**Angular Project Structure With Generated Files Description**

When you create an Angular project using Angular CLI **ng new** command, the CLI installs the necessary Angular npm packages and other dependencies in a new workspace. The workspace root folder contains various support and configuration files, and a README file with generated descriptive text that you can customize. This article gives an overview of the generated files with some explanation about those files.

**File groups**

The generated files for the Angular project can be divided into following groups:

1. [Workspace configuration files](https://www.netjstech.com/2020/03/angular-project-structure-with-file-description.html#WorkspaceConfig)
2. [Application source files](https://www.netjstech.com/2020/03/angular-project-structure-with-file-description.html#ApplicationSource)
3. [Application configuration files](https://www.netjstech.com/2020/03/angular-project-structure-with-file-description.html#ApplicationConfig)
4. [End-to-end test files](https://www.netjstech.com/2020/03/angular-project-structure-with-file-description.html#E2ETest)

**Workspace configuration files**

The top level of the workspace contains workspace-wide configuration files, configuration files for the root-level application, and subfolders for the root-level application source and test files.

|  |  |
| --- | --- |
| **WORKSPACE CONFIG FILES** | **PURPOSE** |
| .editorconfig | Configuration for code editors so that consistent coding styles is maintained for multiple developers working on the same project across various editors and IDEs |
| .gitignore | Specifies intentionally untracked files that GIT should ignore. |
| README.md | Introductory documentation for the root app. |
| angular.json | CLI configuration defaults for all projects in the workspace, including configuration options for build, serve, and test tools that the CLI uses, such as TSLint, Karma and Protractor. |
| package.json | Configures npm package dependencies that are available to all projects in the workspace. |
| package-lock.json | Provides version information for all packages installed into node\_modules by the npm client. |
| src/ | Source files for the root-level application project. |
| node\_modules/ | Provides npm packages to the entire workspace. Workspace-wide node\_modules dependencies are visible to all projects. |
| tsconfig.json | Default TypeScript configuration for projects in the workspace. |
| tslint.json | Default TSLint configuration for projects in the workspace. |
| e2e/ | Contains source files for a set of end-to-end tests. |

**Application source files**

Files at the top level of src/ support testing and running your application. Subfolders contain the application source and application-specific configuration.

|  |  |
| --- | --- |
| **APP SUPPORT FILES** | **PURPOSE** |
| app/ | Contains the component files in which your application logic and data are defined. |
| assets/ | Contains image and other asset files to be copied as-is when you build your application. |
| environments/ | Contains build configuration options for particular target environments. By default there is an unnamed standard development environment and a production ("prod") environment. You can define additional target environment configurations. |
| favicon.ico | An icon to use for this 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 your application. Compiles the application and bootstraps the application's root module (AppModule) to run in the browser. |
| polyfills.ts | Provides polyfill scripts for browser support. |
| styles.sass | Lists CSS files that supply styles for a project. The extension reflects the style preprocessor you have configured for the project. |
| test.ts | The main entry point for your unit tests, with some Angular-specific configuration. You don't typically need to edit this file. |

The app/ folder, inside the /src folder, contains your project's logic and data. Angular components, templates, and styles go here.

|  |  |
| --- | --- |
| **SRC/APP/FILES** | **PURPOSE** |
| app/app.component.ts | Defines the logic for the app's root component, named AppComponent. The view associated with this root component becomes the root of the view hierarchy as you add components and services to your application. |
| app/app.component.html | Defines the HTML template associated with the root AppComponent. |
| app/app.component.css | Defines the base CSS stylesheet for the root AppComponent. |
| app/app.component.spec.ts | Defines a unit test for the root AppComponent. |
| app/app.module.ts | Defines the root module, named AppModule, that tells Angular how to assemble the application. Initially declares only the AppComponent. As you add more components to the app, they must be declared here. |

**Application configuration files**

The application-specific configuration files for the root application reside at the workspace root level.

|  |  |
| --- | --- |
| **APPLICATION-SPECIFIC CONFIG FILES** | **PURPOSE** |
| browserslist | Configures sharing of target browsers and Node.js versions among various front-end tools. |
| karma.conf.js | Application-specific Karma configuration. |
| tsconfig.app.json | Application-specific TypeScript configuration, including TypeScript and Angular template compiler options. |
| tsconfig.spec.json | TypeScript configuration for the application tests. |
| tslint.json | Application-specific TSLint configuration. |

**End-to-end test files**

An e2e/ folder at the top level contains source files for a set of end-to-end tests that correspond to the root-level application, along with test-specific configuration files.

|  |  |
| --- | --- |
| **END-TO-END TEST FILES** | **PURPOSE** |
| src/ | End-to-end tests for the application. |
| protractor.conf.js | Test-tool config |
| tsconfig.json | TypeScript config inherits from workspace |