

# Angular

Benjamin Longearet - @blongearet

# Part 6

# Angular

# Architecture

- Angular CLI
- ng new
- angular.json
- ng serve
- ng generate
- ng test
- ng add
- ng build
- ng doc
- Config files

## Part 6 - Angular Architecture

# Angular CLI

- Angular CLI is a command-line interface (CLI)
- Developed to automate tasks
- It can:
  - Create a new app
  - Run a server w/ LiveReload
  - Add artifacts to our app
  - Run unit tests & end-to-end tests (e2e)
  - Build the app for different environments

## Part 6 - Angular Architecture

# Angular CLI

- CLI Website: <https://cli.angular.io>
- CLI Docs: <https://github.com/angular/angular-cli>
- Install it:

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

- To check if the package is well installed, run:

```
$ ng -v
```

ng new

## Part 6 - Angular Architecture

# ng new

- It will generate all the boilerplate code that you need
- In a terminal, type:  

```
$ ng new myApplicationName
```
- ng new docs: <https://github.com/angular/angular-cli/wiki/new>

angular.json

## Part 6 - Angular Architecture

# angular.json

- It grab all the configuration of our angular project
- Can list schematics to produce file creation (ngrx, angular-bootstrap, angular pwa, etc).
- angular.json docs:  
<https://github.com/angular/angular-cli/wiki/angular-workspace>



ng serve

## Part 6 - Angular Architecture

# ng serve

- It serve files through a local HTTP server
- Ensure to watch files and compile/reload at each changes.
- In a terminal, type (-o mean open browser):  
`$ ng serve -o`
- ng server docs: <https://github.com/angular/angular-cli/wiki/serve>

ng generate

## Part 6 - Angular Architecture

# ng generate

- It use schematics to generate code and update if needed existing files
- All Angular artefacts got their schematics built-in
- In a terminal, type:  

```
$ ng generate <type> <path>
```
- ng generate docs: <https://github.com/angular/angular-cli/wiki/generate>

## Part 6 - Angular Architecture

# ng generate

- Create a component called “foo” and add it to the root module  
`$ ng generate component foo`
- Create a component called “foo” without spec files (tests)  
`$ ng g c foo --spec false`
- Create a module called “feature”  
`$ ng generate module feature`
- Create a component called “bar” into the module called “feature”  
`$ ng generate component bar -m feature`

## Part 6 - Angular Architecture

# ng generate

- Custom schematics
  - <https://github.com/ngrx/platform/tree/master/docs/schematics>
  - <https://blog.angular.io/schematics-an-introduction-dc1dfbc2a2b2>

ng test

## Part 6 - Angular Architecture

# ng test

- It run tests into the browser
- It use Karma & watching changes to be reloaded
- In a terminal, type:  
`$ ng test`
- ng test docs: <https://github.com/angular/angular-cli/wiki/test>



ng add

## Part 6 - Angular Architecture

# ng add

- New in Angular 6
- It allows to install an application/library easily into our angular project:

```
$ ng add @angular/pwa
```

```
$ ng add @ng-bootstrap/schematics
```

- ng add docs: <https://github.com/angular/angular-cli/wiki/add>

ng build

## Part 6 - Angular Architecture

# ng build

- It allows to build the application for a specific environment

```
$ ng build --prod
```

```
$ ng build
```

- ng build docs: <https://github.com/angular/angular-cli/wiki/build>

ng doc

## Part 6 - Angular Architecture

# ng doc

- It opens the Angular website on the related doc:

```
$ ng doc component
```

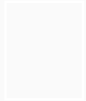
- ng doc docs: <https://github.com/angular/angular-cli/wiki/doc>

config files

## Part 6 - Angular Architecture

# config files: angular.json

- We already talk about it





## Part 6 - Angular Architecture

# config files: package.json

- It contains packages to build & run our application (deps & devDeps)
- It provides “script” that you can call by using:  
`$ npm run scriptName`
- It provides meta data about the current project
  - Name
  - Version
  - etc.

## Part 6 - Angular Architecture

# config files: tsconfig.json

- It contains the TypeScript compiler configuration,
- tsconfig.json doc:

<https://www.typescriptlang.org/docs/handbook/tsconfig-json.html>

## Part 6 - Angular Architecture

# config files: tslint.json

- Angular has a linting tool that checks the TypeScript code for programmatic & stylistic errors.
- TypeScript tslint file:  
<https://github.com/Microsoft/TypeScript/blob/master/tslint.json>
- Configuring TSLint: <https://palantir.github.io/tslint/usage/configuration/>