

# Getting Started with Angular CLI

## Complete Guide

Update on February 27, 2025

Angular is a **JavaScript framework** .

It is designed for modern web application development.

**Angular CLI ( Command Line Interface )** is the official tool for Angular.

It allows you to create, manage and optimize your projects with ease.

In this guide, we will explore the **fundamental Angular CLI commands** for:

- **Start** an Angular application
- **Structuring** an Angular application
- **Developing** an Angular application



# What we are going to do

- **Before You Begin**

What is Angular CLI

- **The orders**

Selection of the most useful commands for creating and managing an Angular application

- **Uninstall and install**

How to use Angular CLI

---

## What is Angular CLI?

**CLI** stands for **Command Line Interface** .

Angular CLI is a **tool** that allows you to use **command-line commands**.

It allows you **to initialize** , **develop** and **maintain Angular applications**.

The latest versions are available below

- <https://github.com/angular/angular-cli/releases>

The installation procedure is detailed on the official Angular website

<https://angular.io/cli>

You must first install Node.js on your workstation.

Angular CLI is a library (or package).

We will install it with **npm** , the **Node.js** package manager.

```
# Uninstalling angular-cli
npm uninstall -g @angular/cli

# Installing angular-cli latest version available
npm install -g @angular/cli

# Installing angular-cli specific version
npm install -g @angular/cli@15.0.2

# Installed version test
ng --version
```

## List of commands

To know the orders two official links

- <https://github.com/angular/angular-cli/wiki>
- <https://angular.io/cli>

Here is the list of those that we will examine

- **Installing/Uninstalling Angular CLI**  
ng new [project-name]
- **Creating an Angular Project**  
ng new [project-name]
- **Creating a component**  
ng generate component [component-name]
- **Creating a module**  
ng generate module [component-name]
- **Creating a class**  
ng generate class [class-name]

---

## Uninstalling and Installing Angular CLI

These are not Angular CLI commands.

But since they are essential, let's talk about them.

You need to use npm (Node Package Manager) from node.js

```
# Uninstalling angular-cli
npm uninstall -g @angular/cli

# Installing angular-cli latest version available
npm install -g @angular/cli

# Installing angular-cli specific version
npm install -g @angular/cli@15.0.2

# Installed version test
ng --version
```

## Creating an Angular Project

Syntax (2 Methods)

- ng **new** [project-name] [options]
- ng **n** [project-name] [options]

**new** or **n** : Angular CLI command (application creation)

**project-name** : Name of the Angular project (the name is arbitrary)

**options** : Adapts the order

Action

- Creates a new **Workspace** and a new Angular application.

Options

- routing

Allows you to manage routing

Automatically creates an app-routing.module.ts file

- defaults

Allows you to automatically select default options

(no routing, CSS type)

```
# Generate a project called angular-starter with manual choice of options
ng new angular-starter

# Generate a project called angular-starter with default options
ng new angular-starter --defaults

# Generate a project called angular-starter with Routing Management
ng new angular-starter --routing
```

## Creating a component

Syntax (2 methods)

- **ng generate component [component-name] [options]**
- **ng gc [component-name] [options]**

**generate** or **g** : Angular CLI command

**component** or **c** : creating a component

**component-name** : Name of the Angular component (the name is arbitrary)

**options** : allows you to adapt component creation

Action

- Creates a component at the specified location

Examples

- **ng generate component home**

Create a home directory in app

Creates 4 files in the app/home directory

- home.component.css
- home.component.html
- home.component.spec.ts
- home.component.ts

Modify the app.module.ts file

- added HomeComponent in declarations

- **ng generate component components/contact**

Create a components directory in app

Creates a contact directory in app/components

Creates 4 files in the app/components/contact directory

- contact.component.css
- contact.component.html
- contact.component.spec.ts
- contact.component.ts

Modify the app.module.ts file

- added ContactComponent in declarations

```
# Creating a home component in app
ng generate component home

# Creating a contact component in the app/modules directory
ng generate component modules/contact
```

## Creating a module

Syntax (2 methods)

- **ng generate module [module-name] [options]**
- **ng gm [module-name] [options]**

**generate** or **g** : Angular CLI command

**module** or **m** : creating a module

**module-name** : Name of the Angular module (the name is arbitrary)

**options** : allowing you to adapt module creation

Examples

- **ng generate module home**  
Create a home directory in app  
Creates 1 file in the app/home directory
  - home.module.ts
- **ng generate module home --flat**  
Create 1 file in the app directory
  - home.module.ts
- **ng generate module modules/contact**  
Create a modules directory in app

Create a contact directory in app/modules

Create 1 file in the app/modules/contact directory

- contact.module.ts

- **ng generate module modules/contact --flat**

Create a modules directory in app

Create 1 file in the app/modules directory

- contact.module.ts

```
# Create a home module in app, create a home directory
ng generate module home

# Creating a contact module in the app/modules directory
ng generate module modules/contact

# Creating a home module in app, not creating a home directory
ng generate module home --flat

# Creating a contact module in the app/modules directory
ng generate module modules/contact --flat
```

## Creating a class

Syntax (2 methods)

- **ng generate class [class-name]**

**generate** or **g** : Angular CLI command

**class** : creating a class

**class-name** : Name of the Angular module (the name is arbitrary)

Examples

- **ng generate class data**

Creates 2 files in the src/app directory

- data.ts

- data.spec.ts

```
# Creating a Data class  
ng generate class data
```

## Code source

The following steps will **get you a prototype application**

- [Step 1: Getting Started with Angular](#)
- [Step 2: Routing with Angular](#)
- [Step 3: Lazy loading with Angular](#)
- [Step 4: Bootstrap with Angular](#)
- [Step 5: Modules with Angular](#)
- [Step 6: Components with Angular](#)
- [Step 7: Services with Angular](#)
- [Step 8: Template Driven Forms with Angular](#)
- [Step 9: Charts with Angular](#)
- [Step 10: Server Side Rendering with Angular](#)
- [Step 11: HttpClient with Angular](#)
- [Step 12: Transfer State with Angular](#)
- [Step 13: Progressive Web App with Angular](#)
- [Step 14: Search Engine Optimization with Angular](#)

The last step is **to obtain a sample application**

- [Step 15: Create a Complete Web Application with Angular](#)

The **source code for this final application** is available on GitHub

<https://github.com/ganatan/angular-app>