

## CLI Overview and Command Reference



The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.

### Installing Angular CLI

Major versions of Angular CLI follow the supported major version of Angular, but minor versions can be released separately.

Install the CLI using the `npm` package manager:

```
npm install -g @angular/cli
```

For details about changes between versions, and information about updating from previous releases, see the Releases tab on GitHub: <https://github.com/angular/angular-cli/releases>

### Basic workflow

Invoke the tool on the command line through the `ng` executable. Online help is available on the command line. Enter the following to list commands or options for a given command (such as [generate](#)) with a short description.

```
ng help
ng generate --help
```

To create, build, and serve a new, basic Angular project on a development server, go to the parent directory of your new workspace use the following commands:

```
ng new my-first-project
cd my-first-project
ng serve
```

In your browser, open <http://localhost:4200/> to see the new app run. When you use the `ng serve` command to build an app and serve it locally, the server automatically rebuilds the app and reloads the page when you change any of the source files.

When you run `ng new my-first-project` a new folder, named `my-first-project`, will be created in the current working directory. Since you want to be able to create files inside that folder, make sure you have sufficient rights in the current working directory before running the command.

If the current working directory is not the right place for your project, you can change to a more appropriate directory by running `cd <path-to-other-directory>` first.

### Workspaces and project files

The `ng new` command creates an *Angular workspace* folder and generates a new app skeleton. A workspace can contain multiple apps and libraries. The initial app created by the `ng new` command is at the top level of the workspace. When you generate an additional app or library in a workspace, it goes into a `projects/` subfolder.

A newly generated app contains the source files for a root module, with a root component and template. Each app has a `src` folder that contains the logic, data, and assets.

You can edit the generated files directly, or add to and modify them using CLI commands. Use the `ng generate` command to add new files for additional components and services, and code for new pipes, directives, and so on. Commands such as [add](#) and [generate](#), which create or operate on apps and libraries, must be executed from within a workspace or project folder.

- See more about the [Workspace file structure](#).

### Workspace and project configuration

A single workspace configuration file, `angular.json`, is created at the top level of the workspace. This is where you can set per-project defaults for CLI command options, and specify configurations to use when the CLI builds a project for different targets.

The `ng config` command lets you set and retrieve configuration values from the command line, or you can edit the `angular.json` file directly. Note that option names in the configuration file must use [camelCase](#), while option names supplied to commands can use either camelCase or dash-case.

- See more about [Workspace Configuration](#).
- See the [complete schema](#) for `angular.json`.

### CLI command-language syntax

Command syntax is shown as follows:

`ng commandNameOrAlias requiredArg [optionalArg] [options]`

- Most commands, and some options, have aliases. Aliases are shown in the syntax statement for each command.
- Option names are prefixed with a double dash (`--`). Option aliases are prefixed with a single dash (`-`). Arguments are not prefixed. For example:

```
ng build my-app -c production
```

- Typically, the name of a generated artifact can be given as an argument to the command or specified with the `--name` option.
- Argument and option names can be given in either [camelCase](#) or [dash-case](#). `--myOptionName` is equivalent to `--my-option-name`.

### Boolean and enumerated options

Boolean options have two forms: `--thisOption` sets the flag, `--noThisOption` clears it. If neither option is supplied, the flag remains in its default state, as listed in the reference documentation.

Allowed values are given with each enumerated option description, with the default value in **bold**.

### Relative paths

Options that specify files can be given as absolute paths, or as paths relative to the current working directory, which is generally either the workspace or project root.

### Schematics

The `ng generate` and `ng add` commands take as an argument the artifact or library to be generated or added to the current project. In addition to any general options, each artifact or library defines its own options in a *schematic*. Schematic options are supplied to the command in the same format as immediate command options.

### Command Overview

COMMAND	ALIAS	DESCRIPTION
add		Adds support for an external library to your project.
analytics		Configures the gathering of Angular CLI usage metrics. See <a href="https://angular.io/cli/usage-analytics-gathering">https://angular.io/cli/usage-analytics-gathering</a> .
build	b	Compiles an Angular app into an output directory named <code>dist/</code> at the given output path. Must be executed from within a workspace directory.
config		Retrieves or sets Angular configuration values in the <code>angular.json</code> file for the workspace.
deploy		Invokes the deploy builder for a specified project or for the default project in the workspace.
doc	d	Opens the official Angular documentation ( <a href="https://angular.io">angular.io</a> ) in a browser, and searches for a given keyword.
e2e	e	Builds and serves an Angular app, then runs end-to-end tests using Protractor.
generate	g	Generates and/or modifies files based on a schematic.
help		Lists available commands and their short descriptions.
lint	l	Runs linting tools on Angular app code in a given project folder.
new	n	Creates a new workspace and an initial Angular app.
run		Runs an Architect target with an optional custom builder configuration defined in your project.
serve	s	Builds and serves your app, rebuilding on file changes.
test	t	Runs unit tests in a project.
update		Updates your application and its dependencies. See <a href="https://update.angular.io/">https://update.angular.io/</a>
version	v	Outputs Angular CLI version.
xi18n	i18n-extract	Extracts i18n messages from source code.