GUIDE TO CODE AND RUN ANGULAR PROJECT

Author: *By VAMOS Team*
Date: January 17, 2024

I. Prerequisite

- Local machine must have:

- Node: 20.10.0
- Angular: 17.0.10
  - Run this command to install (Must have node) -> **npm install -g @angular/cli 17.0.10**

II. Create the project

- Use command line provided by AngularCLI to create a project:

ng new <name> [options] or **ng n <name> [option]**

- <name>: project name
- [option]: some options (flags) that we want to automatically config while creating project
  - List of available option: https://angular.io/cli/new

Note:

- Since Angular 17, Angular projects will be generated as "**standalone**" by default. To avoid that you can add a flag for it before creating projects:
  - **ng new <name> --standalone false**
- By default, AngularCLI will set the "**strict**" mode on. Strict mode **enforces stricter type-checking and coding practices** in your Angular application. You can turn it off (**Optional**) by using:
  - **ng new <name> --strict=false**
- You can combine multiple flags in one command
  - **ng new <name> --strict=false --standalone false [options]**

III. How to run project

- With a project that was created by AngularCLI, we only need to run this command:
  - **ng serve**
- But with a project pulled from git or a project that is missing a node_modules folder. We need to run additional command line before run the project:
  - **npm install.**

IV. Generate items in project

- It's recommended to use AngularCLI to generate items

ng generate <schematic> <name> or **ng g <schematic> <name>**

➢ List of available schematic: https://angular.io/cli/generate

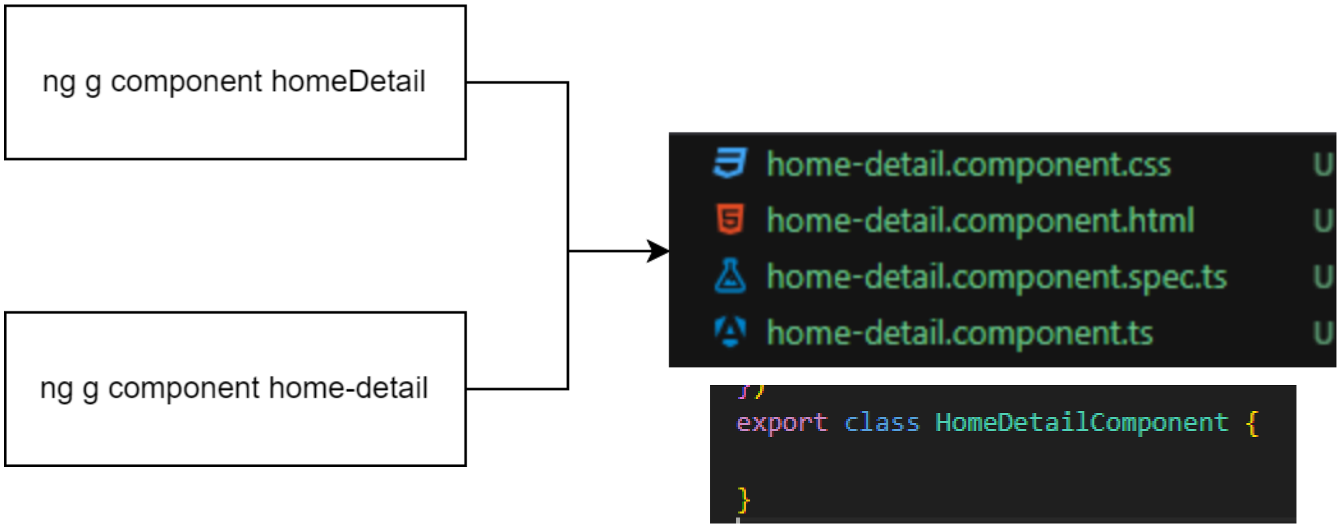
- Angular support these schematics (and their shortcut name):
  - `class (cl)`
  - `guard (g)`
  - `module (m)`
  - `component (c)`
  - `interceptor`

  - pipe (p)
  - directive (d)
  - interface (i)
  - service (s)
  - enum (e)

Note:

- In <name>, you can specify the directory you want to create in.
  Example:
  ng generate component modules/home
  -> This will automatically create a new folder **home** containing **HomeComponent** inside **modules** directory
- If <name> using camelCase or separate with '-', it will generate the same result.

Example:



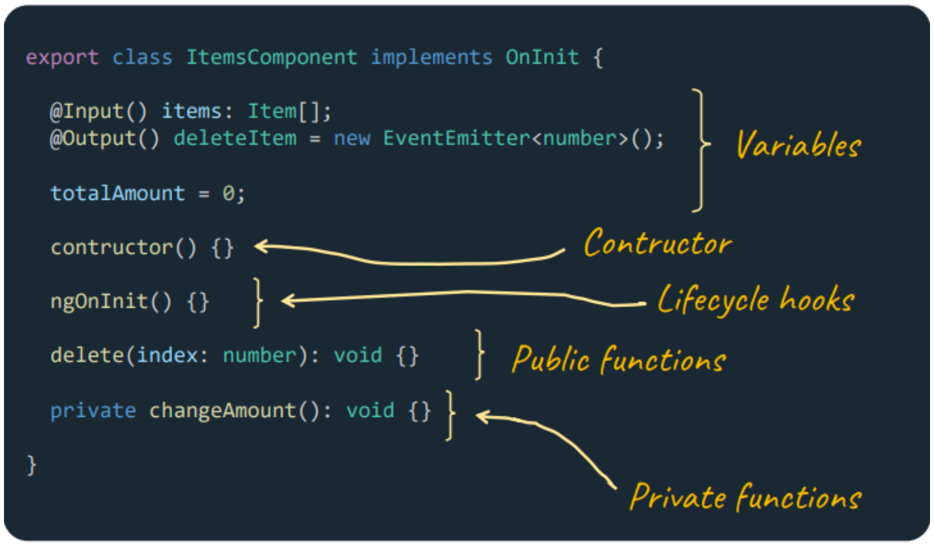
```
home-detail.component.css        U
home-detail.component.html       U
home-detail.component.spec.ts    U
home-detail.component.ts         U
```

```
export class HomeDetailComponent {

}
```

V. Convention

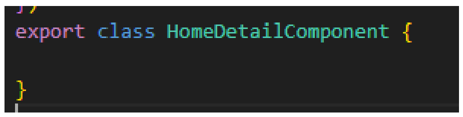
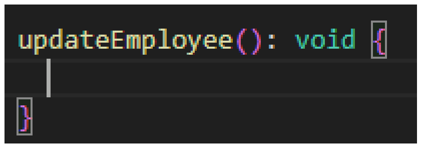
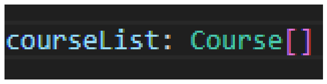
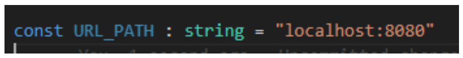
1. Order of variables and functions

Order should follow this order:



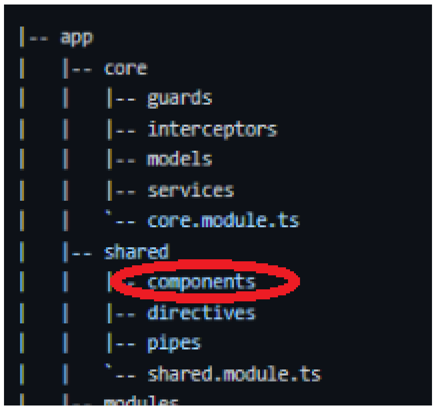
```
export class ItemsComponent implements OnInit {

  @Input() items: Item[];
  @Output() deleteItem = new EventEmitter<number>();

  totalAmount = 0;

  contructor() {}

  ngOnInit() {}

  delete(index: number): void {}

  private changeAmount(): void {}

}
```

2. Naming convention

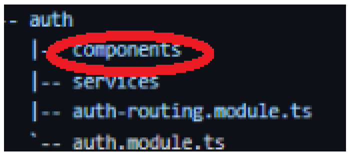
Type	Rules	Examples
classes	UpperCamelCase. Nouns or noun phrases Always end with the purpose of the class (Ex: Component, Service, …)	 `export class HomeDetailComponent {` `}`
methods	lowerCamelCase. Verbs or verb phrases Must always have a return type	 `updateEmployee(): void {` `}`
variables	lowerCamelCase Must have a return type	 `courseList: Course[]`
constants	All uppercase with words separated by underscores Must have a return type	 `const URL_PATH : string = "localhost:8080"`

1. Component break-down

- If a component is reused in many places in your application. You should create a new component for it and put it in **shared components** folder:


```
|-- app
|   |-- core
|   |   |-- guards
|   |   |-- interceptors
|   |   |-- models
|   |   |-- services
|   |   `-- core.module.ts
|   |-- shared
|   |   |-- components
|   |   |-- directives
|   |   |-- pipes
|   |   `-- shared.module.ts
```

- If a page contains hundreds lines of code or the code is complex. You should create a new component and put it in **components** folder of the **same** directory:


```
- auth
  |-- components
  |-- services
  |-- auth-routing.module.ts
  `-- auth.module.ts
```

References:

- 2021_08_09_Angular_Tips_TrungAtom_1703140899779
- https://angular.io/cli/generate
- CodeConvention_Vamos2024