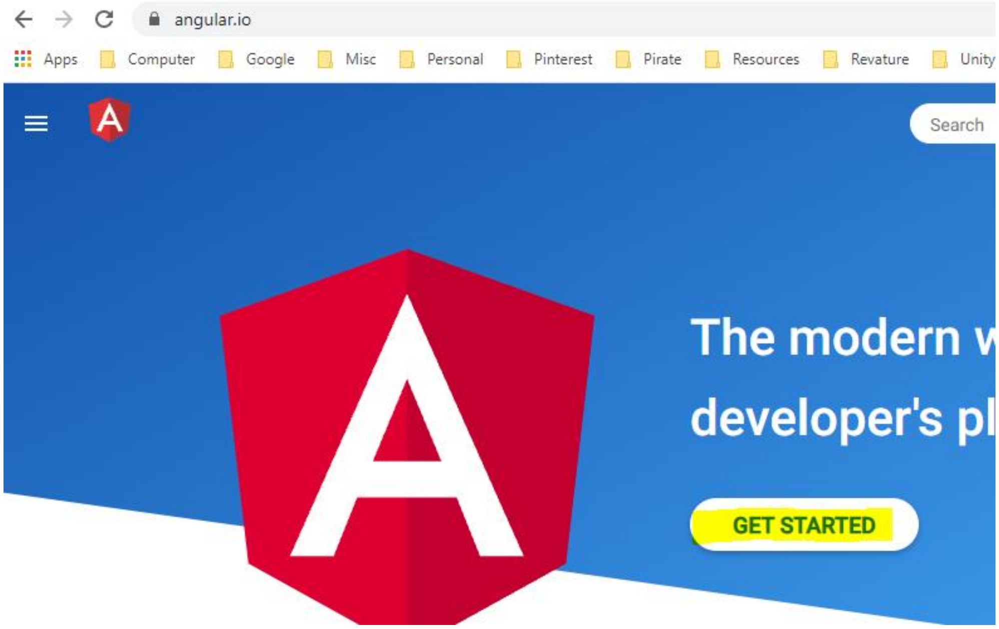
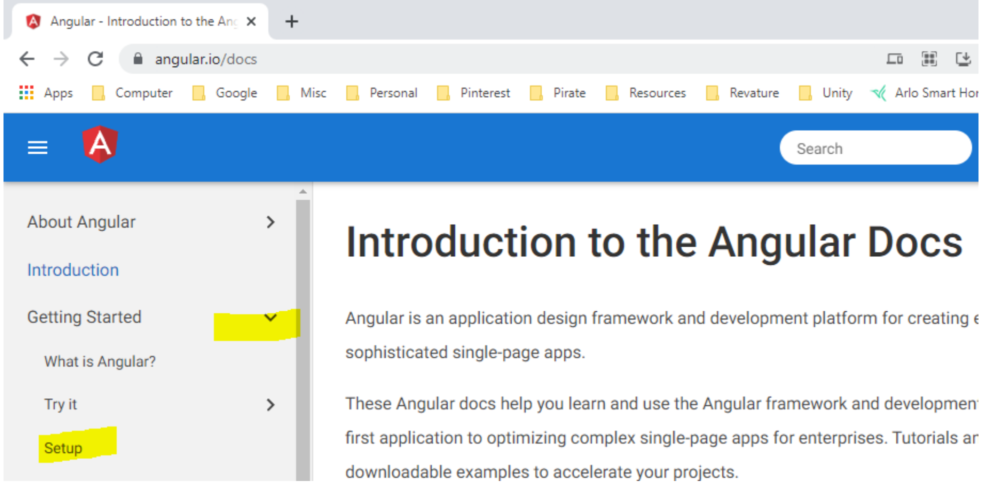
https://angular.io/



https://angular.io/docs

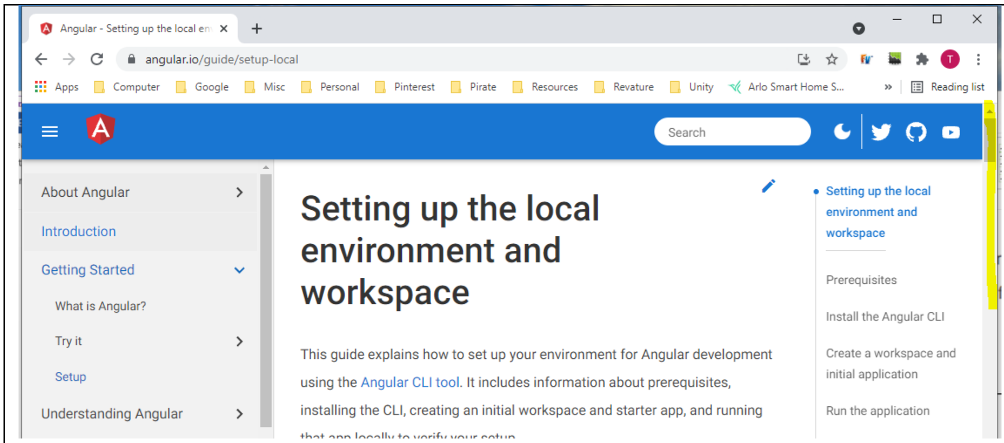


Introduction to the Angular Docs

Angular is an application design framework and development platform for creating e sophisticated single-page apps.

These Angular docs help you learn and use the Angular framework and developmen first application to optimizing complex single-page apps for enterprises. Tutorials ar downloadable examples to accelerate your projects.

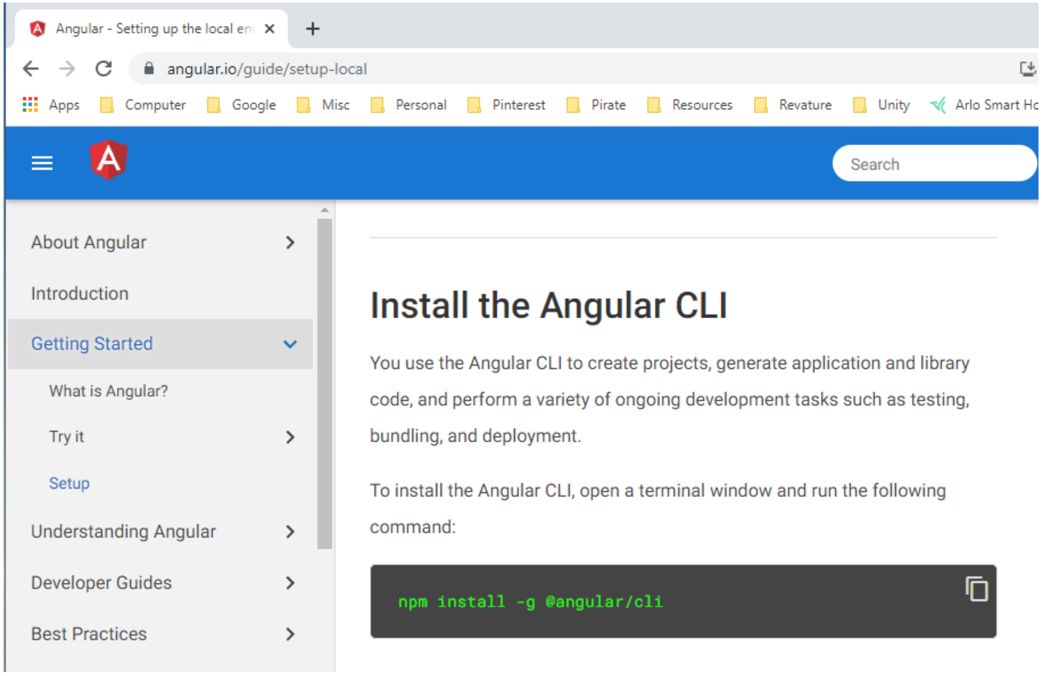
https://angular.io/guide/setup-local



Setting up the local environment and workspace

This guide explains how to set up your environment for Angular development using the Angular CLI tool. It includes information about prerequisites, installing the CLI, creating an initial workspace and starter app, and running that app locally to verify your setup

Scroll down the page



Install the Angular CLI

You use the Angular CLI to create projects, generate application and library code, and perform a variety of ongoing development tasks such as testing, bundling, and deployment.

To install the Angular CLI, open a terminal window and run the following command:

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

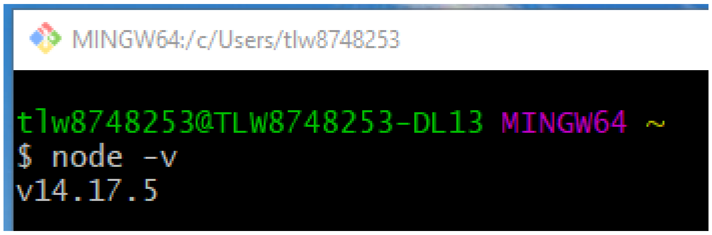
Open Git Bash

From a previous lecture, we should have installed node.js

(if it is not installed go to https://nodejs.org/en/ and download for your environment).

Verify node.js is on your system

node -v



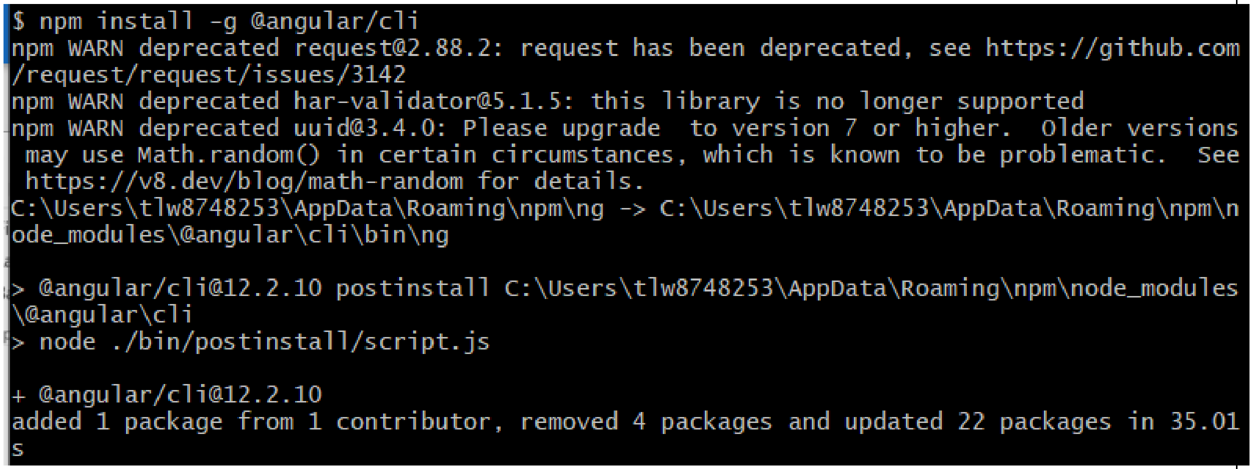
```
tlw8748253@TLW8748253-DL13 MINGW64 ~
$ node -v
v14.17.5
```

With node.js installed, it will come with npm (node package manager).

To install angular globally on your machine type the following in Git Bash

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

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



```
$ npm install -g @angular/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com
/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions
 may use Math.random() in certain circumstances, which is known to be problematic.  See
 https://v8.dev/blog/math-random for details.
C:\Users\tlw8748253\AppData\Roaming\npm\ng -> C:\Users\tlw8748253\AppData\Roaming\npm\n
ode_modules\@angular\cli\bin\ng

> @angular/cli@12.2.10 postinstall C:\Users\tlw8748253\AppData\Roaming\npm\node_modules
\@angular\cli
> node ./bin/postinstall/script.js

+ @angular/cli@12.2.10
added 1 package from 1 contributor, removed 4 packages and updated 22 packages in 35.01
s
```

After installation proceed to next step.

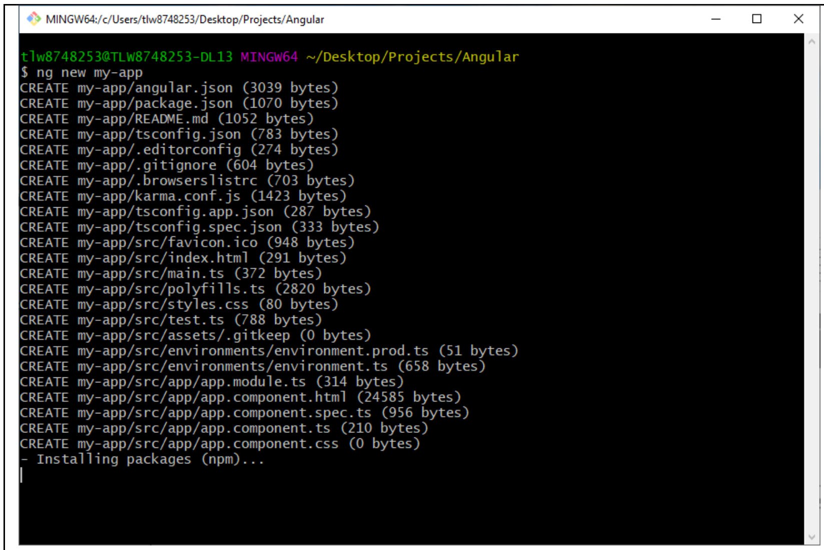
Create a file folder to hold your project, then reopen Git Bash in the folder you created, then:

Create a workspace and initial application

```
ng new my-app
```

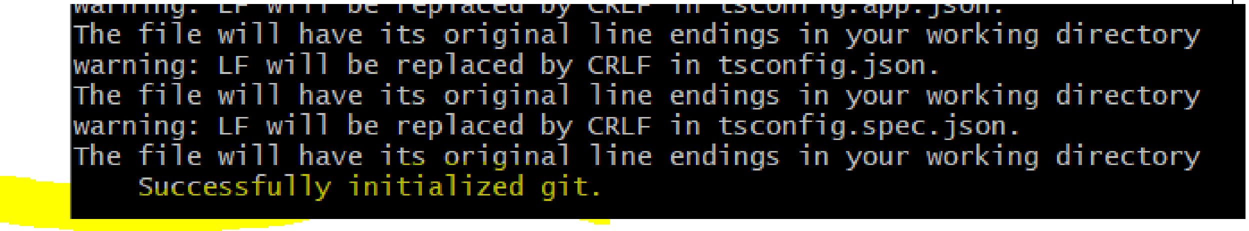
Where my-app is the name of your project.

This will create a minimum shell of an Angular project.



MINGW64:/c/Users/tlw8748253/Desktop/Projects/Angular

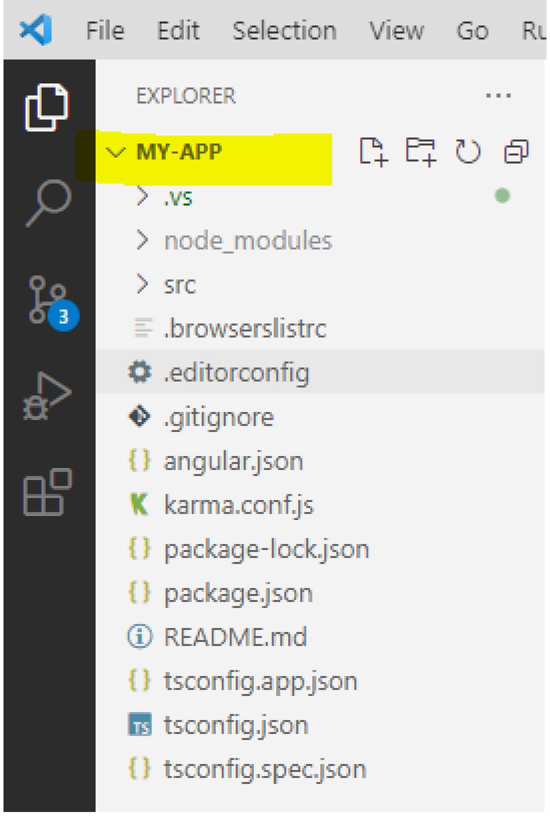
```
tlw8748253@TLW8748253-DL13 MINGW64 ~/Desktop/Projects/Angular
$ ng new my-app
CREATE my-app/angular.json (3039 bytes)
CREATE my-app/package.json (1070 bytes)
CREATE my-app/README.md (1052 bytes)
CREATE my-app/tsconfig.json (783 bytes)
CREATE my-app/.editorconfig (274 bytes)
CREATE my-app/.gitignore (604 bytes)
CREATE my-app/.browserslistrc (703 bytes)
CREATE my-app/karma.conf.js (1423 bytes)
CREATE my-app/tsconfig.app.json (287 bytes)
CREATE my-app/tsconfig.spec.json (333 bytes)
CREATE my-app/src/favicon.ico (948 bytes)
CREATE my-app/src/index.html (291 bytes)
CREATE my-app/src/main.ts (372 bytes)
CREATE my-app/src/polyfills.ts (2820 bytes)
CREATE my-app/src/styles.css (80 bytes)
CREATE my-app/src/test.ts (788 bytes)
CREATE my-app/src/assets/.gitkeep (0 bytes)
CREATE my-app/src/environments/environment.prod.ts (51 bytes)
CREATE my-app/src/environments/environment.ts (658 bytes)
CREATE my-app/src/app/app.module.ts (314 bytes)
CREATE my-app/src/app/app.component.html (24585 bytes)
CREATE my-app/src/app/app.component.spec.ts (956 bytes)
CREATE my-app/src/app/app.component.ts (210 bytes)
CREATE my-app/src/app/app.component.css (0 bytes)
- Installing packages (npm)...
```



```
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
    Successfully initialized git.
```

Wait for the project installation to complete

You can the project with VS-Code



- **package.json** - used to configure npm package dependencies that are available to all projects in the workspace.

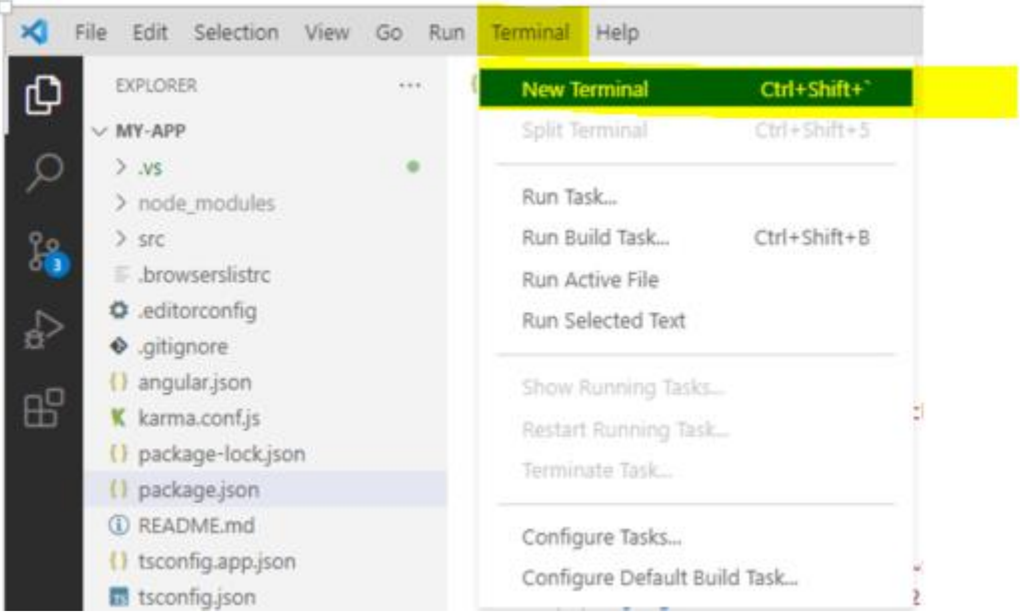
Is similar to the Project Object Model (POM) pom.xml in Maven Java projects.

package.json – dependencies vs. devDependencies.
dependencies are what the final website needs where devDependencies are used during development.

Important scripts to start and build the Angular application:

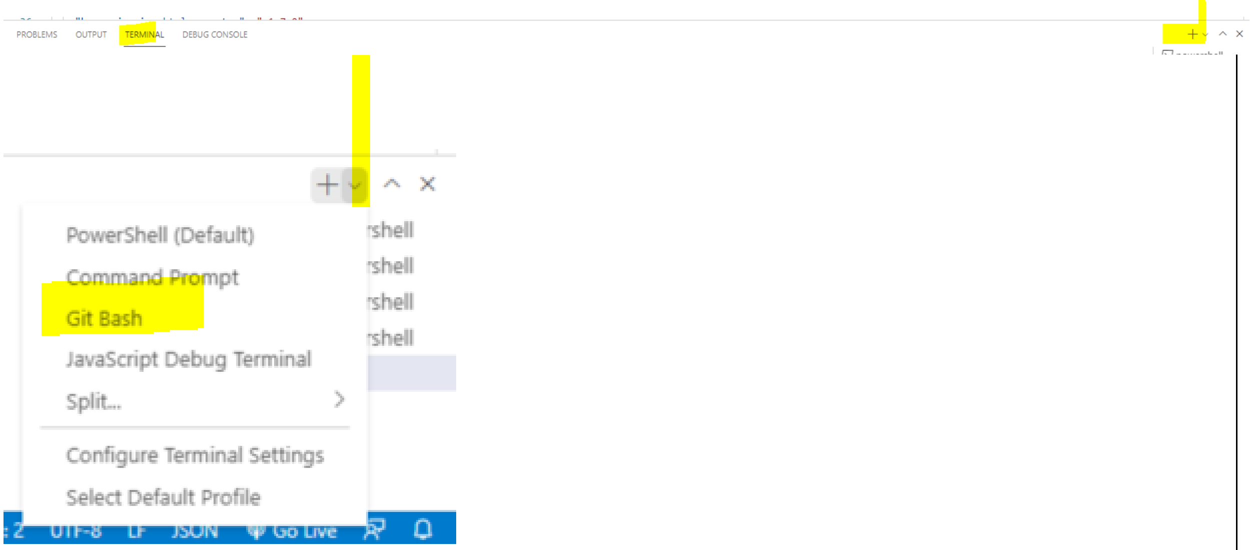
```
   "start": "ng serve",
    "build": "ng build",
```

More information on Angular files.



Will open a Git Bash terminal or PowerShell depending on your VSCode setup

You can change the terminal to Git Bash in the terminal window by clicking the down arrow next to the plus sign.



Type

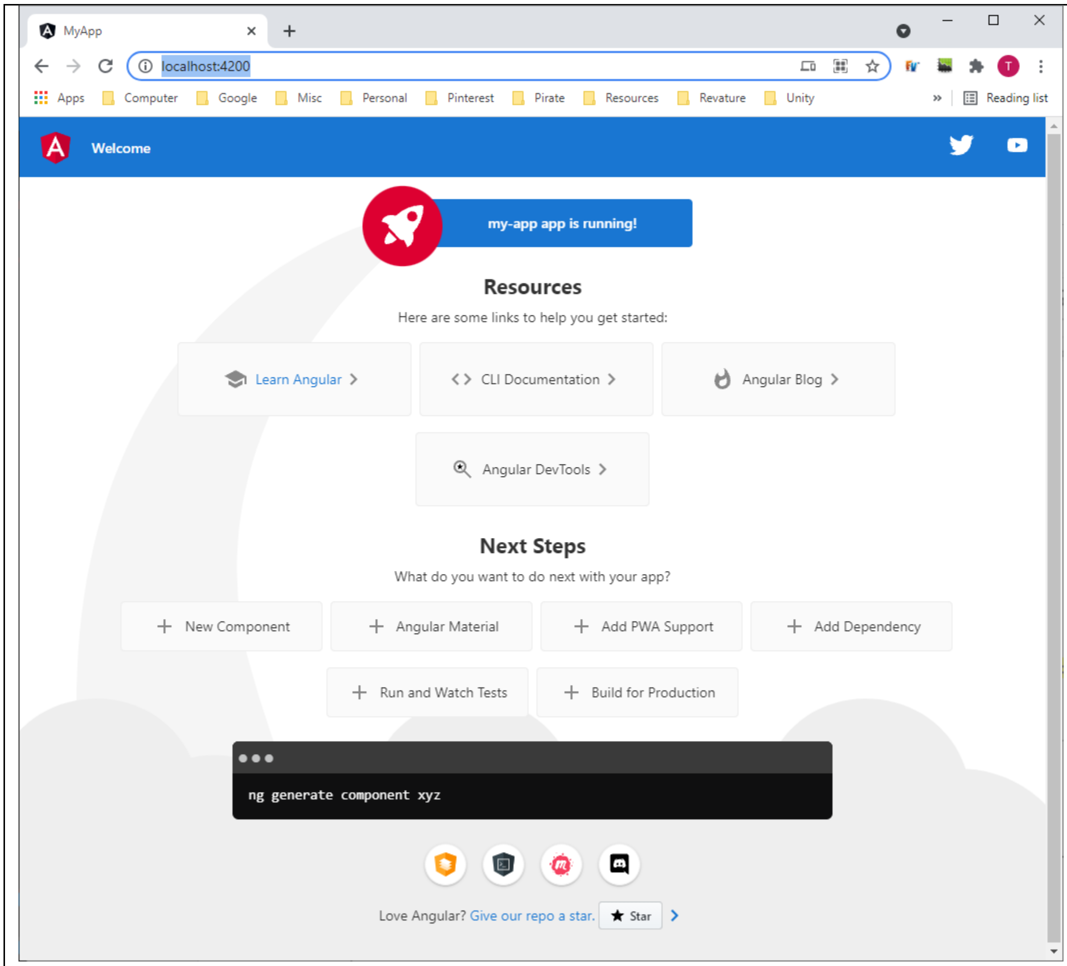
```
npm run start
```

Which runs the ng serve command defined in the package.json file.

```
tlw8748253@TLW8748253-DL13 MINGW64 ~/Desktop/Projects/Angular/my-app (master)
$ npm run start
```

```
> my-app@0.0.0 start C:\Users\tlw8748253\Desktop\Projects\Angular\my-app
> ng serve

✓ Browser application bundle generation complete.

Initial Chunk Files   | Names         |      Size
vendor.js             | vendor        |   2.09 MB
polyfills.js          | polyfills     | 510.57 kB
styles.css, styles.js | styles        | 383.36 kB
main.js               | main          |  55.02 kB
runtime.js            | runtime       |   6.61 kB

                      | Initial Total |   3.02 MB

Build at: 2021-10-19T19:47:14.313Z - Hash: 3cf238b133240949fa8b - Time: 8155ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **


√ Compiled successfully.
```

Type in the address bar of a browser:
http://localhost:4200/
To open the application shell that was just compiled.



my-app app is running!

Resources

Here are some links to help you get started:

Next Steps

What do you want to do next with your app?

```
ng generate component xyz
```

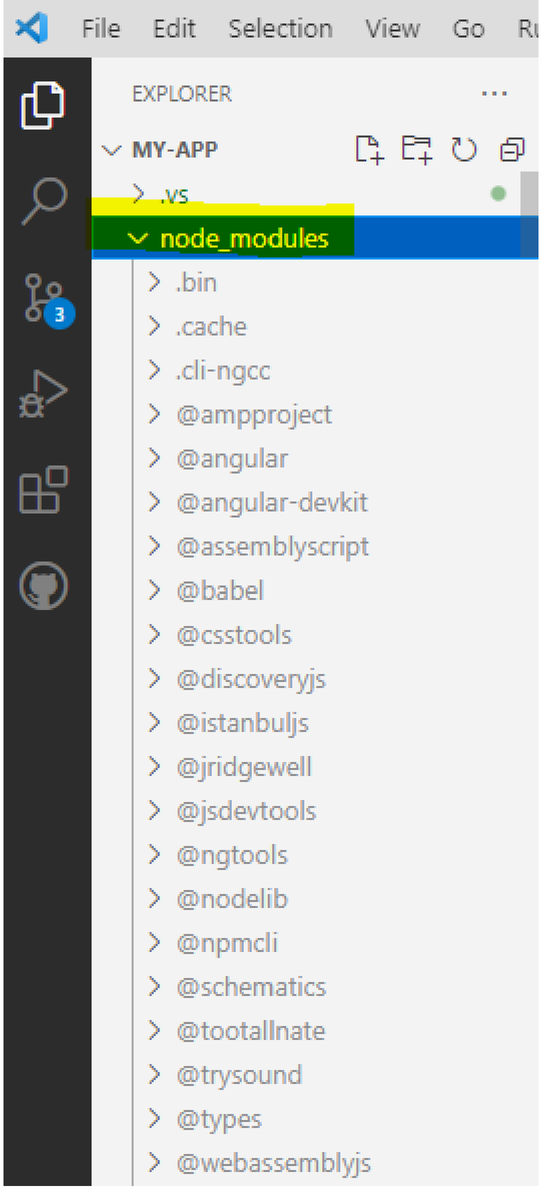
Love Angular? Give our repo a star.

The window above shows your application is working.

Other Angular files:

karma.conf.js – configuration for unit testing similar to JUnit test.
tsconfig.json – type script configuration trans-complier into JavaScript.

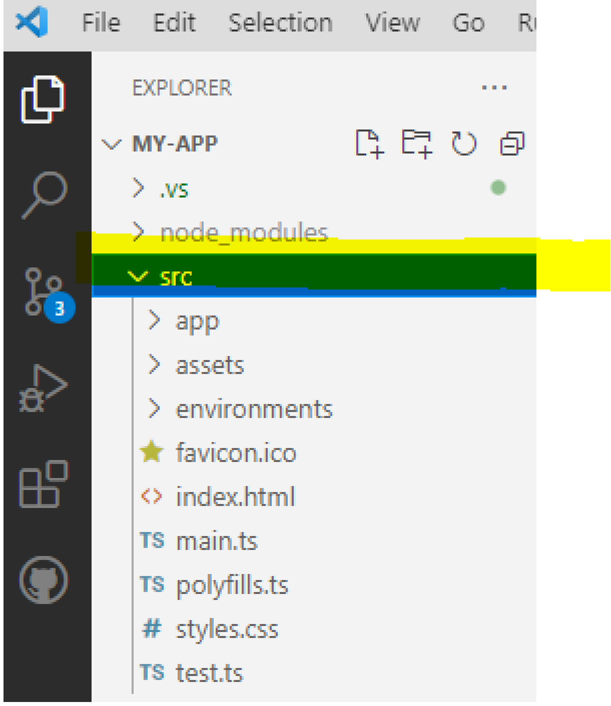
node_modules folder – contains all the dependencies for your project



MY-APP
- .vs
- node_modules
  - .bin
  - .cache
  - .cli-ngcc
  - @ampproject
  - @angular
  - @angular-devkit
  - @assemblyscript
  - @babel
  - @csstools
  - @discoveryjs
  - @istanbuljs
  - @jridgewell
  - @jsdevtools
  - @ngtools
  - @nodelib
  - @npmcli
  - @schematics
  - @tootallnate
  - @trysound
  - @types
  - @webassemblyjs

https://www.npmjs.com/
Website to find information about the dependencies in the node_modules folder.
The site is where the dependencies are downloaded from.

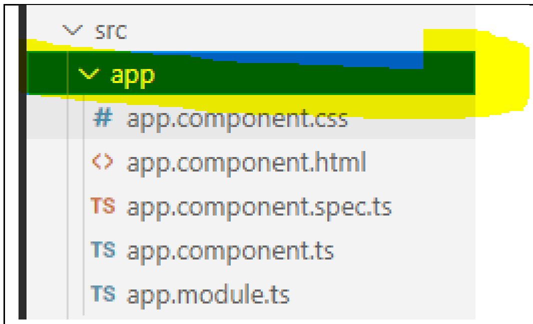
Source folder:



favicon.ico – icon seen on the tab of the browser window.
index.html – initial landing page.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>MyApp</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

<app-root></app-root> refers to a component, in this case the app folder:



```
src
  app
    app.component.css
    app.component.html
    app.component.spec.ts
    app.component.ts
    app.module.ts
```

Which contains the html, css, and script for the project page.

Base model is the app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

bootstrap: [AppComponent] - important element inside @NgModule decorator. Startup process scanning for components that make up your program like <app-root> tag what it refers to.

Even though we see multiple .html source files it is still a single page application when package for runtime. Once the application is built it will reside in a dist\my-app folder for distribution to the runtime environment. Where you will see only the index.html and support file.

To build the file use the following command in the terminal window:

```
npm run build
```

```
$ npm run build

> my-app@0.0.0 build C:\Users\tlw8748253\Desktop\Projects\Angular\my-app
> ng build

✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files              | Names         |      Size
main.e2ae13337439c05d49e0.js      | main          | 134.15 kB
polyfills.fe861a01c9204748df8e.js | polyfills     |  36.19 kB
runtime.47c71d9e559014d0f763.js   | runtime       |   1.02 kB
styles.31d6cfe0d16ae931b73c.css   | styles        |   0 bytes

                                  | Initial Total | 171.35 kB

Build at: 2021-10-19T20:39:58.118Z - Hash: 4bdf9965d757cc7f688f - Time: 23212ms
```

Then you can run the index.html with VSCode using live server.

Recommended to go through the tour of heroes tutorial for more information:
https://angular.io/tutorial

Appendix: Describes Angular files from JwA calendar Aug 23rd 2021.

https://app.revature.com/curriculum/8105/batch/994/viewCalendar

Angular CLI

Angular CLI

The Angular CLI is a command-line interface for Angular that helps us to get started with creating an Angular application. Angular CLI creates an Angular application and uses the Typescript programming language, Webpack for Module bundling, Karma for unit testing, and Protractor for end-to-end testing. The Angular CLI takes care of the configuration and initialization of various libraries. It also allows us to add components, directives, services, etc, to already existing Angular applications.

Installing Angular CLI

Before installing Angular CLI, make sure the development environment includes Node.js and an npm package manager.

- Run the command `npm install -g @angular/cli` on the terminal to install the Angular CLI using npm.
- Run the CLI command `ng new my-app` to create a new angular app with the `my-app` name.
- The Angular CLI includes a server so that we can easily build and serve your app locally. First, go to the `my-app` workspace folder and Launch the server by using the CLI command `ng serve --open`.

The ng serve command launches the server on HTTP port 4200, which watches our files and rebuilds the app as we make changes to those files. The --open (or just -o) option automatically opens the browser to http://localhost:4200.

After running the `ng server -o` command, we will see:

Angular File Structure

Generally, We use Visual Studio Code or Webstrom as a Code Editor for creating and editing Angular Applications. You can download and install Visual Studio Code from this website: https://code.visualstudio.com/download

The file structure of the Angular application described below:

The **e2e** folder at the top level contains source files for a set of end-to-end tests and test-specific configuration files. The **node_modules** folder provides npm packages to the entire workspace. The **src** folder contains the source files which give information about application logic, data, and assets.

- **app** - this folder contains the component files.
  - **app.component.ts** - used to define the logic for the app's root component (AppComponent).
  - **app.component.html** - used to define the HTML template associated with the root AppComponent.

    - **app.component.css** - used to define the base CSS stylesheet for the root AppComponent.
    - **app.component.spec.ts** - used to define the unit test for the root AppComponent.
    - **app.module.ts** - used to define the root module (AppModule) and helps the Angular to assemble the application. All components, including the AppComponent, must be declared inside the AppModule.
- **assets** - this folder contains image and other asset files.
- **environments** - this folder contains build configuration options for particular target environments.
- **favicon.ico** - An icon to used for an application in the bookmark bar.
- **index.html** - The main HTML page that is served when someone visits your site. The CLI automatically adds all JavaScript and CSS files when building your app, so you typically don't need to add any `<script>` or `<link>` tags here manually.
- **main.ts** - The main entry point for an application. Compiles the application with the JIT compiler and bootstraps the application's root module (AppModule) to run in the browser.
- **polyfills.ts** - Provides polyfill scripts for browser support.
- **styles.css** - Lists CSS files that applies the styles for a project.
- **test.ts** - The main entry point for unit tests used in the application.
- **.editorconfig** - this file contains configuration for code editors.
- **.gitignore** - it specifies untracked files that Git should ignore.
- **angular.json** - holds CLI configuration defaults for all projects in the workspace. It includes configuration options for the build, serve, and test tools.
- **browserslist** - used to configure the sharing of target browsers and Node.js versions among various front-end tools.
- **karma.conf.js** - it contains application-specific Karma configuration.
- **package-lock.json** - this provides version information for all packages installed into node_modules by the npm client.
- **package.json** - used to configure npm package dependencies that are available to all projects in the workspace.
- **README.md** - An introductory documentation for the root app.
- **tsconfig.app.json** - it holds application-specific TypeScript configuration, including TypeScript and Angular template compiler options.
- **tsconfig.json** - holds default TypeScript configuration for projects in the workspace.
- **tslint.json** - holds default TSLint configuration for projects in the workspace. TSLint is an extensible static analysis tool that checks TypeScript code for readability, maintainability, and functionality errors.

References

- Angular Docs - CLI Overview and Command Reference

- Angular Docs - Workspace and project file structure