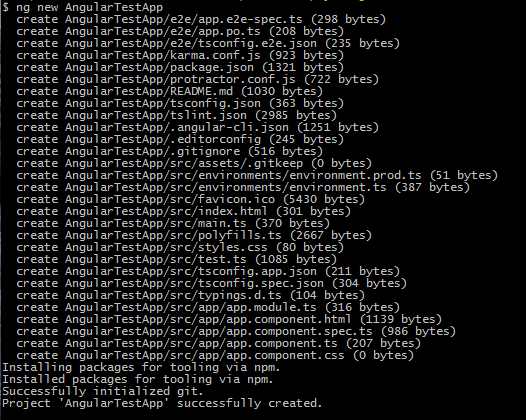
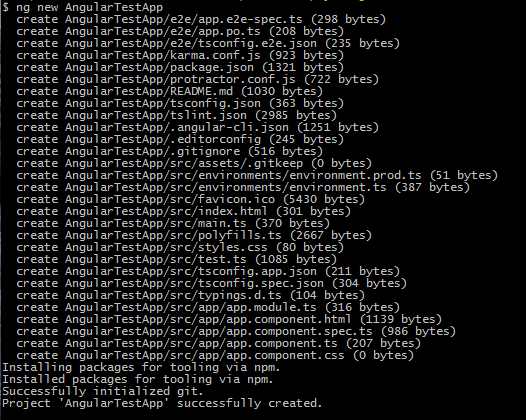
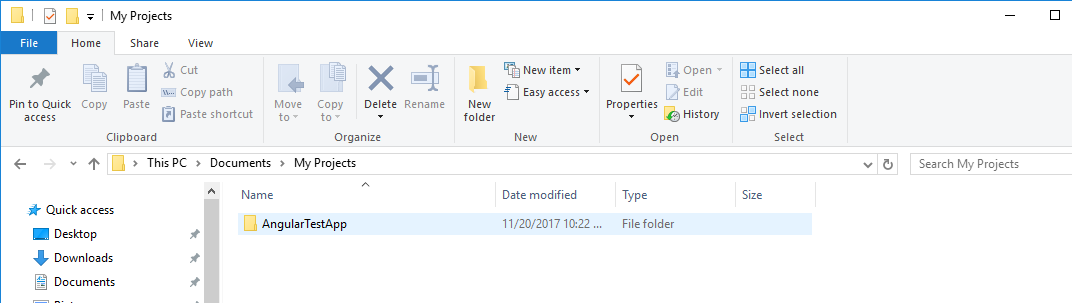
Angular CLI is a command line interface, thus ‘CLI’, which includes the functionality that Webpack provides. It uses Webpack to include all the packaging, importing, BrowserLink etc., but you do not need to know how Webpack works or how it needs to be configured to run in different environments or on different types of machines. All the Webpack configuration is done completely by CLI and leaves it out of the hands of a developer, unless they choose to adjust the settings themselves.

In addition, CLI assists developers by generating code which follows the best practices as defined by https://Angular.io, Angular’s home site. As shown in the next section, creating your Angular project via the command line is done within minutes and you will already have a working app to begin development on.

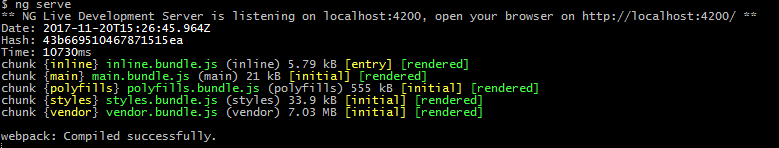
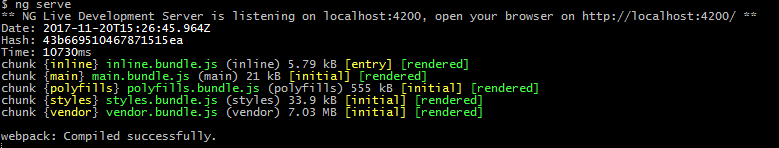
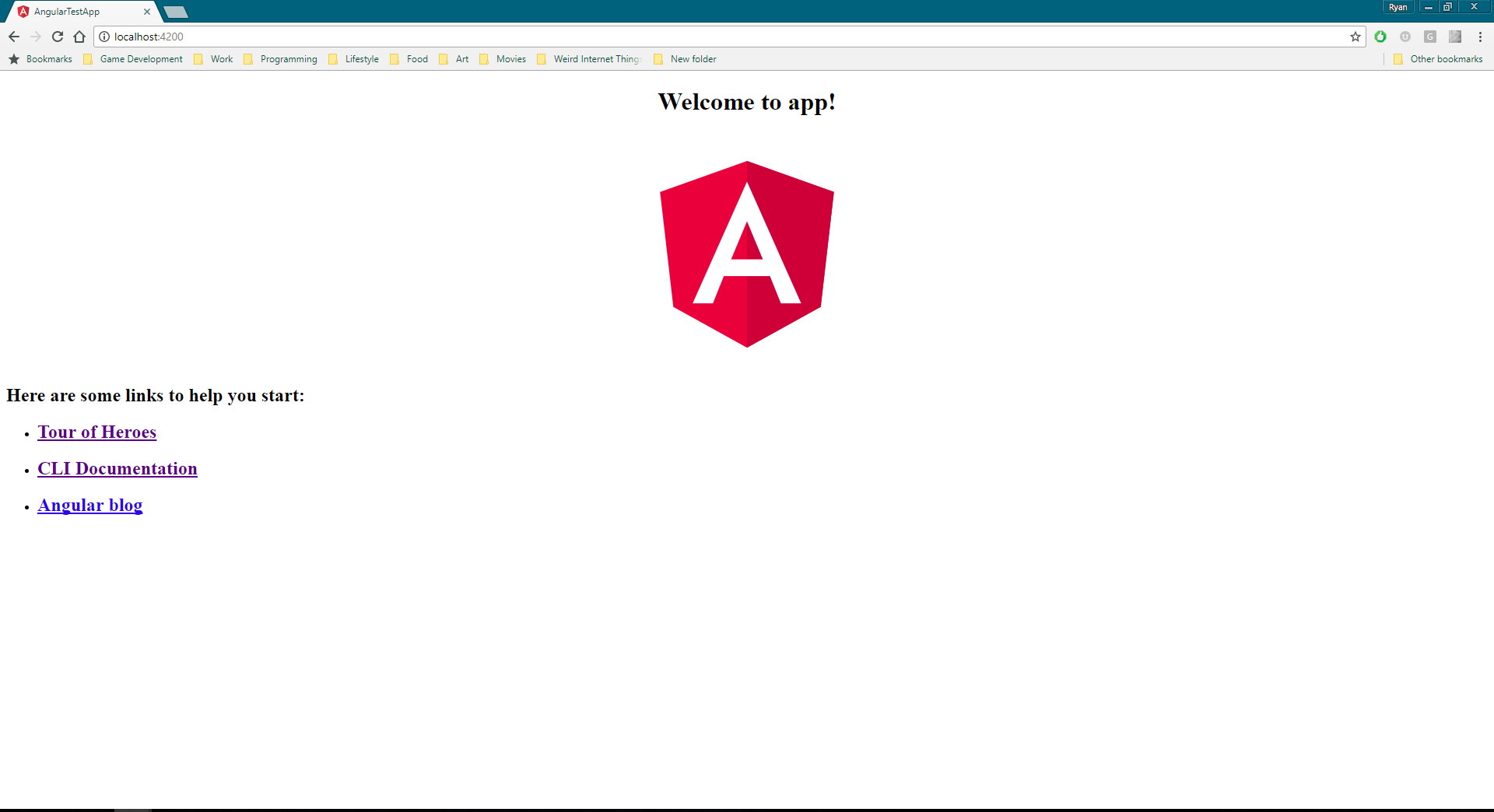
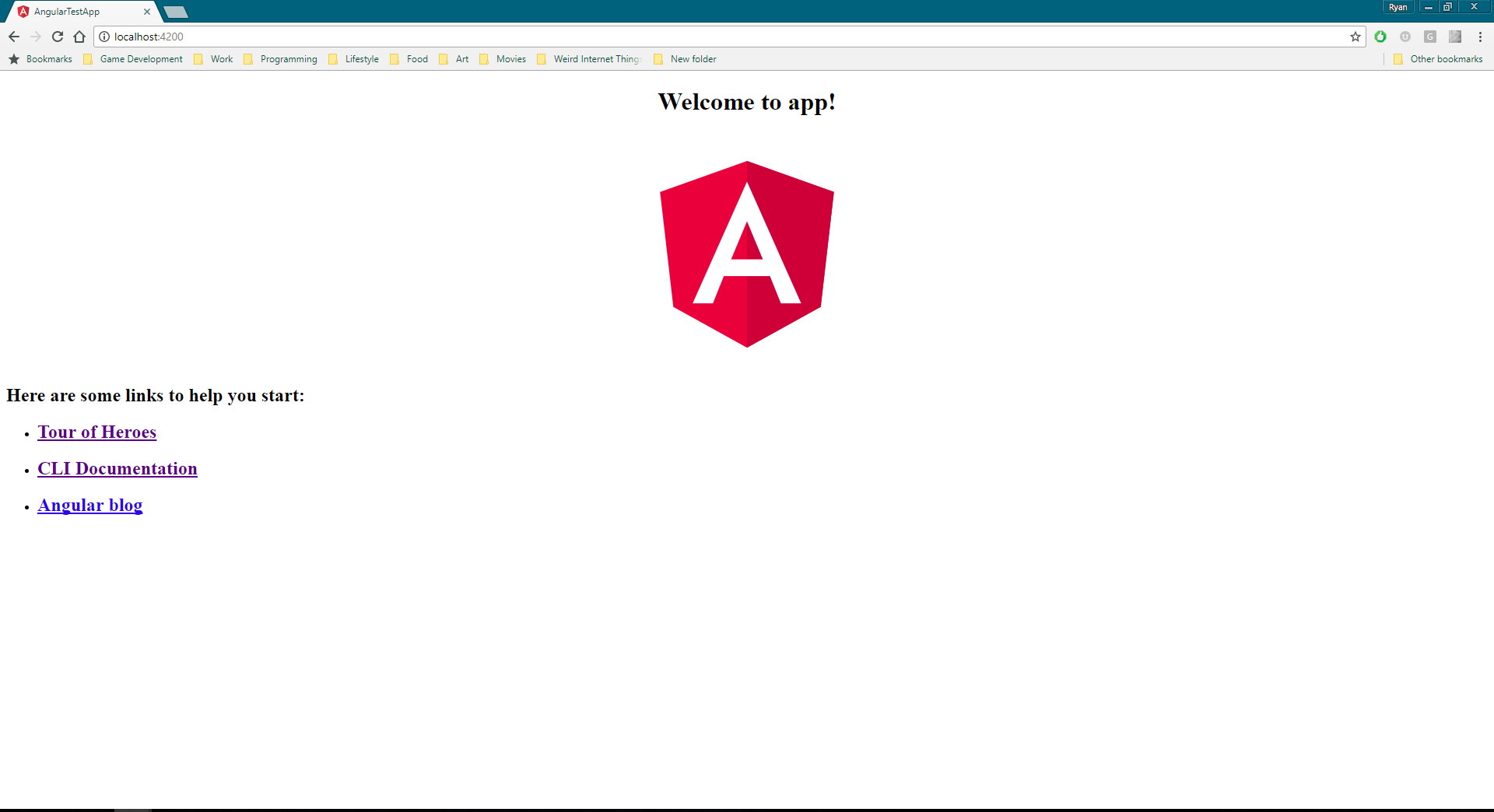
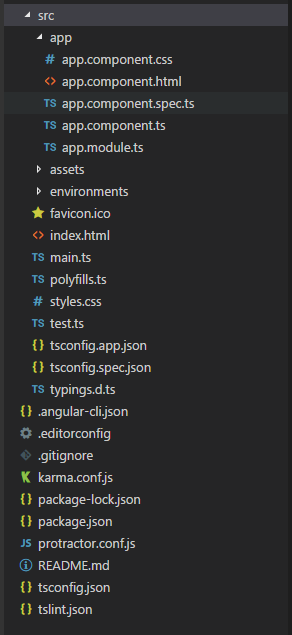
Some are commonly used CLI: -

* ng new: - The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!
* ng generate: - Generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these.
* ng serve: - Easily test your app locally while developing.

## Using and Installing Angular CLI

CLI has made the process very simple and quick to generate a new project to quickly begin development:  
First, if you have never installed CLI before all you need to do is open your command line tool and type ‘**npm install -g @angular/cli**’ which will install CLI globally on your machine.  
Then you can run ‘**ng new AngularTestApp**’ which creates a new project called “AngularTestApp”  
And you will see this appear in the directory you ran ‘ng new’ in:  


* Run ‘cd AngularTestApp’ to step into the newly created project
* Run ‘**ng serve**’ which will run your project locally and you can direct a browser to.

You can see within the ‘ng serve’ result that the “Live Development Server is listening on localhost:4200” Direct a browser to that address to see your executed app:  
And opened in Visual Studio Code, your project will look like this:  


## In Conclusion

And that is all it takes to get started using Angular with CLI and Webpack. Using CLI and Webpack allows for new and experienced users of Angular to easily get started with very few hiccups. With only 4 commands in the command line, you have a functioning app that is set up with Web Link, the most common 3rd party libraries, and a very basic, but executable Angular Project. Developers can very easily begin to add your own modules and components to this base app that CLI provides. You can use very lightweight code editors, like Visual Studio Code, because CLI will manage a majority of the functionality that a larger tool, like Visual Studio 2017, would normally handle. Additional commands for CLI can be found on Angular’s GitHub Page: <https://github.com/angular/angular-cli/wiki> CLI’s functionality allows you to create services, Modules, Components, and more all from the command line, which greatly speeds up development time.

# Architecture overview

Angular is a platform and framework for building client applications in HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

The basic building blocks of an Angular application are *NgModules*, which provide a compilation context for *components*. NgModules collect related code into functional sets; an Angular app is defined by a set of NgModules. An app always has at least a *root module* that enables bootstrapping, and typically has many more *feature modules*.

* Components define *views*, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.
* Components use *services*, which provide specific functionality not directly related to views. Service providers can be *injected* into components as *dependencies*, making your code modular, reusable, and efficient.

Both components and services are simply classes, with *decorators* that mark their type and provide metadata that tells Angular how to use them.

* The metadata for a component class associates it with a *template* that defines a view. A template combines ordinary HTML with Angular *directives* and *binding markup* that allow Angular to modify the HTML before rendering it for display.
* The metadata for a service class provides the information Angular needs to make it available to components through *dependency injection (DI)*.

An app's components typically define many views, arranged hierarchically. Angular provides the [Router](https://angular.io/api/router/Router) service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.

aa