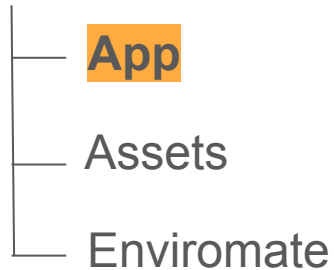


Angular Cli / Module & View

Application source files

For a single-application workspace, the `src/` subfolder of the workspace contains the source files (application logic, data, and assets) for the root application. For a multi-project workspace, additional projects in the `projects/` folder contain a `project-name/src/` subfolder with the same structure.

Src



NgModule

Every Angular app has at least one NgModule class, **the root module**, which is conventionally named `AppModule` and resides in a file named `app.module.ts`. You launch your app by *bootstrapping* the root NgModule.

While a small application might have only one NgModule, most apps have many more *feature modules*. The *root* NgModule for an app is so named because it can include child NgModules in a hierarchy of any depth.

Meta Data

An NgModule is defined by a class decorated with `@NgModule()`. The `@NgModule()` decorator is a function that takes a single metadata object, whose properties describe the module. The most important properties are as follows.

- **declarations**: The components, *directives*, and *pipes* that belong to this NgModule.
- **exports**: The subset of declarations that should be visible and usable in the *component templates* of other NgModules.
- **imports**: Other modules whose exported classes are needed by component templates declared in *this* NgModule.
- **providers**: Creators of services, that this NgModule contributes to the global collection of services; they become accessible in all parts of the app. (You can also specify providers at the component level.)
- **bootstrap**: The main application view, called the *root component*, which hosts all other app views. Only the *root NgModule* should set the `bootstrap` property.

components and templates

Every Angular application has at least one component, the *root component* that connects a component hierarchy with the page document object model (DOM). Each component defines a class that contains application data and logic, and is associated with an HTML *template* that defines a view to be displayed in a target environment.

The `@Component()` decorator identifies the class immediately below it as a component, and provides the template and related component-specific metadata.

- **Selector** : The CSS selector that identifies this directive in a template and triggers instantiation of the directive
- **TemplateUrl** : The relative path or absolute URL of a template file for an Angular component. If provided, do not supply an inline template using `template`.