✓ Angular 7 Tutorial
- Angular 7 Tutorial
- Angular 7 Introduction
- Angular 7 Features
- Angular 7 Installation
- History & Versions
- Angular 7 First App
- Angular 7 Files Explanat
- Angular 7 with Bootstrap
- **All CLI commands**
- Angular 7 Libraries
- Angular 7 Architecture
- Angular 7 Components

✓ Angular 7 Directives
- Angular 7 Directives
- Angular 7 ngIf Directive
- Angular 7 ngStyle Direct

✓ Angular 7 Databinding
- Angular 7 Databinding
- Angular 7 String Interpolation
- Angular 7 Event Binding
- Angular 7 Property Bind

All Angular CLI commands

← prev next →

Angular CLI is a command line interface tool which is used to initialize, develop, scaffold, and maintain Angular applications. You can use these command directly on command prompt or indirectly through an interactive UI i.e. Angular Console.

Command	Alias	Description
add		It is used to add support for an external library to your project.
build	b	It compiles an Angular app into an output directory named dist/ at the given output path. Must be executed from within a workspace directory.
config		It retrieves or sets Angular configuration values in the angular.json file for the workspace.
doc	d	It opens the official Angular documentation (angular.io) in a browser, and searches for a given keyword.
e2e	e	It builds and serves an Angular app, then runs end-to-end tests using Protractor.
generate	g	It generates and/or modifies files based on a schematic.
help		It provides a list of available commands and their short descriptions.

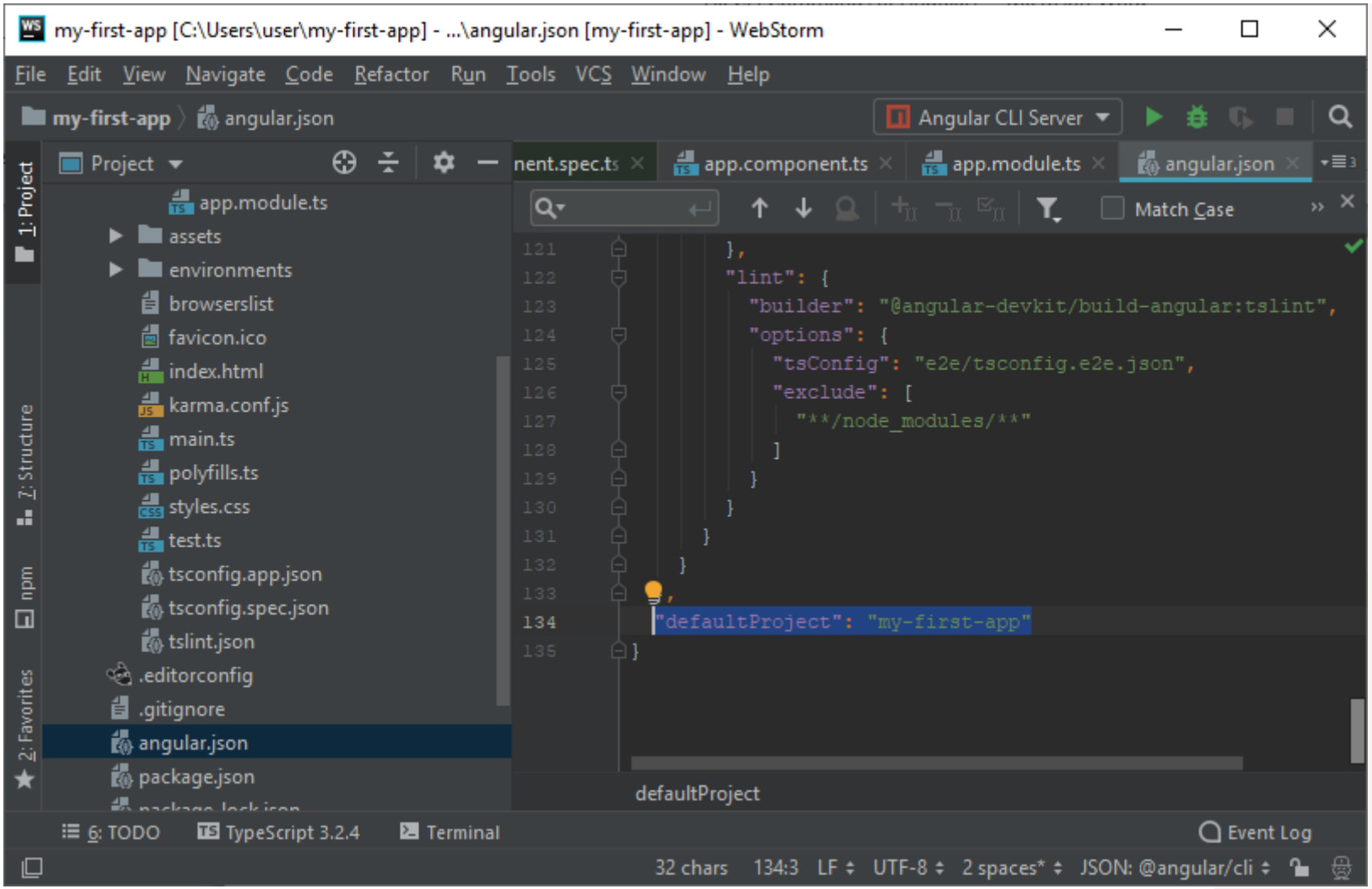
- Angular 7 Pipes
  - Angular 7 Pipes
- Error Fixing
  - Angular 7 Error Fixing
- Angular 7 Forms
  - Angular 7 Forms
  - Data Flow in Angular Fo
  - Angular Reactive Forms
  - Template-driven Forms
- Angular Misc
  - Angular vs React
- Angular + Spring
  - CRUD Example
  - File Upload Example
  - Login & Logout Example
  - Search Field Example

lint	l	It is used to run linting tools on Angular app code in a given project folder.
new	n	It creates a new workspace and an initial Angular app.
run		It runs an Architect target with an optional custom builder configuration defined in your project.
serve	s	It builds and serves your app, rebuilding on file changes.
test	t	It runs unit tests in a project.
update		It updates your application and its dependencies. See https://update.angular.io/
version	v	It utputs Angular CLI version.
xi18n		It extracts i18n messages from source code.

ng add Command

The ng add command is used to add support for an external library to your project. It adds the npm package for a published library to your workspace, and makes your default app project to use that library, in whatever way is specified by the library's schematic. For example, if you add @angular/pwa, then it will configure your project for PWA support.

The default app project is the value of defaultProject in angular.json.



Syntax:

```
ng add <collection> [options]
```

Parameter Explanation:

<collection>: It specifies the package which you want to add.

Options

--defaults=true|false: When true, it disables interactive input prompts for options with a default.

--help=true|false|json|JSON: It is used to show a help message in the console. Default: false

--interactive=true|false: When false, it disables interactive input prompts.

ng build Command

The ng build command is used to compile an Angular app into an output directory named dist/ at the given output path. It must be executed from within a workspace directory.

Syntax:

```
ng build <project> [options]
ng b <project> [options]
```

Parameter Explanation:

<project>: It specifies the name of the project to build. It can be an app or a library.

Options

--aot=true|false: It builds using Ahead of Time compilation. Default: false

--baseHref=baseHref: It specifies the base url for the application being built.

--buildEventLog=buildEventLog: (experimental) Output file path for Build Event Protocol events.

--buildOptimizer=true|false: It enables '@angular-devkit/build-optimizer' optimizations when using the 'aot' option. Default: false

--commonChunk=true|false: It uses a separate bundle containing code used across multiple bundles. Default: true

--configuration=configuration: A named build target, as specified in the "configurations" section of angular.json. Each named target is accompanied by a configuration of option defaults for that target.

Aliases: -c

--deleteOutputPath=true|false: It is used to delete the output path before building. Default: true

--deployUrl=deployUrl: URL where files will be deployed.

--es5BrowserSupport=true|false: Enables conditionally loaded ES2015 polyfills. Default: false

--extractCss=true|false: It is used to extract css from global styles into css files instead of js ones. Default: false

--extractLicenses=true|false: It is used to extract all licenses in a separate file. Default: false

--forkTypeChecker=true|false: It is used to run the TypeScript type checker in a forked process. Default: true

--help=true|false|json|JSON: It is used to show a help message for this command in the console. Default: false

--i18nFile=i18nFile: Localization file to use for i18n.

--i18nFormat=i18nFormat: Format of the localization file specified with --i18n-file.

--i18nLocale=i18nLocale: Locale to use for i18n.

--i18nMissingTranslation=i18nMissingTranslation: How to handle missing translations for i18n.

--index=index: The name of the index HTML file.

--lazyModules: List of additional NgModule files that will be lazy loaded. Lazy router modules will be discovered automatically.

--main=main: The full path for the main entry point to the app, relative to the current workspace.

--namedChunks=true|false: Use file name for lazy loaded chunk Default: true

--ngswConfigPath=ngswConfigPath: Path to ngsw-config.json.

--optimization=true|false: Enables optimization of the build output.

--outputHashing= none|all|media|bundles: Define the output filename cache-busting hashing mode.

Default: none

--outputPath=outputPath: The full path for the new output directory, relative to the current workspace.

By default, writes output to a folder named dist/ in the current project.

--poll: Enable and define the file watching poll time period in milliseconds.

--polyfills=polyfills: The full path for the polyfills file, relative to the current workspace.

--preserveSymlinks=true|false: Do not use the real path when resolving modules.

Default: false

--prod=true|false: When true, sets the build configuration to the production target. All builds make use of bundling and limited tree-shaking. A production build also runs limited dead code elimination.

--profile=true|false: Output profile events for Chrome profiler.

Default: false

--progress=true|false: Log progress to the console while building.

--resourcesOutputPath= resourcesOutputPath: The path where style resources will be placed, relative to outputPath.

--serviceWorker=true|false: Generates a service worker config for production builds. Default: false

--showCircularDependencies=true|false: Show circular dependency warnings on builds. Default: true

--sourceMap=true|false: It is used to show Output sourcemaps.

Default: true

--statsJson=true|false: It generates a 'stats.json' file which can be analyzed using tools such as: 'webpack-bundle-analyzer' or https://webpack.github.io/analyse.

Default: false

--subresourceIntegrity=true|false: It enables the use of subresource integrity validation.

Default: false

--tsConfig=tsConfig: The full path for the TypeScript configuration file, relative to the current workspace.

--vendorChunk=true|false: It uses a separate bundle containing only vendor libraries.

Default: true

--verbose=true|false: It adds more details to output logging.

Default: false

--watch=true|false: It runs build when files change.

Default: false

ng config Command

The ng config command is used to retrieve or set Angular configuration values in the angular.json file for the workspace.

Syntax:

```
ng config <jsonPath> <value> [options]
```

Parameter Explanation:

<jsonPath>: The configuration key to set or query, in JSON path format. For example: "a[3].foo.bar[2]". If no new value is provided, returns the current value of this key.

<value>: If provided, a new value for the given configuration key.

Options

--global=true|false: When true, it accesses the global configuration in the caller's home directory.

Default: false

Aliases: -g

--help= true|false|json|JSON: It is used to show a help message for this command in the console.

Default: false

ng doc Command

The ng doc command is used to open the official Angular documentation (angular.io) in a browser, and searches for a given keyword.

Syntax:

```
ng doc <keyword> [options]
ng d <keyword> [options]
```

Parameter Explanation:

<keyword>: It is used to specify the keyword to search for, as provided in the search bar in angular.io.

Options

--help=true|false|json|JSON: It is used to show a help message for this command in the console.

Default: false

--search=true|false: When true, it searches all of angular.io. Otherwise, searches only API reference documentation.

Default: false

Aliases: -s

ng e2e Command

It is used to build and serve an Angular app, then runs end-to-end tests using Protractor.

Syntax:

```
ng e2e <project> [options]
ng e <project> [options]
```

It must be executed from within a workspace directory. When you don't specify the project name, it executes for all projects.

Parameter Explanation:

<project>: It specifies the name of the project you want to build. It can be an app or a library.

Options

--baseUrl=baseUrl: It specifies the base URL for protractor to connect to.

--configuration=configuration: It is used to specify named build target, as specified in the "configurations" section of angular.json. Each named target is accompanied by a configuration of option defaults for that target.

Aliases: -c

--devServerTarget=devServerTarget: It specifies dev server target to run tests against.

--elementExplorer=true|false: It starts Protractor's Element Explorer for debugging.

Default: false

--help= true|false|json|JSON: It shows a help message for this command in the console.

Default: false

--host=host: Host to listen on.

Default: localhost

--port: It specifies the port to serve the application.

--prod=true|false: When true, it sets the build configuration to the production target. All builds make use of bundling and limited tree-shaking. A production build also runs limited dead code elimination.

--protractorConfig= protractorConfig: It specifies the name of the Protractor configuration file.

--specs: It overrides specs in the protractor config.

--suite=suite: It overrides suite in the protractor config.

--webdriverUpdate=true|false: It is used to update webdriver.

Default: true

ng generate Command

The ng generate command is used to generate and/or modify file based on schematic.

```
ng generate <schematic> [options]
ng g <schematic> [options]
```

Parameter Explanation:

<schematic >: It specifies the schematic or collection:schematic which you want to generate. It can take one of the following sub-commands.

- appShell
- application
- class
- component
- directive
- enum
- guard

- interface
- library
- module
- pipe
- service
- serviceWorker
- universal

Schematic Command Explanation

appShell:

It is used to generate an app shell for running a server-side version of an app.

Syntax:

```
ng generate appShell [options]
ng g appShell [options]
```

application

It is used to create a new basic app definition in the "projects" subfolder of the workspace.

Syntax:

```
ng generate application <name> [options]
ng g application <name> [options]
```

class

It is used to create a new generic class definition in the given or default project.

Syntax:

```
ng generate class <name> [options]
ng g class <name> [options]
```

component

It is used to create a new generic component definition in the given or default project.

Syntax:

```
ng generate component <name> [options]
ng g component <name> [options]
```

directive

It is used to create a new generic directive definition in the given or default project.

Syntax:

```
ng generate directive <name> [options]
ng g directive <name> [options]
```

enum

It is used to create a new, generic enum definition for the given or default project.

Syntax:

```
ng generate enum <name> [options]
ng g enum <name> [options]
```

enum

It is used to create a new, generic enum definition for the given or default project.

Syntax:

```
ng generate enum <name> [options]
ng g enum <name> [options]
```

guard

It is used to generate a new, generic route guard definition in the given or default project.

Syntax:

```
ng generate enum <name> [options]
ng g enum <name> [options]
```

interface

It is used to create a new generic interface definition in the given or default project.

Syntax:

```
```

```
ng generate interface <name> <type> [options]
ng g interface <name> <type> [options]
```

library

It is used to create a new generic library project in the current workspace.

Syntax:

```
ng generate library <name> [options]
ng g library <name> [options]
```

module

It is used to create a new generic NgModule definition in the given or default project.

Syntax:

```
ng generate module <name> [options]
ng g module <name> [options]
```

pipe

It is used to create a new generic pipe definition in the given or default project.

Syntax:

```
ng generate pipe <name> [options]
ng g pipe <name> [options]
```

service

It is used to create a new, generic service definition in the given or default project.

Syntax:

```
ng generate service <name> [options]
ng g service <name> [options]
```

serviceWorker

This is used to pass this schematic to the "run" command to create a service worker.

Syntax:

```
ng generate serviceWorker [options]
ng g serviceWorker [options]
```

universal

This command is used to pass this schematic to the "run" command to set up server-side rendering for an app.

Syntax:

```
ng generate universal [options]
ng g universal [options]
```

Options

--defaults=true|false: When true, it disables interactive input prompts for options with a default.

--dryRun=true|false: When true, it runs through and reports activity without writing out results.

Default: false

Aliases: -d

--force=true|false: When true, it forces overwriting of existing files.

Default: false

Aliases: -f

--help=true|false|json|JSON: It is used to show a help message in the console.

Default: false

--interactive=true|false: When false, it disables interactive input prompts.

Next Topic Angular 7 Libraries

← prev

next →

Help Others, Please Share

Learn Latest Tutorials

UML Tutorial

UML

Artificial Neural Network Tutorial

ANN

ES6 Tutorial

ES6

Flutter Tutorial

Flutter

Selenium Python

Selenium Py

Firebase Tutorial

Firebase

Cobol Tutorial

Cobol

Ansible Tutorial

Ansible

Mockito Tutorial

Mockito

Talend Tutorial

Talend

Microsoft Azure Tutorial

Azure

Sharepoint Tutorial

SharePoint

Preparation

Aptitude

Aptitude

Logical Reasoning

Reasoning

Verbal Ability

Verbal A.

Interview Questions

Interview

Company Interview Questions

Company

Trending Technologies

Artificial Intelligence Tutorial

AWS Tutorial

AWS

Selenium tutorial

Selenium

Cloud tutorial

Cloud

Hadoop tutorial

Hadoop

AI

ReactJS Tutorial

ReactJS

Data Science Tutorial

D. Science

Angular 7 Tutorial

Angular 7

Blockchain Tutorial

Blockchain

Git Tutorial

Git

Machine Learning Tutorial

ML

DevOps Tutorial

DevOps

B.Tech / MCA

DBMS tutorial

DBMS

Data Structures tutorial

DS

DAA tutorial

DAA

Operating System tutorial

OS

Computer Network tutorial

C. Network

Compiler Design tutorial

Compiler D.

Computer Organization and Architecture

COA

Discrete Mathematics Tutorial

D. Math.

Ethical Hacking Tutorial

E. Hacking

Computer Graphics Tutorial

C. Graphics

Software Engineering Tutorial

Software E.

html tutorial

Web Tech.

Cyber Security tutorial

Cyber Sec.

Automata Tutorial

Automata

C Language tutorial

C

C++

Java

.Net

Python

Programs

Control S.

Data Mining

Javatpoint Services

JavaTpoint offers too many high quality services. Mail us on hr@javatpoint.com, to get more information about given services.

- Website Designing
- Website Development
- Java Development
- PHP Development
- WordPress
- Graphic Designing
- Logo
- Digital Marketing
- On Page and Off Page SEO
- PPC
- Content Development
- Corporate Training
- Classroom and Online Training
- Data Entry

Training For College Campus

JavaTpoint offers college campus training on Core Java, Advance Java, .Net, Android, Hadoop, PHP, Web Technology and Python. Please mail your requirement at hr@javatpoint.com.

Duration: 1 week to 2 week

Like/Subscribe us for latest updates or newsletter