# Angular CLI Angular workspace tool

## CONCEPTS

Run ng from anywhere inside a workspace; it finds angular.json upwards. ng new is the only command that works outside one.

**angular.json** projects, targets
↓ ng <command>
**Builder** build, serve, test
↓ runs
**dist/** output or dev server

Every command is a target in angular.json: ng run <project>:<target> is the generic form behind them.

Flags are dash-case on the command line (--output-path) but camelCase in angular.json (outputPath).

## WORKSPACE

`ng new <name>`
Create a new workspace and app

`ng new <name> --routing --style=scss`
Create with routing and SCSS

`ng version`
Show CLI, Angular and package versions

`ng add <package>`
Install a package and run its setup

`ng new <name> --skip-install --skip-git`
Create without install or git init

`ng new <name> --package-manager=pnpm`
Pick the package manager

`ng new <name> --ssr`
Create with server-side rendering

## SERVE AND BUILD

`ng serve`
Start the dev server (port 4200)

`ng serve -o`
Start and open the browser

`ng serve --port <n>`
Listen on another port

## SERVE AND BUILD (cont.)

`ng build`
Build the app into dist/

`ng build -c production`
Build with a named configuration

`ng serve -c <config>`
Serve using a named configuration

`ng serve --host 0.0.0.0`
Listen on all interfaces

`ng serve --proxy-config <file>`
Proxy API calls to a backend

`ng serve --ssl`
Serve over HTTPS

`ng build --watch`
Rebuild on file changes

`ng build --base-href <path>`
Set the base URL for deployment

`ng build --output-path <dir>`
Write output elsewhere

## GENERATE

`ng generate component <name>`
Create a component

`ng generate service <name>`
Create a service

`ng generate directive <name>`
Create a directive

`ng generate pipe <name>`
Create a pipe

`ng generate guard <name>`
Create a route guard

`ng generate interface <name>`
Create an interface

`ng generate component <path/name>`
Create inside a subfolder

`ng generate component <name> --dry-run`
Preview files, write nothing

`ng generate component <name> --skip-tests`
Skip the .spec.ts file

`ng generate component <name> --inline-template --inline-style`
Single-file component

## GENERATE (cont.)

`ng generate interceptor <name>`
Create an HTTP interceptor

## TEST AND LINT

`ng test`
Run unit tests

`ng test --no-watch`
Run once and exit

`ng test --coverage`
Collect a coverage report

`ng test --filter <regex>`
Run tests whose name matches

`ng test --browsers <name>`
Run in a real browser

`ng test --include <glob>`
Limit to matching spec files

`ng test --runner karma`
Use Karma instead of Vitest

`ng lint`
Run the configured linter

`ng add angular-eslint`
Set up ESLint for lint

`ng e2e`
Build, serve and run e2e tests

## MAINTAIN

`ng update`
List packages that can be updated

`ng update @angular/core @angular/cli`
Update Angular and run migrations

`ng update --create-commits`
Commit each update and migration

`ng config <path>`
Read a value from angular.json

`ng config <path> <value>`
Set a value in angular.json

`ng cache info`
Show cache location and size

`ng cache clean`
Delete the build cache

`ng run <project>:<target>`
Run any target from angular.json