

Files and Folders in an Angular project

These files and folders are created and populated when creating a new Angular application using the `ng new` command. The `.ts` extension indicates a TypeScript file.

node_modules: Contains the installed dependencies;
Must exist for `ng serve` command to work;
May be created by `npm install`, if missing
This file is quite large and is typically not saved to GitHub;
(should be included in `.gitignore` file)
Not included in production deploy

src: Folder containing source code for the application;
Angular components, services, and other code that makes up the application;
Contains the **app** folder

app: Folder containing main Angular application code; Files in this folder:

app.component.css: CSS for your app component

app.component.html: Html for the app component;
Template file todo the data binding

app.component.spec.ts: Unit testing file for the app component

app.component.ts: Defines the root component of the application;
Most important typescript file;
Includes the view logic behind the component.

app.config.server.ts: Configure a Node.js server;
Responsible for bootstrapping the Angular application and
Providing it with the necessary configuration

app.config.ts: Tell Angular how to assemble the application;
Defines the application's root component - entry point for the application;
Defines the application's providers - services used in the application

app.routes.ts: Routing configuration for the application;
Defines all the possible routes in the application; specifies which
component should be displayed for each route.

app.module.ts: Configures the application modules;
All the dependencies for the website;
Used to define the needed modules to be imported, the components to
be declared and the main component to be bootstrapped
No longer required after Angular 17

Files and Folders in an Angular project

assets: Folder containing static assets, such as images, fonts, and CSS files;
Assets are typically referenced in the application's HTML and CSS files

favicon.ico: Small icon that is displayed in the browser's tab bar;
Typically a 16x16 pixel image, but can be up to 32x32 pixels

assets: Folder containing static assets, such as images, fonts, and CSS files;
Assets are typically referenced in the application's HTML and CSS files

favicon.ico: Small icon that is displayed in the browser's tab bar;
Typically a 16x16 pixel image, but can be up to 32x32 pixels

index.html: Primary HTML file for the application;
The basic structure of the application;
Loads the Angular application code.

favicon.ico: Small icon that is displayed in the browser's tab bar;
Typically a 16x16 pixel image, but can be up to 32x32 pixels

main.server.ts: Entry point for server-side rendering (SSR);
Responsible for configuring and starting the server-side rendering process

main.ts: Entry point for an Angular application;
Responsible for bootstrapping the application and loading the root module

styles.css: CSS styling to be applied to all pages in an application

angular.json: Configuration options for the Angular CLI;
i.e Configure the build process, the development server, and other aspects of the Angular CLI.

package-lock.json: Lock down the versions of your dependencies so that a project will always use the same versions, regardless of when they were installed

package.json: Information about the project's dependencies;
Used by the Angular CLI to install/manage dependencies, and to run scripts.

README.md: Contains description of application, directions on use and any other information to be conveyed to a user of the application.

server.ts: Used to run an Angular application on a server;
Responsible for starting the Node.js server/configure it to serve the Angular app;
May also include code to handle routing and to provide access to data in a database.

Files and Folders in an Angular project

tsconfig.json: Specifies the TypeScript compiler options for the project;
Control how the TypeScript compiler transpiles TypeScript into JavaScript

tsconfig.app.json: Additional TypeScript compiler options for the project;
Extends/Adds to the options in **tsconfig.json**

tsconfig.spec.json: Additional TypeScript compiler options for the project;
Extends/Adds to the options in tsconfig.json

environments: Folder containing environment-specific configuration files;
Used to configure different aspects of the application ie. API endpoint URLs, for different environments (*development, staging, and production*)
Not required for Angular application to run