JavaScript - Frameworks - Angular - Application Structure

# Directory Structure

The base directory structure for a standard Angular application is genereated using the ng new <project-name> command. This structure is suitable for mutli-repo design where each application is in its own repo. It is also possible to have multiple projects in a repo, and monorepo styles of projects.

### Folder Structure

The src subfolder contain the source files for the root application. It contains the following folders:

* app/ - component files for application logic
* assets/ - contains images and other assets
* environments/ - contains build config for particular target environments

### Projects and Libraries

Additional projects and libraries go in the /projects folder.

# Project Configuration

The angular.json file provides workspace-wide project configuration. The properties at the top level configure the workspace:

* version - The configuration-file version.
* newProjectRoot - Path where new projects are created. Absolute or relative to the workspace folder.
* defaultProject - Default project name to use in commands, where not provided as an argument. When you use ng new to create a new app in a new workspace, that app is the default project for the workspace until you change it here.
* schematics - A set of schematics that customize the ng generate sub-command option defaults for this workspace. See Generation schematics below.
* projects - Contains a subsection for each project (library or application) in the workspace, with the per-project configuration options.

## Project Config

The project config is built up of:

* root - The root folder for this project's files, relative to the workspace folder. Empty for the initial app, which resides at the top level of the workspace.
* sourceRoot - The root folder for this project's source files.
* projectType - One of "application" or "library". An application can run independently in a browser, while a library cannot.
* prefix- A string that Angular prepends to generated selectors. Can be customized to identify an app or feature area.
* schematics -A set of schematics that customize the ng generate sub-command option defaults for this project. See Generation schematics below.
* architect - Configuration defaults for Architect builder targets for this project.

## Strict Mode

Strict mode can be used to enforce better coding style.

## Environments

The src/environements files contain environment specific configuration files, however, they require replacing on build. Set the replacement in the angular.json file, under:

projects.<project-name>. architect.build.configurations.<environment>.fileReplacements

E.g.:

"configurations": {

"production": {

"fileReplacements": [

{

"replace": "src/environments/environment.ts",

"with": "src/environments/environment.prod.ts"

}

],

Then build using the --configuration=<env-name> flag

## Browser Compatibiity

Angular used Autoprefixer to provider browser compatibliy. Browserslist can be used to define the level of compatiblity in the .browerslistrc.

<https://github.com/browserslist/browserslist>

For example:

### Supported Browsers

> 1%

last 2 versions

To support older browser the target may require changing from es2015 to es5 in the tsconfig.json target

Angular uses polyfills to aid compatibility with older browsers. Angular 8 uses differential loading to reduce file downloads when polyfils are not required.

Import optional polyfils in src/polyfills.ts

<https://angular.io/guide/browser-support>