
LES COMPONENTS

ULT – DÉPARTEMENT GÉNIE INFORMATIQUE



Module: Projet Framework et Développement

Public cible: 5° Génie Logiciel

Houcem Hedhly

houssem.hedhli@ult-tunisie.com

2021/2022

PLAN

- I. Définition et utilité
- II. Création

I. DÉFINITION ET UTILITÉ

- Au moins un **Root Component** qui connecte les autres composants avec le DOM.
- Component = Comportement (classe TypeScript) + Template (HTML) + Style (CSS)
- Le décorateur @Component offre un ensemble de meta-données.
- Cycle de vie : création, mise à jour, destruction.
- Syntaxe de la Template: Data binding (Interpolation, Property, Event, Model) ,
Directives (*ngFor, *ngIf), Pipes (Dates, Currency, etc.)

AppComponent

[Home](#)

[Users](#)

[Logout](#)

Contenu

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore ms consequat. Duis aute irure dolor in

Side menu

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

Component

Template



Class



Metadata

View
HTML

Code
TypeScript
Data &
Methods

Information
Decorator

app.component.html

app.component.ts

II. CRÉER UN NOUVEAU COMPONENT

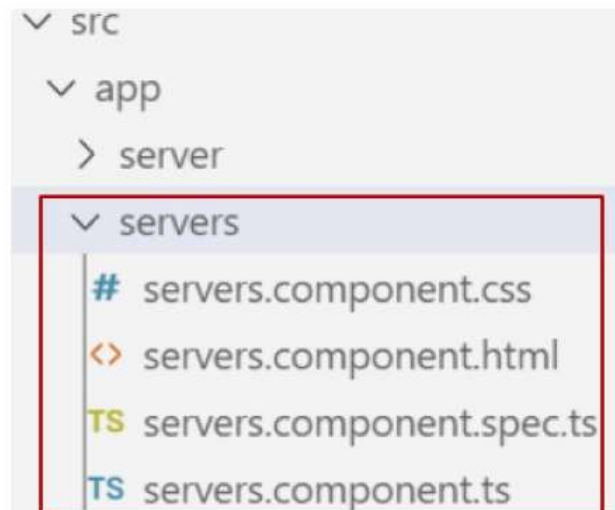
- On peut créer un component à partir la ligne de commande en utilisant le CLI

> ng generate component servers

Ou

> ng g c servers

- Ceci va générer automatiquement un dossier "servers" ainsi que le contenu associé
- Ca va générer aussi toutes les dépendances nécessaires



```

▼ src
  ▼ app
    > server
      ▼ servers
        # servers.component.css
        <> servers.component.html
        TS servers.component.spec.ts
        TS servers.component.ts

```

The image shows a file explorer view of the generated project structure. The 'servers' folder is highlighted with a red box, showing its contents: 'servers.component.css', 'servers.component.html', 'servers.component.spec.ts', and 'servers.component.ts'.



src > app > servers > TS servers.component.ts > ...

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4      selector: 'app-servers',
5      templateUrl: './servers.component.html',
6      styleUrls: ['./servers.component.css']
7  })
8  export class ServersComponent implements OnInit {
9
10     constructor() { }
11
12     ngOnInit() {
13     }
14
15 }
16
```

OPEN EDITORS 1 UNSAVED

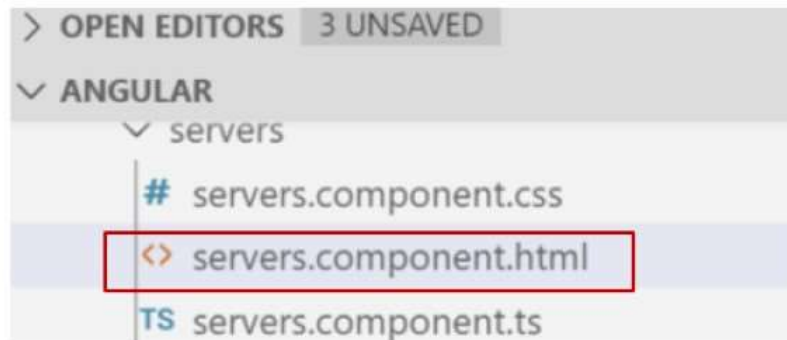
MY-FIRST-APP

- app
 - server
 - servers
 - # servers.component.css
 - <> servers.component.html
 - TS servers.component.ts
 - # app.component.css
 - <> app.component.html
 - TS app.component.spec.ts
 - TS app.component.ts
 - TS app.module.ts
 - assets
 - environments
 - ★ favicon.ico
 - <> index.html
 - TS main.ts
 - TS polyfills.ts

src > app > TS app.module.ts > AppModule

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { ServerComponent } from './server/server.component';
6 import { ServersComponent } from './servers/servers.component';
7
8 @NgModule({
9   declarations: [
10     AppComponent,
11     ServerComponent,
12     ServersComponent
13   ],
14   imports: [
15     BrowserModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
21
```


Au niveau du template du component « Servers »



my-first-app > src > app > servers > <> servers.component.html > ...

1 <h2>Servers Component</h2>

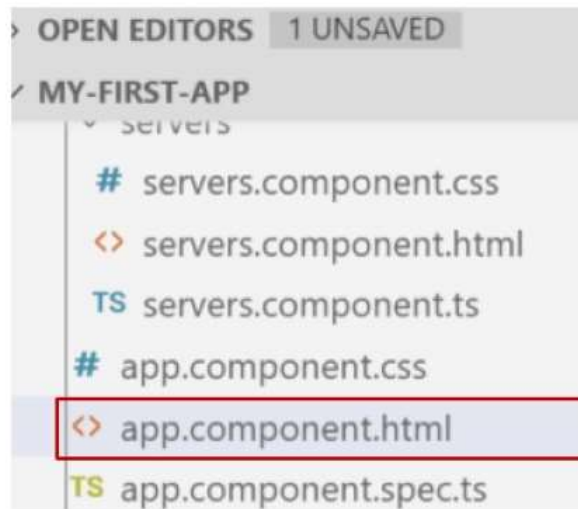
2

3

4

5

Au niveau du template du component « appComponent »



src > app > <> app.component.html > ...

1 <h3>I'm in the app component</h3>

2 <hr>

3 <app-servers></app-servers>

4

I'm in the app component

Servers Component

```
src > app > servers > TS servers.component.ts > ...
```

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-servers',
5    templateUrl: './servers.component.html',
6    styleUrls: ['./servers.component.css']
7  })
8  export class ServersComponent implements OnInit {
9
10     constructor() { }
11
12     ngOnInit() {
13     }
14
15 }
16
```

@Component decorator: Il fournit des métadonnées sur le composant, y compris ses modèles, ses styles et un sélecteur

selector : Utilisé pour identifier le component.
Par convention, ils commencent par le préfixe **-app**, suivi du nom du composant

templateUrl & styleUrls : définissent les fichiers qui vont contenir le code HTML et CSS.