Strict mode

Contents

When you create a new workspace or a project you have an option to create them in a strict mode using the `--strict` flag.

Enabling this flag initializes your new workspace or project with a few new settings that improve maintainability, help you catch bugs ahead of time, and allow the CLI to perform advanced optimizations on your application. Additionally, applications that use these stricter settings are easier to statically analyze, which can help the `ng update` command refactor code more safely and precisely when you are updating to future versions of Angular.

Specifically, the `strict` flag does the following:

- Enables `strict` mode in TypeScript, as well as other strictness flags recommended by the TypeScript team. Specifically, `forceConsistentCasingInFileNames`, `noImplicitReturns`, `noFallthroughCasesInSwitch`.
- Turns on strict Angular compiler flags `strictTemplates` and `strictInjectionParameters`
- Bundle size budgets have been reduced by ~75%
- Turns on `no-any` tslint rule to prevent declarations of type `any`
- Marks your application as side-effect free to enable more advanced tree-shaking

You can apply these settings at the workspace and project level.

To create a new workspace and application using the strict mode, run the following command:

```
ng new [project-name] --strict
```

To create a new application in the strict mode within an existing non-strict workspace, run the following command:

```
ng generate application [project-name] --strict
```

Non-local side effects in applications

When you create projects and workspaces using the `strict` mode, you'll notice an additional `package.json` file, located in `src/app/` directory. This file informs tools and bundlers that the code under this directory is free of non-local side effects. Non-local side effects in the application code are not common and using them is not considered a good coding pattern. More importantly, code with these types of side effects cannot be optimized, resulting in increased bundle sizes and applications that load more slowly.

If you need more information, the following links may be helpful.

- Tree-shaking
- Dealing with side effects and pure functions in JavaScript
- How to deal with dirty side effects in your pure function JavaScript

RESOURCES

- About
- Resource Listing
- Press Kit
- Blog
- Usage Analytics

HELP

- Stack Overflow
- Gitter
- Report Issues
- Code of Conduct

COMMUNITY

- Events
- Meetups
- Twitter
- GitHub
- Contribute

LANGUAGES

- 简体中文版
- 正體中文版
- 日本語版
- 한국어

Version 10.0.10-local+sha.84d1ba792b.