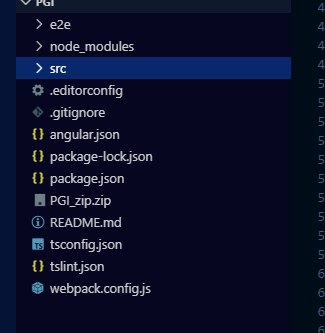
**PGI Project Structure**

**Intoduction**

The following figure shows the basic structure of an angular project:



All projects within a workspace share a CLI configuration context. 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.

Below is the description of all these files:

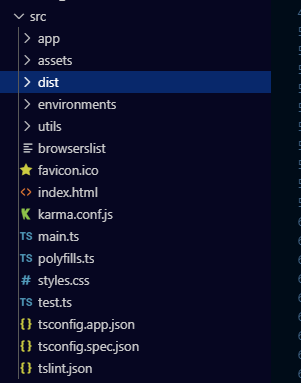
|  |  |
| --- | --- |
| **WORKSPACE CONFIG FILES** | **PURPOSE** |
| .editorconfig | Configuration for code editors. See EditorConfig. |
| .gitignore | Specifies intentionally untracked files that Git should ignore. |
| README.md | Introductory documentation for the root app. |
| angular.json | CLI configuration 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. For details, see Angular Workspace Configuration. |
| package.json | Configures npm package dependencies that are available to all projects in the workspace. See npm documentation for the specific format and contents of this file. |
| package-lock.json | Provides version information for all packages installed into node\_modules by the npm client. See npm documentation for details. If you use the yarn client, this file will be yarn.lock instead. |
| 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/ | 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. |

**Application project files**

By default, the CLI command ng new PGI creates a workspace folder named "PGI" and generates a new application skeleton in a src/ folder at the top level of the workspace. A newly generated application contains source files for a root module, with a root component and template.

### 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.

Below is the structure and explanation for the same:

|  |  |
| --- | --- |
| **APP SUPPORT FILES** | **PURPOSE** |
| app/ | Contains the component files in which your application logic and data are defined. See details below. |
| 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 with the JIT compiler and bootstraps the application's root module (AppModule) to run in the browser. You can also use the AOT compiler without changing any code by appending the --aot flag to the CLI build and serve commands. |
| 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. |

Application configuration files:

The application-specific configuration files for the root application reside at the workspace root level. For a multi-project workspace, project-specific configuration files are in the project root, under projects/project-name/.

Project-specific [TypeScript](https://www.typescriptlang.org/) configuration files inherit from the workspace-wide tsconfig.json, and project-specific [TSLint](https://palantir.github.io/tslint/) configuration files inherit from the workspace-wide tslint.json.

|  |  |
| --- | --- |
| **APPLICATION-SPECIFIC CONFIG FILES** | **PURPOSE** |
| browserslist | Configures sharing of target browsers and Node.js versions among various front-end tools. See Browserslist on GitHub for more information. |
| 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. |