

# Angular CLI Cheat Sheet



**Tavish Aggarwal**

June 18, 2020

**T**he Angular team has brought in a really handy tool to bootstrap Angular application and also to add a new feature to the app. The tool is known as angular-cli.

To install angular-cli to the system we need to install nodeJS. Once we have installed it we need to install npm dependency angular-cli (<https://www.npmjs.com/package/@angular/cli>) globally. To install it open terminal/ command prompt and run a command:

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

This will install an angular-cli package globally and now we can create a new application. To create new application we need to type the command:

```
ng new <Project name>
```

Once we have project ready, go to the project directory and serve the application:

```
cd <Project name>  
ng serve
```

This will serve the application on 4200 port. That's it.

Below is the cheatsheet file that can help you to quickly refer and run the commands. If any point in time you need more clarity on angular cli commands please refer document (<https://www.npmjs.com/package/@angular/cli#usage>).

## Angular CLI Cheat Sheet

```
ng new <project name>  # To create a new project  
  
ng serve  # To host the project on 4200 port  
  
ng serve --port <port no> --host <host name> # To host application specific host/port
```

We can use a short alias as well:

```
ng s --p <port no> --h <hostname>
```

## Some other helpful commands

```
ng lint # To lint and look for JavaScript errors
```

```
ng lint --format stylish # Linting and formatting the output
```

```
ng lint --fix # Lint and attempt to fix all the problems
```

```
ng build # to build a project in the dist folder
```

```
ng build ---target <target name> # Target for which we want to build
```

```
ng build --prod # To build in production mode
```

```
ng test # To run spec files
```

```
ng e2e # To run e2e test cases
```

```
ng doc # To look for angular documentation
```

```
ng help # To get help on angular cli commands
```

## To change the .angular-cli.json config

```
ng set # to change properties
```

```
# For e.g. ng set default.styleExt scss
```

```
# ng set default.styleExt scss -g -- To set in global angular-cli file
```

## Components

```
ng generate component <component name> # To generate new component
ng g c <component name> # Short notation to generate component

ng g c <component name> --flat # Want to generate folder name as well?

ng g c <component name> --inline-template # Want to generate HTML file?
ng g c <component name> -it # Short notation

ng g c <component name> --inline-style # Want to generate css file?
ng g c <component name> -is # Short notation

ng g c <component name> --spec # Want to generate spec file?

ng g c <component name> --view-encapsulation # View encapsulation strategy
ng g c <component name> -ve # Short notation

ng g c <component name> --change-detection # Change detection strategy
ng g c <component name> -cd # Short notation

ng g c <component name> --dry-run # To only report files and don't write them
ng g c <component name> -d # Short notation

ng g c <component name> -m <module name> -d
# Name of module where we need to add component as dependency
```

## Directives and services

```
ng generate directive <directive-name> # To generate directive
ng g d <directive-name> # short notation

ng g d <directive-name> -d # To only report files and don't write them

ng generate service <service-name> # To generate service
ng g s <service-name> # short notation

ng g s <service-name> -d # To only report files and don't write them

ng g s <service-name> -m <module name>
# Name of module where we need to add service as dependency
```

## Classes, Interface, pipe, and enums

```
ng generate class <class name> # To generate class
ng g cl <class name> # short notation

ng generate interface <interface name> # To generate interface
ng g i <interface name> # short notation

ng generate pipe <pipe name> # To generate pipe
ng g p <pipe name> # short notation

ng generate enum <enum name> # To generate enum
ng g e <enum name> # short notation
```

## Module and routing

```
ng generate module <module name> # To generate module
ng g m <module name> # To short notation

ng g m <module name> --spec true -d # To generate spec file for the module

ng g m <module name> --routing # To generate module with routing file

ng g guard <guard name> # To generate guard to route
```

## Bonus Topic

### Source Map Explorer

Source Map Explorer is a tool to view the treemap visualization to know where the code(from which file or module) is coming from. We can better understand it by using it. To download the tool run the command as shown below:

```
npm install -g source-map-explorer
```

Now we have source map explorer installed globally. To see the visualization firstly we need to build our angular app and then see the visualization. Please refer the command shown below:

```
ng build # To build the solution
source-map-explorer dist/main.bundle.js # Open a treemap visualization
```

In this post, I have tried to cover all the possible ng commands. Please let me know in comment section if I have missed any command.

## Author Info



Tavish Aggarwal

Website: <http://tavishaggarwal.com> (<http://tavishaggarwal.com>)

Living in Hyderabad and working as a research-based Data Scientist with a specialization to improve the major key performance business indicators in the area of sales, marketing, logistics, and plant productions. He is an innovative team leader with data wrangling out-of-the-box capabilities such as outlier treatment, data discovery, data transformation with a focus on yielding high-quality results.

Category

Angular (/javascript/angular)

## ALSO ON TECH LADDER

**Ultimate Guide to Machine Learning**

2 years ago

Have you ever wonder how much data we or companies produce each day!. The ...

**Change detection in Angular**

3 years ago

It is important to know how change detection works in Angular 2 to write ...

**Lasso and Ridge Regression Detailed ...**

3 years ago

In Linear Regression we saw that the complexity of the model is not ...

**Hypothesis Testing explained using ...**

3 years ago

An in-depth explanation of hypothesis testing using real-world examples. The ...



## What do you think?

13 Responses



Upvote



Funny



Love



Surprised



Angry



Sad

2 Comments

Tech Ladder

Disqus' Privacy Policy

Login ▾

Favorite

Tweet

Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

**Bhadresh Arya** • 2 years ago

Thanks for the sharing.. &lt;3

1 ^ | ▾ • Reply • Share ›

**Tavish Aggarwal** Mod ➔ Bhadresh Arya • 2 years ago

:)

^ | ▾ • Reply • Share ›

Subscribe Add Disqus to your siteAdd DisqusAdd Do Not Sell My Data

For all the latest Industry updates - subscribe to our newsletter

**Email**

*The subscriber's email address.*

**Subscribe**



Copyright © 2022. All rights reserved

Designed By Tavish Aggarwal (<https://www.github.com/tavishaggarwal>)