ANGULAR CLI

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

Angular CLI makes it easy to start with any Angular project. Angular CLI comes with commands that help us create and start on our project very fast. It has commands to create a project, a component and services, change the port, etc.

Audience

This tutorial is designed for software programmers who want to learn the basics of Angular CLI and its concepts in a simple and easy manner. This tutorial will give you enough understanding on the various functionalities of Angular CLI with suitable examples.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of HTML, CSS, JavaScript, Typescript, and Document Object Model (DOM).

Copyright & Disclaimer

Table of Contents

About the Tutorial i

Audience i

Prerequisites i

Copyright & Disclaimer i

Table of Contents ii

1. Angular CLI – Overview 1

Commands for Angular 4 1

2. Angular CLI — Environment Setup 3

Download Node.js archive 3

Installation on UNIX/Linux/Mac OS X, and SunOS 3

Installation on Windows 4

3. Angular CLI — ng version Command 6

4. Angular CLI — ng new Command 8

5. Angular CLI — ng help Command 13

6. Angular CLI — ng generate Command 18

7. Angular CLI — ng build Command 20

8. Angular CLI — ng run Command 26

9. Angular CLI — ng serve Command 28

10. Angular CLI — ng lint Command 32

11. Angular CLI — ng test Command 36

12. Angular CLI — ng e2e Command 41

13. Angular CLI — ng add Command 45

14. Angular CLI — ng analytics Command 47

15. Angular CLI — ng config Command 49

16. Angular CLI — ng doc Command 51

17. Angular CLI — ng update Command 53

18. Angular CLI — ng xi18n Command 55

19. Angular CLI — Code Coverage 58

1. Angular CLI – Overview

Angular command-line interface (CLI) makes it easy to start with any Angular project. It comes with commands that help us create and start on our project very fast. Let us now go through the commands available to create a project, a component and services, change the port, etc.

To work with Angular CLI, we need to have it installed on our system. Let us use the following command for the same:

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

To create a new project, we can run the following command in the command line and the project will be created.

```
ng new PROJECT-NAME
cd PROJECT-NAME
ng serve //
```

ng serve // will compile and you can see the output of your project in the browser:

```
http://localhost:4200/
```

4200 is the default port used when a new project is created. You can change the port with the following command:

```
ng serve --host 0.0.0.0 --port 4201
```

Commands for Angular 4

The following table lists down a few important commands which are required while working with the Angular 4 projects.

Sr.No	Commands & Description
1	**Component** ng g component new-component
2	**Directive** ng g directive new-directive
3	**Pipe**

	ng g pipe new-pipe
4	**Service** ng g service new-service
5	**Module** ng g module my-module

Whenever a new module, a component, or a service is created, the reference of the same is updated in the parent module **app.module.ts**.

2. Angular CLI — Environment Setup

To work with Angular CLI, we need to have Node installed on our system. Let us understand about the environment setup required for Angular CLI in detail.

Download Node.js archive

Download latest version of Node.js installable archive file from **Node.js Downloads**, which is available at https://nodejs.org/download/. At the time of writing this tutorial, the versions available on different OS are listed below:

OS	Archive name
Windows	node-v6.3.1-x64.msi
Linux	node-v6.3.1-linux-x86.tar.gz
Mac	node-v6.3.1-darwin-x86.tar.gz
SunOS	node-v6.3.1-sunos-x86.tar.gz

Installation on UNIX/Linux/Mac OS X, and SunOS

Based on your OS architecture, download and extract the archive **node-v6.3.1-osname.tar.gz** into /tmp, and then, finally move extracted files into /usr/local/nodejs directory.

For example:

```
$ cd /tmp
$ wget http://nodejs.org/dist/v6.3.1/node-v6.3.1-linux-x64.tar.gz
$ tar xvfz node-v6.3.1-linux-x64.tar.gz
$ mkdir -p /usr/local/nodejs
$ mv node-v6.3.1-linux-x64/* /usr/local/nodejs
```

Add /usr/local/nodejs/bin to the PATH environment variable.

OS	Output
Linux	export PATH=$PATH:/usr/local/nodejs/bin

Mac	export PATH=$PATH:/usr/local/nodejs/bin
FreeBSD	export PATH=$PATH:/usr/local/nodejs/bin

Installation on Windows

Use the MSI file and follow the prompts to install the **Node.js**. By default, the installer uses the Node.js distribution in **C:\Program Files\nodejs**.

The installer should set the **C:\Program Files\nodejs\bin** directory in windows PATH environment variable. Restart any open command prompts for the change to take effect.

Verify installation: Executing a File

Create a js file named **main.js** on your machine (Windows or Linux) having the following code:

```
/* Hello, World! program in node.js */
console.log("Hello, World!")
```

The link for live demo is https://www.tutorialspoint.com/execute_nodejs_online.php.

Now, execute main.js file using Node.js interpreter to see the result:

```
$ node main.js
```

If everything is fine with your installation, this should produce the following result:

```
Hello, World!
```

Now, the Node is installed. You can run the following command to install Angular CLI.

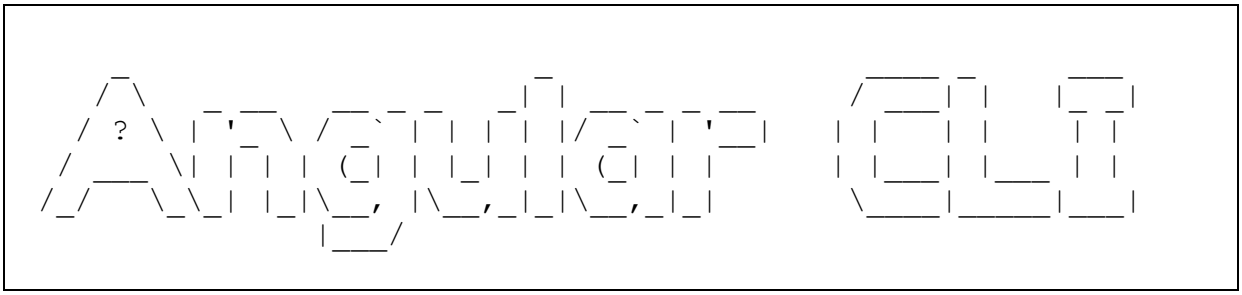
```
npm install -g @angular/cli
```

Verify the installation

Now, run the following command to see the result:

```
$ ng --version
```

If everything is fine with your installation, this should produce the following result:



```
Angular CLI: 9.1.0
Node: 12.16.1
OS: win32 x64

Angular:
...
Ivy Workspace:

Package                      Version
------------------------------------------------------
@angular-devkit/architect    0.901.0
@angular-devkit/core         9.1.0
@angular-devkit/schematics   9.1.0
@schematics/angular          9.1.0
@schematics/update           0.901.0
rxjs                         6.5.4
```

On Windows, in case of ng being not recognised as internal or external command, update the system path variable to include the following path.

```
C:\Users\<User Directory>\AppData\Roaming\npm
```

3. Angular CLI — ng version Command

This chapter explains the syntax, options of ng version command along with an example.

Syntax

The syntax for ng version command is as follows:

```
ng version [options]
ng v [options]
```

ng version command shows the Angular CLI version installed.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--help= true|false|json|JSON	Shows a help message for this command in the console. Default: false

Example

An example for ng version command is given below:

```
\>Node ng version


     _                      _                 ____ _     ___
    / \   _ __   __ _ _   _| | __ _ _ __     / ___| |   |_ _|
   / ? \ | '_ \ / _` | | | | |/ _` | '__|   | |   | |    | |
  / ___ \| | | | (_| | |_| | | (_| | |      | |___| |___ | |
 /_/   \_\_| |_|\__, |\__,_|_|\__,_|_|       \____|_____|___|
                |___/


Angular CLI: 9.1.0
Node: 12.16.1
```

```
OS: win32 x64

Angular:
...
Ivy Workspace:

Package                      Version
------------------------------------------------------
@angular-devkit/architect    0.901.0
@angular-devkit/core         9.1.0
@angular-devkit/schematics   9.1.0
@schematics/angular          9.1.0
@schematics/update           0.901.0
rxjs                         6.5.4
```

4. Angular CLI — ng new Command

This chapter explains the syntax, argument and options of ng new command along with an example.

Syntax

The syntax for ng new command is as follows:

```
ng new <name> [options]
ng n <name> [options]
```

ng new command creates a workspace of given **name** with a default Angular Application. It provides interactive prompts to set optional configurations. All prompts have default values to choose.

Argument

The argument for ng new command is as follows:

Sr.No.	Argument & Syntax	Description
1	<name>	The name of the new workspace and initial project.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--collection=collection	A collection of schematics to use in generating the initial app. Aliases: -c.
2	--commit=true|false	Initial git repository commit information. Default: true.
3	--createApplication=true|false	When true (the default), creates a new initial app project in the src folder of the new workspace. When false, creates an empty

		workspace with no initial app. You can then use to generate application command so that all apps are created in the projects folder. Default: true.
4	--defaults=true|false	When true, disables interactive input prompts for options with a default.
5	--directory=directory	The directory name to create the workspace in.
6	--dryRun=true|false	When true, runs through and reports activity without writing out results. Default: false. Aliases: -d.
7	--force=true|false	When true, forces overwriting of existing files. Default: false. Aliases: -f.
8	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false.
9	--inlineStyle=true|false	When true, includes styles inline in the component TS file. By default, an external styles file is created and referenced in the component TS file. Default: false.
10	--inlineStyle=true|false	When true, includes styles inline in the component

		TS file. By default, an external styles file is created and referenced in the component TS file. Default: false. Aliases: -t.
11	--interactive=true|false	When false, disables interactive input prompts.
12	--minimal=true|false	When true, creates a project without any testing frameworks. (Use for learning purposes only.) Default: false.
13	--newProjectRoot=newProjectRoot	The path where new projects will be created, relative to the new workspace root. Default: projects.
14	--packageManager=npm|yarn|pnpm|cnpm	The package manager used to install dependencies.
15	--prefix=prefix	The prefix to apply to generated selectors for the initial project. Default: app. Aliases: -p.
16	--routing=true|false	When true, generates a routing module for the initial project.
17	--skipGit=true|false	When true, does not initialize a git repository. Default: false. Aliases: -g.

18	--skipInstall=true|false	When true, does not install dependency packages. Default: false.
19	--skipTests=true|false	When true, does not generate "spec.ts" test files for the new project. Default: false. Aliases: -S.
20	--strict=true|false	Creates a workspace with stricter TypeScript compiler options. Default: false.
21	--style=css|scss|sass|less|styl	The file extension or preprocessor to use for style files.
22	--verbose=true|false	When true, adds more details to output logging. Default: false. Aliases: -v.
23	--viewEncapsulation=Emulated|Native|None|ShadowDom	The view encapsulation strategy to use in the initial project.

Example

An example for ng new command is given below:

```
\>Node ng new TutorialsPoint
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE TutorialsPoint/angular.json (3630 bytes)
CREATE TutorialsPoint/package.json (1291 bytes)
CREATE TutorialsPoint/README.md (1031 bytes)
CREATE TutorialsPoint/tsconfig.json (489 bytes)
```

```
CREATE TutorialsPoint/tslint.json (3125 bytes)
CREATE TutorialsPoint/.editorconfig (274 bytes)
CREATE TutorialsPoint/.gitignore (631 bytes)
CREATE TutorialsPoint/browserslist (429 bytes)
CREATE TutorialsPoint/karma.conf.js (1026 bytes)
CREATE TutorialsPoint/tsconfig.app.json (210 bytes)
CREATE TutorialsPoint/tsconfig.spec.json (270 bytes)
CREATE TutorialsPoint/src/favicon.ico (948 bytes)
CREATE TutorialsPoint/src/index.html (300 bytes)
CREATE TutorialsPoint/src/main.ts (372 bytes)
CREATE TutorialsPoint/src/polyfills.ts (2835 bytes)
CREATE TutorialsPoint/src/styles.css (80 bytes)
CREATE TutorialsPoint/src/test.ts (753 bytes)
CREATE TutorialsPoint/src/assets/.gitkeep (0 bytes)
CREATE TutorialsPoint/src/environments/environment.prod.ts (51 bytes)
CREATE TutorialsPoint/src/environments/environment.ts (662 bytes)
CREATE TutorialsPoint/src/app/app-routing.module.ts (246 bytes)
CREATE TutorialsPoint/src/app/app.module.ts (393 bytes)
CREATE TutorialsPoint/src/app/app.component.html (25755 bytes)
CREATE TutorialsPoint/src/app/app.component.spec.ts (1083 bytes)
CREATE TutorialsPoint/src/app/app.component.ts (218 bytes)
CREATE TutorialsPoint/src/app/app.component.css (0 bytes)
CREATE TutorialsPoint/e2e/protractor.conf.js (808 bytes)
CREATE TutorialsPoint/e2e/tsconfig.json (214 bytes)
CREATE TutorialsPoint/e2e/src/app.e2e-spec.ts (647 bytes)
CREATE TutorialsPoint/e2e/src/app.po.ts (301 bytes)
 Packages installed successfully.
```

Here, ng new command has created an angular workspace and a project with name TutorialsPoint in our Node directory.

5. Angular CLI — ng help Command

This chapter explains the syntax and options of ng help command along with an example.

Syntax

An example for ng help command is given below:

```
ng help [options]
```

ng help command lists the available commands with their short descriptions.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--help= true|false|json|JSON	Shows a help message for this command in the console. Default: false

Example 1

An example for ng help command is given below:

```
\>Node ng help
Available Commands:
  add Adds support for an external library to your project.
  analytics Configures the gathering of Angular CLI usage metrics. See
https://v
8.angular.io/cli/usage-analytics-gathering.
  build (b) Compiles an Angular app into an output directory named dist/ at the
given output path. Must be executed from within a workspace directory.
  deploy Invokes the deploy builder for a specified project or for the default
p
roject in the workspace.
  config Retrieves or sets Angular configuration values in the angular.json
file
 for the workspace.
  doc (d) Opens the official Angular documentation (angular.io) in a browser,
an
```

```
d searches for a given keyword.

  e2e (e) Builds and serves an Angular app, then runs end-to-end tests using
Pro

tractor.

  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
def

ined 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
https://update.angul

ar.io/

  version (v) Outputs Angular CLI version.

  xi18n (i18n-extract) Extracts i18n messages from source code.


For more detailed help run "ng [command name] --help"
```

In case of individual commands, use the --help or -h option with the command. First move to an angular project created using **ng new** command and then run the command. The chapter which explains the ng new command is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_new.htm.

Example 2

An example is given below:

```
\>Node\>TutorialsPoint> ng serve --help

Builds and serves your app, rebuilding on file changes.

usage: ng serve <project> [options]


arguments:

   project

      The name of the project to build. Can be an application or a library.


options:

   --allowed-hosts
```

```
      Whitelist of hosts that are allowed to access the dev server.
   --aot
      Build using Ahead of Time compilation.
   --base-href
      Base url for the application being built.
   --browser-target
      Target to serve.
   --build-event-log
      **EXPERIMENTAL** Output file path for Build Event Protocol events
   --common-chunk
      Use a separate bundle containing code used across multiple bundles.
   --configuration (-c)
      A named build target, as specified in the "configurations" section of
angula
r.json.
   Each named target is accompanied by a configuration of option defaults for t
hat target.
    Setting this explicitly overrides the "--prod" flag
   --deploy-url
      URL where files will be deployed.
   --disable-host-check
      Don't verify connected clients are part of allowed hosts.
   --eval-source-map
      Output in-file eval sourcemaps.
   --help
      Shows a help message for this command in the console.
   --hmr
      Enable hot module replacement.
   --hmr-warning
      Show a warning when the --hmr option is enabled.
   --host
      Host to listen on.
   --live-reload
     Whether to reload the page on change, using live-reload.
   --open (-o)
      Opens the url in default browser.
```

```
   --optimization

      Enables optimization of the build output.

   --poll

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

   --port

      Port to listen on.

   --prod

      Shorthand for "--configuration=production".

      When true, sets the build configuration to the production target.

      By default, the production target is set up in the workspace
configuration s

uch that all builds make use of bundling, limited tree-shaking, and also
limited

  dead code elimination.

   --progress

      Log progress to the console while building.

   --proxy-config

      Proxy configuration file.

   --public-host

      The URL that the browser client (or live-reload client, if enabled)
should u

see to connect to the development server. Use for a complex dev server setup,
such as one with reverse proxies.

   --serve-path

      The pathname where the app will be served.

   --serve-path-default-warning

      Show a warning when deploy-url/base-href use unsupported serve path
values.

   --source-map

      Output sourcemaps.

   --ssl

      Serve using HTTPS.

   --ssl-cert

      SSL certificate to use for serving HTTPS.

   --ssl-key

      SSL key to use for serving HTTPS.

   --vendor-chunk
```

```
      Use a separate bundle containing only vendor libraries.
   --vendor-source-map
      Resolve vendor packages sourcemaps.
   --verbose
      Adds more details to output logging.
   --watch
      Rebuild on change.
```

6. Angular CLI — ng generate Command

This chapter explains the syntax, argument and options of ng generate command along with an example.

Syntax

The syntax for ng generate command is as follows:

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

ng generate command generates and/or modifies files based on a schematic.

Argument

The argument for ng help command is as follows:

Sr.No.	Argument & Syntax	Description
1	<schematic>	The schematic or collection:schematic to generate. This option can take one of the following sub-commands: • appShell • application • class • component • directive • enum • guard • interceptor • interface • library • module • pipe • service • serviceWorker • webWorker

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--defaults=true|false	When true, disables interactive input prompts for options with a default.
2	--dryRun=true|false	When true, runs through and reports activity without writing out results. Default: false. Aliases: -d.
3	--force=true|false	When true, forces overwriting of existing files. Default: false. Aliases: -f.
4	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false.
5	--interactive=true|false	When false, disables interactive input prompts.

First move to an angular project created using **ng new** command and then run the command. This chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_new.htm.

Example

An example for ng generate command is given below:

```
\>Node\>TutorialsPoint> ng generate component goals
CREATE src/app/goals/goals.component.html (20 bytes)
CREATE src/app/goals/goals.component.spec.ts (621 bytes)
CREATE src/app/goals/goals.component.ts (271 bytes)
CREATE src/app/goals/goals.component.css (0 bytes)
UPDATE src/app/app.module.ts (471 bytes)
```

Here, ng generate command has created a new component in our project TutorialsPoint and added this new component entry in app.module.ts.

7. Angular CLI — ng build Command

This chapter explains the syntax, argument and options of ng build command along with an example.

Syntax

The syntax for ng build command is as follows:

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

ng build command compiles an angular application/library into an output directory named dist at given path.

Arguments

The argument for ng build command is as follows:

Sr.No.	Argument & Syntax	Description
1	<project>	The name of the application or library to be built.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--aot=true|false	Build using Ahead of Time compilation. Default: false.
2	--baseHref=baseHref	Base url for the application being built.
3	--buildEventLog=buildEventLog	EXPERIMENTAL Output file path for Build Event Protocol events.
4	--buildOptimizer=true|false	Enables '@angular-devkit/build-optimizer' optimizations when using the 'aot' option. Default: false.

5	--commonChunk=true|false	Use a separate bundle containing code used across multiple bundles. Default: false.
6	--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. Setting this explicitly overrides the "--prod" flag. Aliases: -c.
7	--crossOrigin=none|anonymous|use-credentials	Define the crossorigin attribute setting of elements that provide CORS support. Default: none.
8	--deleteOutputPath=true|false	Delete the output path before building. Default: true.
9	--deployUrl=deployUrl	URL where files will be deployed.
10	--experimentalRollupPass=true|false	Concatenate modules with Rollup before bundling them with Webpack. Default: false.
11	--extractCss=true|false	Extract css from global styles into css files instead of js ones. Default: false.
12	--extractLicenses=true|false	Extract all licenses in a separate file. Default: false.
13	--forkTypeChecker=true|false	Run the TypeScript type checker in a forked process. Default: true.
14	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false.

15	--i18nMissingTranslation=warning|error|ignore	How to handle missing translations for i18n. Default: warning.
16	--index=index	Configures the generation of the application's HTML index.
17	--localize=true|false	
18	--main=main	The full path for the main entry point to the app, relative to the current workspace.
19	--namedChunks=true|false	Use file name for lazy loaded chunks. Default: true.
20	--ngswConfigPath=ngswConfigPath	Path to ngsw-config.json.
21	--optimization=true|false	Enables optimization of the build output.
22	--outputHashing=none|all|media|bundles	Define the output filename cache-busting hashing mode. Default: none.
23	--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.
24	--poll	Enable and define the file watching poll time period in milliseconds.
25	--polyfills=polyfills	The full path for the polyfills file, relative to the current workspace.
26	--preserveSymlinks=true|false	Do not use the real path when resolving modules. Default: false.

27	--prod=true|false	Shorthand for "--configuration=production". When true, sets the build configuration to the production target. By default, the production target is set up in the workspace configuration such that all builds make use of bundling, limited tree-shaking, and also limited dead code elimination.
28	--progress=true|false	Log progress to the console while building. Default: true.
27	--resourcesOutputPath=resourcesOutputPath	The path where style resources will be placed, relative to outputPath.
28	--serviceWorker=true|false	Generates a service worker config for production builds. Default: false.
29	--showCircularDependencies=true|false	Show circular dependency warnings on builds. Default: true.
30	--sourceMap=true|false	Output sourcemaps. Default: true.
31	--statsJson=true|false	Generates a 'stats.json' file which can be analyzed using tools such as 'webpack-bundle-analyzer'. Default: false.
32	--subresourceIntegrity=true|false	Enables the use of subresource integrity validation. Default: false.
33	--tsConfig=tsConfig	The full path for the TypeScript configuration file, relative to the current workspace.

34	--vendorChunk=true|false	Use a separate bundle containing only vendor libraries. Default: true.
35	--verbose=true|false	Adds more details to output logging. Default: true.
36	--watch=true|false	Run build when files change. Default: false.
37	--webWorkerTsConfig=webWorkerTsConfig	TypeScript configuration for Web Worker modules.

First, move to an angular project updated using **ng generate** command. Replace content of app.component.html with following contents and then, run the command. This chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_generate.htm.

```
<app-goals></app-goals>
<router-outlet></router-outlet>
```

Example

An example for ng build command is given below:

```
\>Node\>TutorialsPoint> ng build
Compiling @angular/animations : es2015 as esm2015
Compiling @angular/core : es2015 as esm2015
Compiling @angular/compiler/testing : es2015 as esm2015
Compiling @angular/animations/browser : es2015 as esm2015
Compiling @angular/core/testing : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/common/http : es2015 as esm2015
Compiling @angular/common/testing : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015
Compiling @angular/platform-browser/testing : es2015 as esm2015
Compiling @angular/router : es2015 as esm2015
Compiling @angular/animations/browser/testing : es2015 as esm2015
Compiling @angular/common/http/testing : es2015 as esm2015
Compiling @angular/forms : es2015 as esm2015
```

```
Compiling @angular/platform-browser/animations : es2015 as esm2015
Compiling @angular/platform-browser-dynamic/testing : es2015 as esm2015
Compiling @angular/router/testing : es2015 as esm2015
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {polyfills} polyfills-es2015.js, polyfills-es2015.js.map (polyfills) 141
kB [initial] [rendered]
chunk {runtime} runtime-es2015.js, runtime-es2015.js.map (runtime) 6.16 kB
[entry] [rendered]
chunk {runtime} runtime-es5.js, runtime-es5.js.map (runtime) 6.16 kB [entry]
[rendered]
chunk {styles} styles-es2015.js, styles-es2015.js.map (styles) 12.4 kB
[initial] [rendered]
chunk {styles} styles-es5.js, styles-es5.js.map (styles) 13.9 kB [initial]
[rendered]
chunk {main} main-es2015.js, main-es2015.js.map (main) 61.4 kB [initial]
[rendered]
chunk {main} main-es5.js, main-es5.js.map (main) 65 kB [initial] [rendered]
chunk {polyfills-es5} polyfills-es5.js, polyfills-es5.js.map (polyfills-es5)
656 kB [initial] [rendered]
chunk {vendor} vendor-es2015.js, vendor-es2015.js.map (vendor) 2.67 MB
[initial] [rendered]
chunk {vendor} vendor-es5.js, vendor-es5.js.map (vendor) 3.11 MB [initial]
[rendered]
Date: 2020-06-04T01:31:35.612Z - Hash: d5fd9371cdc40ae353bc - Time: 210494ms
```

Here, ng build command has built our project TutorialsPoint successfully.

8. Angular CLI — ng run Command

This chapter explains the syntax, argument and options of ng run command along with an example.

Syntax

The syntax for ng run command is as follows:

```
ng run project:target[:configuration]
```

ng run command runs an Architect target with an optional custom builder configuration defined in angular.json in your project. Here project is the name of the application as defined in angular.json.

Arguments

The argument for ng run command is as follows:

Sr.No.	Argument & Syntax	Description
1	<target>	Architect target to run.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--configuration=configuration	A named builder configuration, defined in the "configurations" section of angular.json. The builder uses the named configuration to run the given target. Aliases: -c.
2	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false.

First, move to an angular project updated using **ng generate** command and then, run the command. This chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_generate.htm.

Example

An example for ng run command is given below:

```
\>Node\>TutorialsPoint> ng run TutorialsPoint:build

Generating ES5 bundles for differential loading...

ES5 bundle generation complete.


chunk {polyfills} polyfills-es2015.js, polyfills-es2015.js.map (polyfills) 141
kB [initial] [rendered]

chunk {polyfills-es5} polyfills-es5.js, polyfills-es5.js.map (polyfills-es5)
656 kB [initial] [rendered]

chunk {main} main-es2015.js, main-es2015.js.map (main) 12.9 kB [initial]
[rendered]

chunk {main} main-es5.js, main-es5.js.map (main) 15.1 kB [initial] [rendered]

chunk {runtime} runtime-es2015.js, runtime-es2015.js.map (runtime) 6.16 kB
[entry] [rendered]

chunk {runtime} runtime-es5.js, runtime-es5.js.map (runtime) 6.16 kB [entry]
[rendered]

chunk {styles} styles-es2015.js, styles-es2015.js.map (styles) 12.4 kB
[initial] [rendered]

chunk {styles} styles-es5.js, styles-es5.js.map (styles) 13.9 kB [initial]
[rendered]

chunk {vendor} vendor-es2015.js, vendor-es2015.js.map (vendor) 2.66 MB
[initial] [rendered]

chunk {vendor} vendor-es5.js, vendor-es5.js.map (vendor) 3.11 MB [initial]
[rendered]

Date: 2020-06-04T02:31:28.919Z - Hash: dd73885c28e550d01341 - Time: 13742ms
```

Here, ng build command has built our project TutorialsPoint successfully.

9. Angular CLI — ng serve Command

This chapter explains the syntax, argument and options of ng serve command along with an example.

Syntax

The syntax for ng serve command is as follows:

```
ng serve <project> [options]
ng s <project> [options]
```

ng serve command builds and serve the application. It rebuilds the application if changes occur. Here, project is the name of the application as defined in angular.json.

Argument

The argument for ng serve command is as follows:

Sr.No.	Argument & Syntax	Description
1	<project>	The name of the project to build. Can be an application or a library.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--allowedHosts	Whitelist of hosts that are allowed to access the dev server.
2	--aot=true|false	Build using Ahead of Time compilation.
3	--baseHref=baseHref	Base url for the application being built.
4	--buildEventLog=buildEventLog	EXPERIMENTAL Output file path for Build Event Protocol events
5	--commonChunk=true|false	Use a separate bundle containing code used across multiple bundles.

6	--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. Setting this explicitly overrides the "--prod" flag. Aliases: -c
7	--deployUrl=deployUrl	URL where files will be deployed.
8	--disableHostCheck=true|false	Don't verify connected clients are part of allowed hosts. Default: false
9	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
10	--hmr=true|false	Enable hot module replacement. Default: false
11	--hmrWarning=true|false	Show a warning when the --hmr option is enabled. Default: true
12	--host=host	Host to listen on. Default: localhost
13	--liveReload=true|false	Whether to reload the page on change, using live-reload. Default: true
14	--open=true|false	Opens the url in default browser. Default: false Aliases: -o
15	--optimization=true|false	Enables optimization of the build output.
16	--poll	Enable and define the file watching poll time period in milliseconds.

17	--port	Port to listen on. Default: 4200
18	--prod=true|false	Shorthand for "--configuration=production". When true, sets the build configuration to the production target. By default, the production target is set up in the workspace configuration such that all builds make use of bundling, limited tree-shaking, and also limited dead code elimination.
19	--progress=true|false	Log progress to the console while building.
20	--proxyConfig=proxyConfig	Proxy configuration file.
21	--publicHost=publicHost	The URL that the browser client (or live-reload client, if enabled) should use to connect to the development server. Use for a complex dev server setup, such as one with reverse proxies.
22	--servePath=servePath	The pathname where the app will be served.
23	--servePathDefaultWarning=true|false	Show a warning when deploy-url/base-href use unsupported serve path values. Default: true
24	--sourceMap=true|false	Output sourcemaps.
25	--ssl=true|false	Serve using HTTPS. Default: false
26	--sslCert=sslCert	SSL certificate to use for serving HTTPS.
27	--sslKey=sslKey	SSL key to use for serving HTTPS.
28	--vendorChunk=true|false	Use a separate bundle containing only vendor libraries.
29	--verbose=true|false	Adds more details to output logging.
30	--watch=true|false	Rebuild on change.

		Default: true

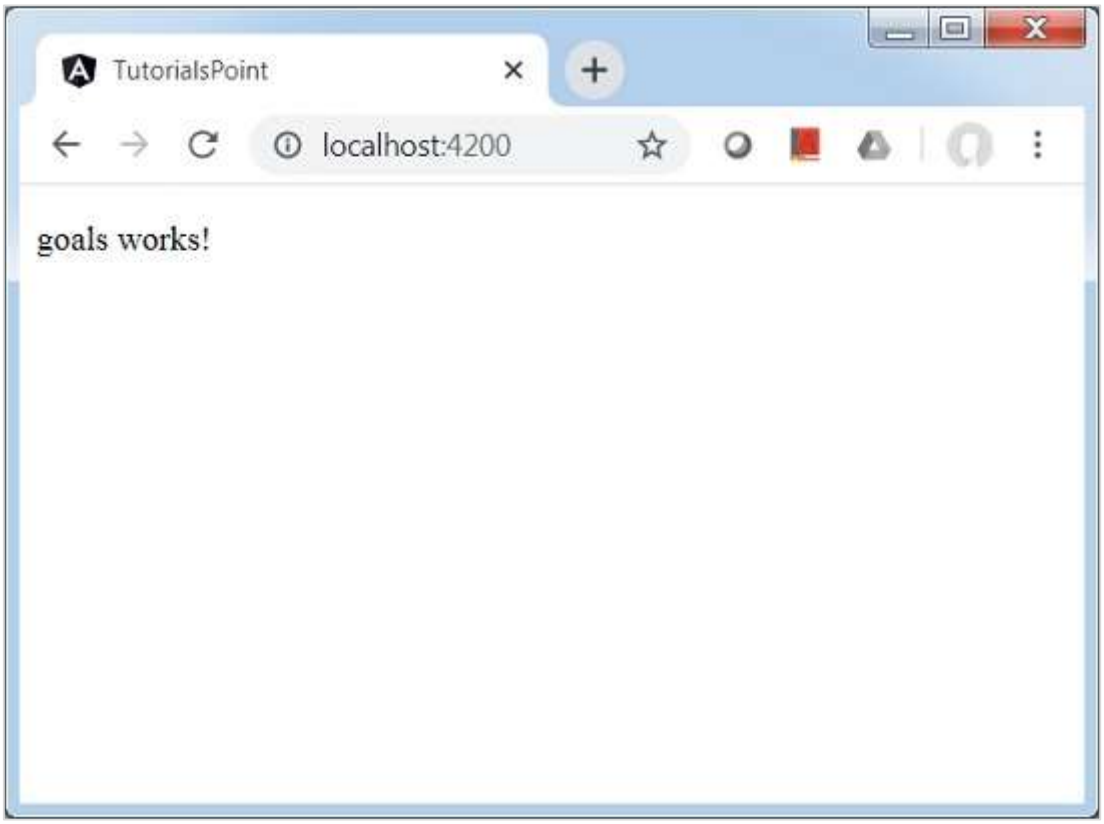
First, move to an angular project updated using **ng build** command and then, run the command. The chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Example

An example for ng serve command is given below:

```
\>Node\>TutorialsPoint> ng serve

chunk {main} main.js, main.js.map (main) 14.3 kB [initial] [rendered]

chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial]
[rendered]

chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]

chunk {styles} styles.js, styles.js.map (styles) 12.4 kB [initial] [rendered]

chunk {vendor} vendor.js, vendor.js.map (vendor) 3 MB [initial] [rendered]

Date: 2020-06-04T04:01:47.562Z - Hash: a90c5fc750c475cdc4d1 - Time: 10164ms

** Angular Live Development Server is listening on localhost:4200, open your
browser on http://localhost:4200/ **

: Compiled successfully.
```

Here ng serve command has built and serve our project TutorialsPoint successfully. Now, open http://localhost:4200 in a browser window and verify the output.



10. Angular CLI — ng lint Command

This chapter explains the syntax, argument and options of ng lint command along with an example.

Syntax

The syntax for ng lint command is as follows:

```
ng lint <project> [options]
ng l <project> [options]
```

ng lint run the linting tool on angular app code. It checks the code quality of angular project specified. It uses TSLint as default linting tool and uses the default configuration available in tslint.json file.

Argument

The argument for ng lint command is as follows:

Sr.No.	Argument & Syntax	Description
1	<project>	The name of the project to lint.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--configuration=configuration	The linting configuration to use. Aliases: -c
2	--exclude	Files to exclude from linting.
3	--files	Files to include in linting.
4	--fix=true|false	Fixes linting errors (may overwrite linted files). Default: false
5	--force=true|false	Succeeds even if there was linting errors.

		Default: false
6	--format=format	Output format (prose, json, stylish, verbose, pmd, msbuild, checkstyle, vso, fileslist). Default: prose
7	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
8	--silent=true|false	Show output text. Default: false
9	--tsConfig=tsConfig	The name of the TypeScript configuration file.
10	--tslintConfig=tslintConfig	The name of the TSLint configuration file.
11	--typeCheck=true|false	Controls the type check for linting. Default: false

First move to an angular project updated using **ng build** command, which is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Update goals.component.html and goals.component.ts as follows:

goals.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
   selector: 'app-goals',
   templateUrl: './goals.component.html',
   styleUrls: ['./goals.component.css']
})
export class GoalsComponent implements OnInit {
   title = 'Goal Component'
   constructor() { }
   ngOnInit(): void {
   }
}
```

goals.component.html

```
<p>{{title}}</p>
```

Now, run the linting command.

Example

An example for ng lint command is given below:

```
\>Node\>TutorialsPoint> ng lint
Linting "TutorialsPoint"...


ERROR: D:/Node/TutorialsPoint/src/app/goals/goals.component.ts:9:27 - Missing
semicolon
ERROR: D:/Node/TutorialsPoint/src/app/goals/goals.component.ts:13:2 - file
should end with a newline


Lint errors found in the listed files.
```

Here, ng lint command has checked the code quality of application and prints linting status.

Now, correct the errors in goals.component.ts.

goals.component.ts

```
import { Component, OnInit } from '@angular/core';


@Component({
   selector: 'app-goals',
   templateUrl: './goals.component.html',
   styleUrls: ['./goals.component.css']
})
export class GoalsComponent implements OnInit {
   title = 'Goal Component';
   constructor() { }
   ngOnInit(): void {
   }
}
```

Now, run the linting command.

Example

An example is stated below:

```
\>Node\>TutorialsPoint> ng lint
Linting "TutorialsPoint"...
All files pass linting.
```

11. Angular CLI — ng test Command

This chapter explains the syntax, argument and options of ng test command along with an example.

Syntax

The syntax for ng test command is as follows:

```
ng test <project> [options]
ng t <project> [options]
```

ng test run the unit test cases on angular app code.

Argument

The argument for ng test command is as follows:

Sr.No.	Argument & Syntax	Description
1	<project>	The name of the project to test.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--browsers=browsers	Override which browsers tests are run against.
2	--codeCoverage=true|false	Output a code coverage report. Default: false
3	--codeCoverageExclude	Globs to exclude from code coverage.
4	--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. Setting this explicitly overrides the "--prod" flag

		Aliases: -c
5	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
6	--include	Globs of files to include, relative to workspace or project root. There are 2 special cases: • when a path to directory is provided, all spec files ending ".spec.@(ts|tsx)" will be included. • when a path to a file is provided, and a matching spec file exists it will be included instead.
7	--karmaConfig=karmaConfig	The name of the Karma configuration file.
8	--main=main	The name of the main entry-point file.
9	--poll	Enable and define the file watching poll time period in milliseconds.
10	--polyfills=polyfills	The name of the polyfills file.
11	--preserveSymlinks=true|false	Do not use the real path when resolving modules. Default: false
12	--prod=true|false	Shorthand for "--configuration=production". When true, sets the build configuration to the production target. By default, the production target is set up in the workspace configuration such that all builds make use of bundling, limited tree-shaking, and also limited dead code elimination.
13	--progress=true|false	Log progress to the console while building.

14	--reporters	Karma reporters to use. Directly passed to the karma runner.
15	--sourceMap=true|false	Output sourcemaps. Default: true
16	--tsConfig=tsConfig	The name of the TypeScript configuration file.
17	--watch=true|false	Run build when files change.
18	--webWorkerTsConfig=webWorkerTsConfig	TypeScript configuration for Web Worker modules.

First move to an angular project updated using **ng build** command. The link for this chapter is https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Now, run the test command.

Example

An example for ng test command is given below:

```
\>Node\>TutorialsPoint> ng test

...

WARN: ''app-goals' is not a known element:

1. If 'app-goals' is an Angular component, then verify that it is part of this
module.

2. If 'app-goals' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the
'@NgModule.schemas' of this component to suppress this message.'

Chrome 83.0.4103 (Windows 7.0.0): Executed 0 of 4 SUCCESS (0 secs / 0 secs)

...

AppComponent should render title FAILED

   TypeError: Cannot read property 'textContent' of null

      at <Jasmine>

      at UserContext.<anonymous>
(http://localhost:9876/_karma_webpack_/src/app/app.component.spec.ts:33:51)

            ...

Chrome 83.0.4103 (Windows 7.0.0): Executed 1 of 4 (1 FAILED) (0 secs / 0.203
secs)

...

Chrome 83.0.4103 (Windows 7.0.0): Executed 2 of 4 (1 FAILED) (0 secs / 0.221

secs)
```

```
...
Chrome 83.0.4103 (Windows 7.0.0): Executed 4 of 4 (1 FAILED) (0 secs / 0.244
sec
Chrome 83.0.4103 (Windows 7.0.0): Executed 4 of 4 (1 FAILED) (0.282 secs /
0.244
 secs)
TOTAL: 1 FAILED, 3 SUCCESS
```

Now to fix failures, update the app.component.spec.ts

app.component.spec.ts

```
import { TestBed, async } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
   beforeEach(async(() => {
      TestBed.configureTestingModule({
         imports: [
            RouterTestingModule
         ],
         declarations: [
            AppComponent
         ],
      }).compileComponents();
   }));

   it('should create the app', () => {
      const fixture = TestBed.createComponent(AppComponent);
      const app = fixture.componentInstance;
      expect(app).toBeTruthy();
   });
});
```

Now, run the test command.

Example

An example is given below:

```
\>Node\>TutorialsPoint> ng test
```

```
...
WARN: ''app-goals' is not a known element:
1. If 'app-goals' is an Angular component, then verify that it is part of this m
odule.
2. If 'app-goals' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@
NgModule.schemas' of this component to suppress this message.'
Chrome 83.0.4103 (Windows 7.0.0): Executed 1 of 2 SUCCESS (0 secs / 0.053 secs)
...
Chrome 83.0.4103 (Windows 7.0.0): Executed 2 of 2 SUCCESS (0.097 secs / 0.073 se
cs)
TOTAL: 2 SUCCESS
```

ng test also opens the browser and displays the test status.

12. Angular CLI — ng e2e Command

This chapter explains the syntax, argument and options of ng e2e command along with an example. Here, e2e refers to end to end.

Syntax

The syntax for ng e2e command is as follows:

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

ng e2e builds, serves an application and then, runs the end to end test cases using protractor.

Argument

The argument for ng e2e command is as follows:

Sr.No.	Argument & Syntax	Description
1	<project>	The name of the project to test.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--baseUrl=baseUrl	Base URL for protractor to connect to.
2	--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. Setting this explicitly overrides the "--prod" flag Aliases: -c
3	--devServerTarget=devServerTarget	Dev server target to run tests against.
4	--grep=grep	Execute specs whose names match the pattern, which is internally compiled to a RegExp.

5	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
6	--host=host	Host to listen on.
7	--invertGrep=true|false	Invert the selection specified by the 'grep' option. Default: false
8	--port	The port to use to serve the application.
9	--prod=true|false	Shorthand for "--configuration=production". When true, sets the build configuration to the production target. By default, the production target is set up in the workspace configuration such that all builds make use of bundling, limited tree-shaking, and also limited dead code elimination.
10	--protractorConfig=protractorConfig	The name of the Protractor configuration file.
11	--specs	Override specs in the protractor config.
12	--suite=suite	Override suite in the protractor config.
13	--webdriverUpdate=true|false	Try to update webdriver. Default: true

First move to an angular project updated using ng build command, which is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Now, run the e2e command.

Example

An example for ng e2e command is given below:

```
\>Node\>TutorialsPoint> ng e2e

...

chunk {main} main.js, main.js.map (main) 14.3 kB [initial] [rendered]

chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial]
[rendered]
```

```
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 12.4 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3 MB [initial] [rendered]
Date: 2020-06-06T04:20:15.029Z - Hash: 16f321e3d4599af26622 - Time: 20899ms
** Angular Live Development Server is listening on localhost:4200, open your
bro
wser on http://localhost:4200/ **
: Compiled successfully.
...
   workspace-project App
    x should display welcome message
      - Failed: No element found using locator: By(css selector, app-root
.content span)
   ...
      From: Task: Run it("should display welcome message") in control flow
   ...
*************************************************
*                    Failures                   *
*************************************************

1) workspace-project App should display welcome message
  - Failed: No element found using locator: By(css selector, app-root .content
span)

Executed 1 of 1 spec (1 FAILED) in 2 secs.
```

Now to fix the failures, update the app.component.html.

app.component.html

```
<div class="content" role="main">
   <span>{{ title }} app is running!</span>
</div>
<app-goals></app-goals>
<router-outlet></router-outlet>
```

Now, run the e2e command.

Example

An example for ng e2e command is given below:

```
\>Node\>TutorialsPoint> ng e2e

...

chunk {main} main.js, main.js.map (main) 14.9 kB [initial] [rendered]

chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial]
[rendered]

chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]

chunk {styles} styles.js, styles.js.map (styles) 12.4 kB [initial] [rendered]

chunk {vendor} vendor.js, vendor.js.map (vendor) 3 MB [initial] [rendered]

Date: 2020-06-06T04:28:33.514Z - Hash: 5d8bf2fc7ff59fa390b0 - Time: 10529ms

** Angular Live Development Server is listening on localhost:4200, open your
browser on http://localhost:4200/ **

: Compiled successfully.

...

   workspace-project App

      √ should display welcome message


Executed 1 of 1 spec SUCCESS in 2 secs.
```

ng e2e also opens the browser and use it to run acceptance test cases with the help of User Interface (UI).

13. Angular CLI — ng add Command

This chapter explains the syntax, argument and options of ng add command along with an example.

Syntax

The syntax for ng add command is as follows:

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

ng add a npm package to workspace.

Argument

The argument for ng add command is as follows:

Sr.No.	Argument & Syntax	Description
1	<collection>	The name of the package to be added.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--defaults=true|false	When true, disables interactive input prompts for options with a default.
2	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
3	--registry=registry	The NPM registry to use.
4	--verbose=true|false	Display additional details about internal operations during execution. Default: false

First, move to an angular project updated using ng build command, which is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Now run the add command.

Example

An example for ng add command is given below:

```
\>Node\>TutorialsPoint> ng add @angular/pwa
Installing packages for tooling via npm.
Installed packages for tooling via npm.
CREATE ngsw-config.json (620 bytes)
CREATE src/manifest.webmanifest (1352 bytes)
CREATE src/assets/icons/icon-128x128.png (1253 bytes)
CREATE src/assets/icons/icon-144x144.png (1394 bytes)
CREATE src/assets/icons/icon-152x152.png (1427 bytes)
CREATE src/assets/icons/icon-192x192.png (1790 bytes)
CREATE src/assets/icons/icon-384x384.png (3557 bytes)
CREATE src/assets/icons/icon-512x512.png (5008 bytes)
CREATE src/assets/icons/icon-72x72.png (792 bytes)
CREATE src/assets/icons/icon-96x96.png (958 bytes)
UPDATE angular.json (3803 bytes)
UPDATE package.json (1332 bytes)
UPDATE src/app/app.module.ts (682 bytes)
UPDATE src/index.html (482 bytes)
√ Packages installed successfully.
```

14. Angular CLI — ng analytics Command

This chapter explains the syntax, argument and options of ng analytics command along with an example.

Syntax

The syntax for ng analytics command is as follows:

```
ng analytics <settingOrProject> <projectSetting> [options]
```

ng analytics command configures Angular CLI usage metrics.

Arguments

The arguments for ng analytics command is as follows:

Sr.No.	Argument & Syntax	Description
1	<settingOrProject>=on|off|ci|project|prompt	Directly enables or disables all usage analytics for the user, or prompts the user to set the status interactively, or sets the default status for the project. Following are the details of options. • on – Enables analytics gathering and reporting for the user. • off – Disables analytics gathering and reporting for the user. • ci – Enables analytics and configures reporting for use with Continuous Integration, which uses a common CI user. • prompt – Prompts the user to set the status interactively. • project – Sets the default status for the project to the projectSetting value, which can be any of the other values. The projectSetting argument is

		ignored for all other values of settingOrProject.
2	<projectSetting>=on|off|prompt	Sets the default analytics enablement status for the project.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false

First move to an angular project updated using ng build command, which is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Now, run the analytics command.

Example

An example for ng analytics command is given below:

```
\>Node\>TutorialsPoint> ng analytics off
```

15. Angular CLI — ng config Command

This chapter explains the syntax, arguments and options of ng config command along with an example.

Syntax

The syntax for ng config command is as follows:

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

ng config command retrieves or sets angular configuration values in angular.json.

Arguments

The arguments for ng config command is as follows:

Sr.No.	Argument & Syntax	Description
1	<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.
2	<value>	If provided, a new value for the given configuration key.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--global=true|false	When true, accesses the global configuration in the caller's home directory. Default: false Aliases: -g
2	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false

First, move to an angular project updated using ng build command. This chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Now, run the config command.

Example

An example for ng config command is given below:

```
\>Node\>TutorialsPoint> ng config projects.TutorialsPoint.projectType
application
```

16. Angular CLI — ng doc Command

This chapter explains the syntax, arguments and options of ng doc command along with an example.

Syntax

The syntax for ng doc command is as follows:

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

ng doc command opens the official Angular documentation in a browser, and searches for a given keyword.

Arguments

The arguments for ng doc command is as follows:

Sr.No.	Argument & Syntax	Description
1	<keyword>	The keyword to search for, as provided in the search bar in angular.io.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false

First move to an angular project updated using ng build command. This chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

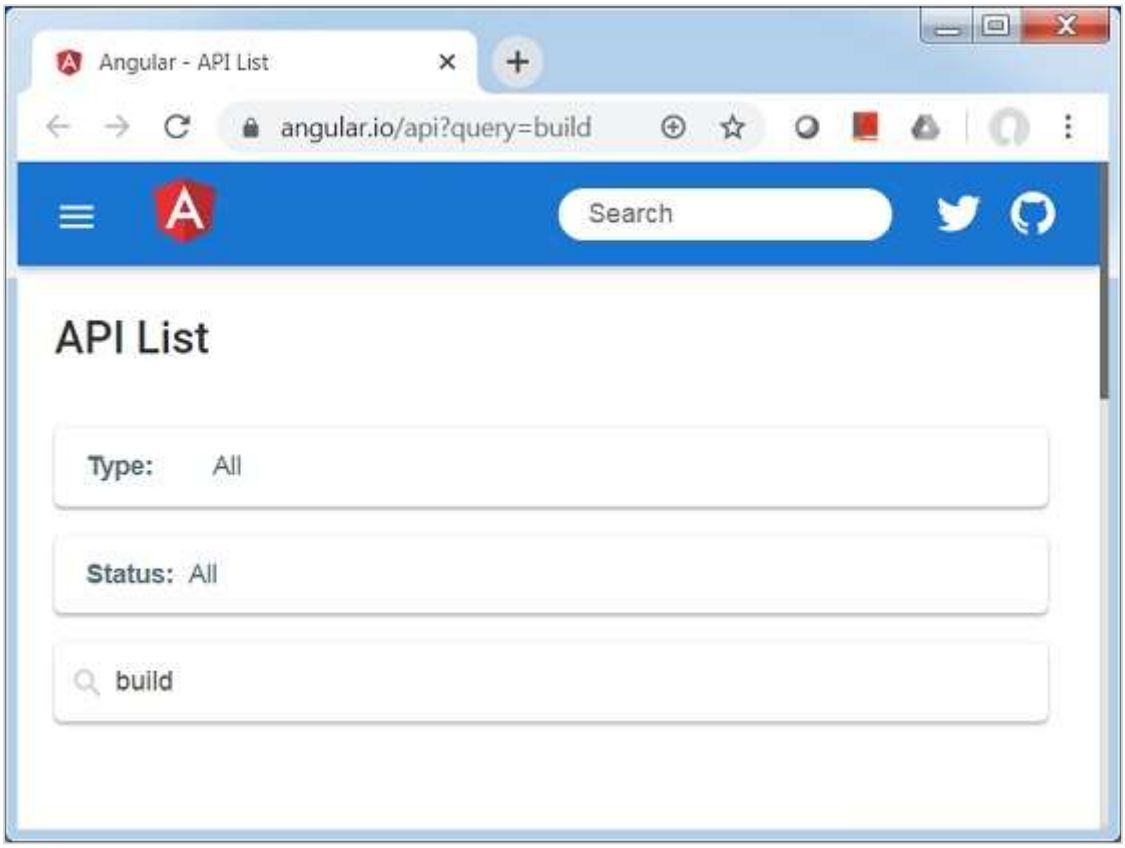
Now, run the doc command.

Example

An example for ng doc command is given below:

```
\>Node\>TutorialsPoint> ng doc build
```

Now, a browser window will open and search the relevant keyword.



17. Angular CLI — ng update Command

This chapter explains the syntax, argument and options of ng update command along with an example.

Syntax

The syntax for ng update command is as follows:

```
ng update [options]
```

ng update command updates the application and its dependencies.

Argument

The argument for ng update command is as follows:

Sr.No.	Argument & Syntax	Description
1	<keyword>	The keyword to search for, as provided in the search bar in angular.io.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--all=true|false	Whether to update all packages in package.json. Default: false
2	--allowDirty=true|false	Whether to allow updating when the repository contains modified or untracked files.
3	--createCommits=true|false	Create source control commits for updates and migrations. Default: false Aliases: -C
4	--force=true|false	If false, will error out if installed packages are incompatible with the update.

		Default: false
5	--from=from	Version from which to migrate from. Only available with a single package being updated, and only on migration only.
6	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
7	--migrateOnly=true|false	Only perform a migration, does not update the installed version.
8	--next=true|false	Use the largest version, including beta and RCs. Default: false
9	--packages	The names of package(s) to update.
10	--to=to	Version up to which to apply migrations. Only available with a single package being updated, and only on migrations only. Requires from to be specified. Default to the installed version detected.
11	--verbose=true|false	Display additional details about internal operations during execution. Default: false

First move to an angular project updated using ng build command. The chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm.

Now, run the update command. ng will update dependencies using npm.

Example

An example for ng update command is given below:

```
\>Node\>TutorialsPoint> ng update
Using package manager: 'npm'
Collecting installed dependencies...
Found 31 dependencies.
   We analyzed your package.json and everything seems to be in order. Good
work!
```

18. Angular CLI — ng xi18n Command

This chapter explains the syntax, arguments and options of ng xi18n command along with an example.

Syntax

The syntax for ng xi18n command is as follows:

```
ng xi18n <project> [options]
ng i18n-extract <project> [options]
```

ng xi18n command extracts i18n messages from source code.

Argument

The argument for ng xi18n command is as follows:

Sr.No.	Argument & Syntax	Description
1	<project>	The name of the project. It can be an application or library.

Options

Options are optional parameters.

Sr.No.	Option & Syntax	Description
1	--browserTarget=browserTarget	Target to extract from.
2	--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. Setting this explicitly overrides the "--prod" flag. Aliases: -c
3	--createCommits=true|false	Create source control commits for updates and migrations. Default: false Aliases: -C

4	--format=xmb|xlf|xlif|xliff|xlf2|xliff2	Output format for the generated file. Default: xlf
5	--help=true|false|json|JSON	Shows a help message for this command in the console. Default: false
6	--outFile=outFile	Name of the file to output.
7	--outputPath=outputPath	Path where output will be placed.
8	--prod=true|false	Shorthand for "--configuration=production". When true, sets the build configuration to the production target. By default, the production target is set up in the workspace configuration such that all builds make use of bundling, limited tree-shaking, and also limited dead code elimination.
9	--progress=true|false	Log progress to the console. Default: true

First move to an angular project updated using **ng build** command. The chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_build.htm. Update the app.component.html as follows:

app.component.spec.ts

```
<div class="content" role="main">
   <span i18n>app is running!</span>
</div>
<app-goals></app-goals>
<router-outlet></router-outlet>
```

Now, run the xi18n command.

Example

An example for ng xi18n command is given below:

```
\>Node\>TutorialsPoint> ng xi18n
```

Add localization support.

```
\>Node\>TutorialsPoint> ng add @angular/localize
```

```
Installing packages for tooling via npm.
Installed packages for tooling via npm.
UPDATE src/polyfills.ts (3064 bytes)
```

Now, ng will create a messages.xlf file in root folder which is an industry standard translation file.

messages.xlf

```
<?xml version="1.0" encoding="UTF-8" ?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
   <file source-language="en-US" datatype="plaintext" original="ng2.template">
      <body>
         <trans-unit id="6226cbeebaffaec0342459915ef7d9b0e9e92977"
datatype="html">
            <source>app is running!</source>
            <context-group purpose="location">
               <context context-
type="sourcefile">src/app/app.component.html</context>
               <context context-type="linenumber">2</context>
            </context-group>
         </trans-unit>
      </body>
   </file>
</xliff>
```

19. Angular CLI — Code Coverage

This chapter explains the syntax of code coverage command along with an example.

Syntax

The syntax for code coverage command is as follows:

```
ng test <project> --codeCoverage=true
```

ng test command allows to check code coverage using the test cases written. See the example below.

Move to an angular project updated using **ng xi18n** command. This chapter is available at https://www.tutorialspoint.com/angular_cli/angular_cli_ng_xi18n.htm. Now, run the test with codeCoverage command.

Example

An example for ng code coverage command is given below:

```
\>Node\>TutorialsPoint> ng test --codeCoverage=true

10% building 2/2 modules 0 active07 06 2020 15:21:46.292:WARN [karma]: No
captur

ed browser, open http://localhost:9876/

07 06 2020 15:21:46.299:INFO [karma-server]: Karma v4.4.1 server started at
http

://0.0.0.0:9876/

07 06 2020 15:21:46.300:INFO [launcher]: Launching browsers Chrome with
concurre

ncy unlimited

07 06 2020 15:21:46.312:INFO [launcher]: Starting browser Chrome

07 06 2020 15:21:55.456:WARN [karma]: No captured browser, open
http://localhost

:9876/

07 06 2020 15:21:55.533:INFO [Chrome 83.0.4103 (Windows 7.0.0)]: Connected on
so

cket gJgRaX_rXI6ZqoAiAAAA with id 261512

...

Chrome 83.0.4103 (Windows 7.0.0): Executed 1 of 2 SUCCESS (0 secs / 0.053 secs)

...

Chrome 83.0.4103 (Windows 7.0.0): Executed 2 of 2 SUCCESS (0.107 secs / 0.082
se
```

```
cs)
TOTAL: 2 SUCCESS


============================== Coverage summary
==============================


Statements   : 100% ( 8/8 )
Branches     : 100% ( 0/0 )
Functions    : 100% ( 3/3 )
Lines        : 100% ( 6/6 )
================================================================================
=
```

Now, ng test command has created a coverage folder within the TutorialsPoint, the project folder and has prepared the coverage report in html format available as \>Node\>TutorialsPoint\>coverage\>TutorialsPoint\>index.html.