
HOW ANGULAR WORKS ?

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. Principes de base
- II. TypeScript
- III. Comment ça fonctionne ?



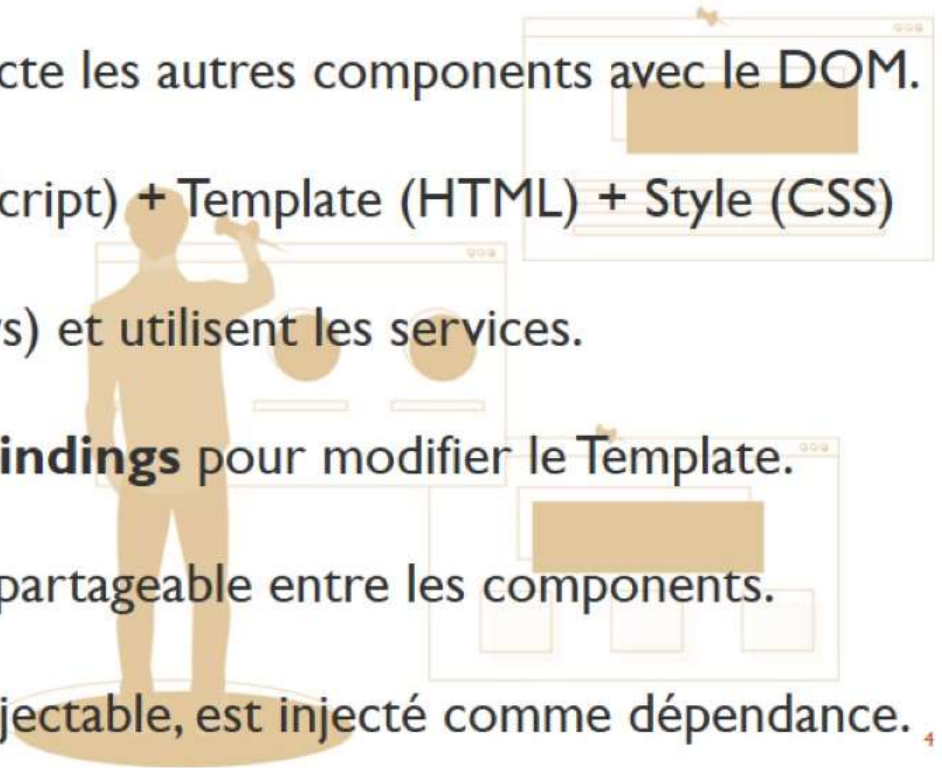
I. PRINCIPE DE BASE

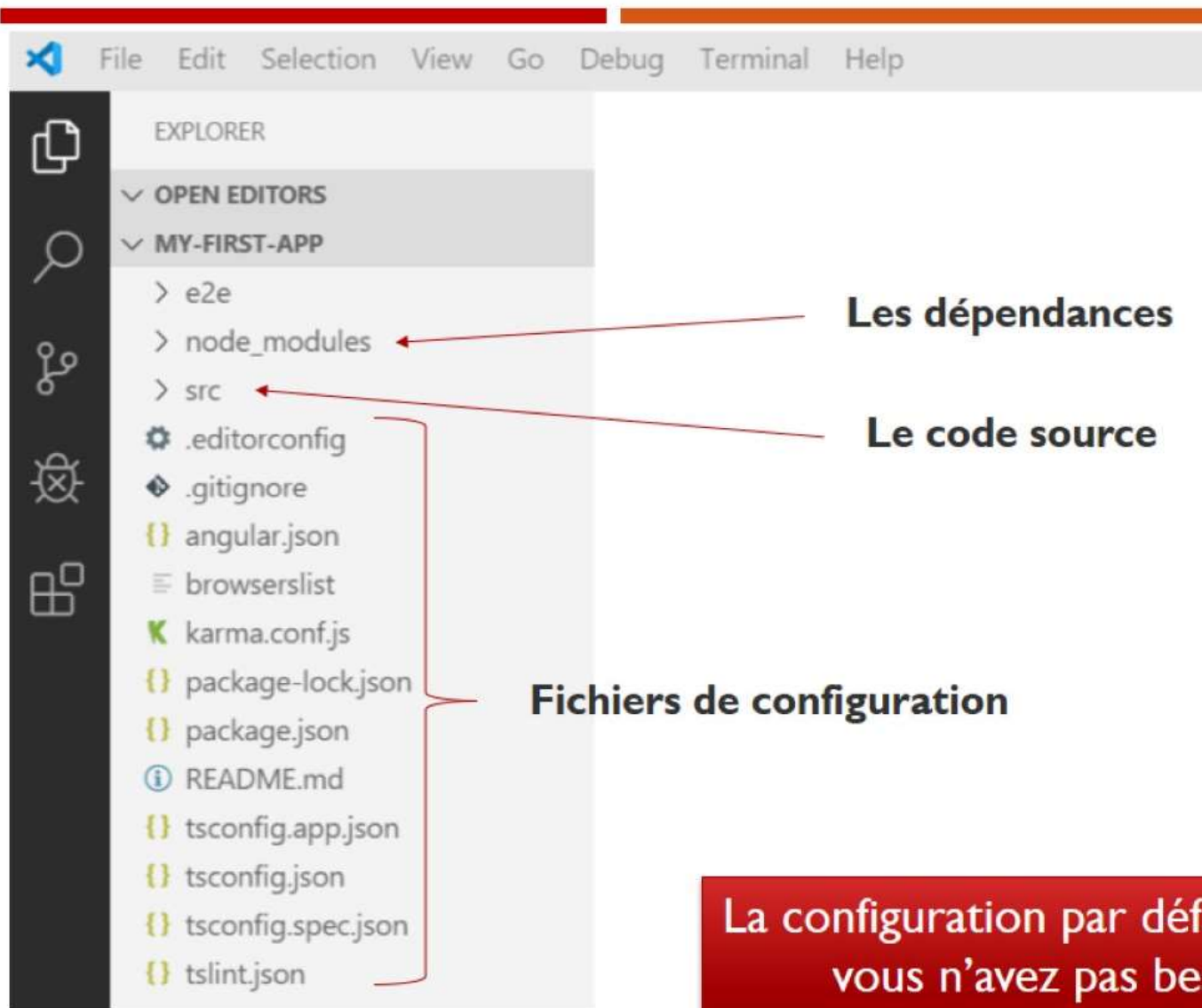
- Application Angular = ensemble de **Components** organisés dans des **Modules**.
- Au moins un **Root Module** (appelé AppModule), et typiquement plusieurs **Feature Modules** (une collection de code relative à une entité fonctionnelle).
- Les Modules sont reconnus avec le décorateur **@NgModule**.
- Fonctionnalités des Modules peuvent être importées et/ou exportées.
- Avantages: diviser pour régner, réutilisabilité, lazy-loading.

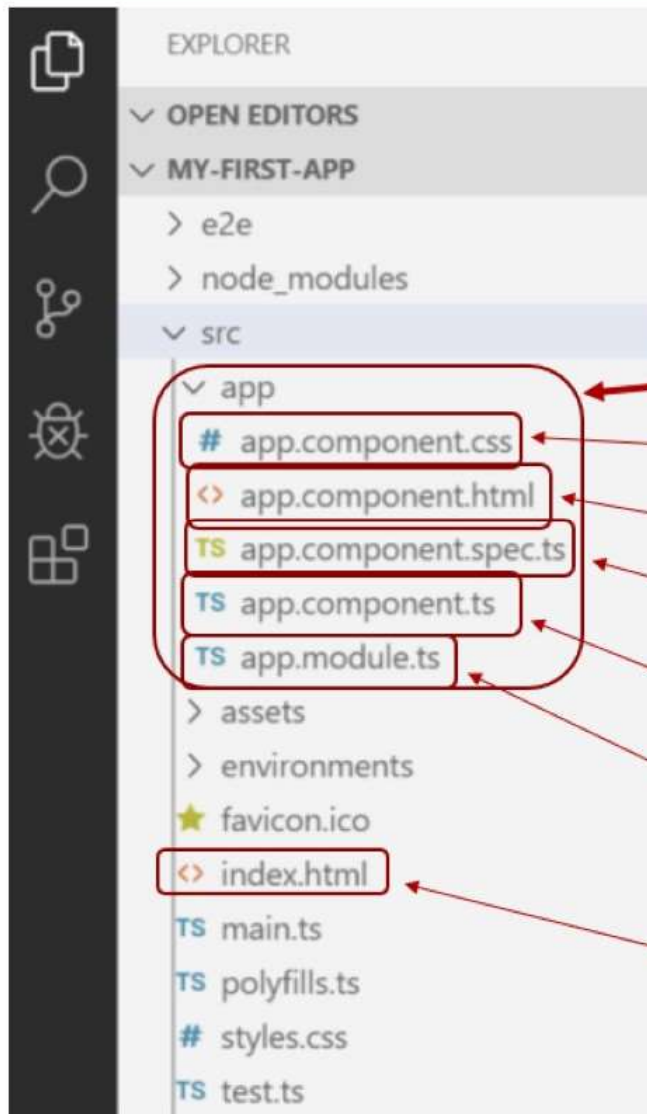


I. PRINCIPE DE BASE

- Au moins un **Root Component** qui connecte les autres composants avec le DOM.
- Component = Comportement (classe TypeScript) + Template (HTML) + Style (CSS)
- Components définissent les interfaces (Views) et utilisent les services.
- Evaluation des **Directives**, résolution des **Bindings** pour modifier le Template.
- **Service** = Logique métier indépendante et partageable entre les composants.
- Un service, reconnu avec le décorateur `@Injectable`, est injecté comme dépendance.







Votre app

Style CSS du component principal

Contenu HTML du component principal

Pour les tests

Le contenu métier du **Component principal**

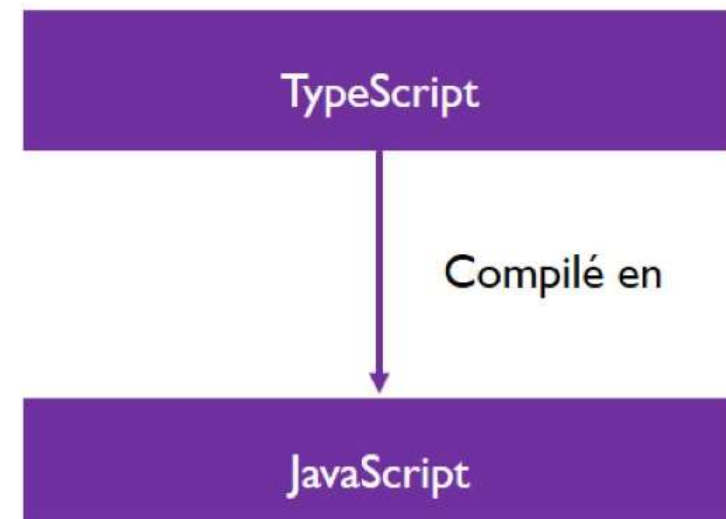
Le contenu métier du **Module principal**

La page HTML (SPA)

II. TYPESCRIPT



- Langage de programmation open-source développé par Microsoft
- Améliorer et sécuriser la production de code JavaScript
- Sur-ensemble (superset) de JavaScript
- Plus de fonctionnalités que JavaScript
 - Typage des variables et des fonctions
 - Classes et interfaces
 - Etc.



III. COMMENT ÇA FONCTIONNE ?

The image shows a screenshot of an IDE with two panels. The left panel is the 'OPEN EDITORS' sidebar, showing a project named 'MY-FIRST-APP'. Under this project, there are several files: 'app.component.html', 'app.component.spec.ts', 'app.component.ts', 'app.module.ts', 'assets', 'environments', 'favicon.ico', 'index.html', 'main.ts', 'polyfills.ts', and 'styles.css'. The file 'main.ts' is highlighted with a red box. The right panel shows the content of 'main.ts'. The code is as follows:

```
src > TS main.ts > ...
1  import { enableProdMode } from '@angular/core';
2  import { platformBrowserDynamic } from '@angular/platform-brows
3
4  import { AppModule } from './app/app.module';
5  import { environment } from './environments/environment';
6
7  if (environment.production) {
8    enableProdMode();
9  }
10
11  platformBrowserDynamic().bootstrapModule(AppModule)
12    .catch(err => console.error(err));
13
```

In the code, 'AppModule' is highlighted with a red box. A red line connects the 'main.ts' box in the file explorer to the 'AppModule' box in the code. Another red box highlights the 'bootstrapModule(AppModule)' part of the code, with a red line pointing to a text box on the right.

main.ts : c'est le fichier qui va être exécuté en premier

Va lancer (bootstrap) le module
« AppModule »

III. COMMENT ÇA FONCTIONNE ?

> OPEN EDITORS

✓ MY-FIRST-APP

> e2e

> node_modules

✓ src

✓ app

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

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
6 @NgModule({
7   declarations: [
8     AppComponent
9   ],
10  imports: [
11    BrowserModule
12  ],
13  providers: [],
14  bootstrap: [AppComponent]
15 })
16 export class AppModule { }
17
```

Va spécifier quels composants
(css, html, ts) du module vont
être utilisés

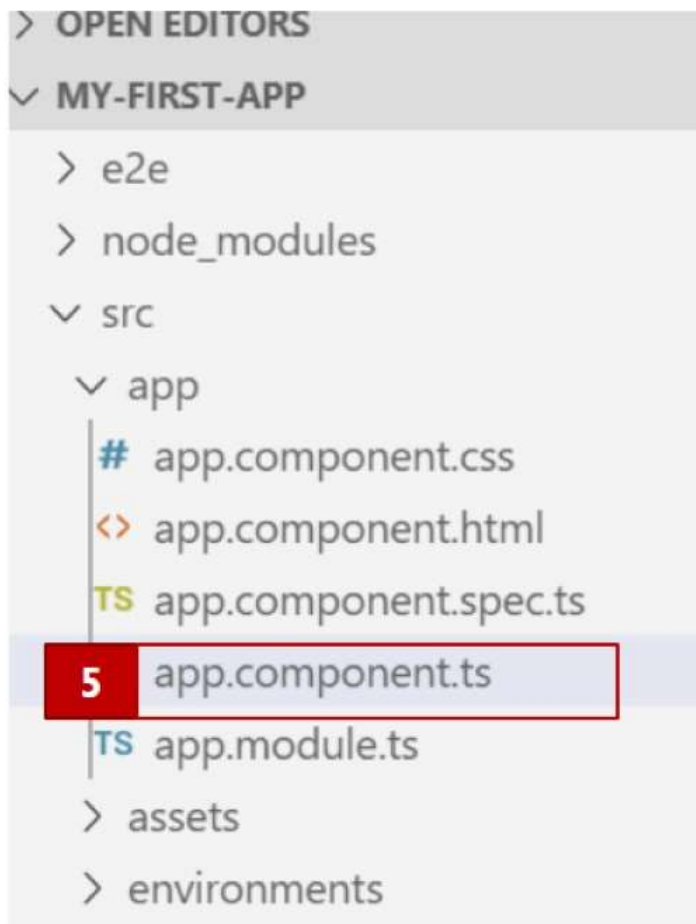
> OPEN EDITORS

✓ MY-FIRST-APP

- > e2e
- > node_modules
- ✓ src
 - ✓ app
 - 4 app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - 2 app.module.ts
 - > assets
 - > environments
 - ★ favicon.ico
 - <> index.html
 - 1 main.ts
 - TS polyfills.ts
 - # styles.css
 - TS test.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
6  @NgModule({
7    declarations: [
8      AppComponent
9    ],
10   imports: [
11     BrowserModule
12   ],
13   providers: [],
14   3 bootstrap: [AppComponent]
15 })
16 export class AppModule { }
17
```

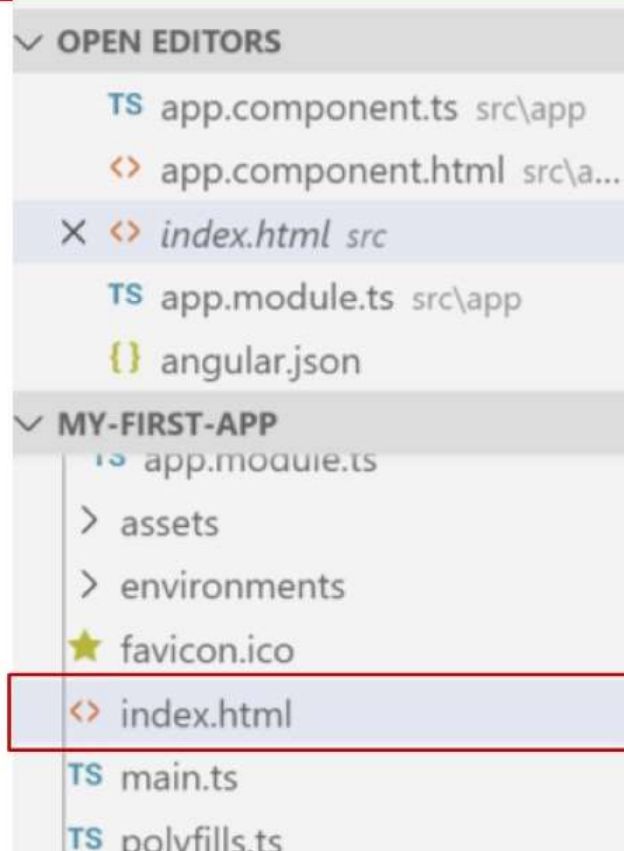
1. Angular démarre
2. On lance notre module de démarrage (app.module.ts)
3. On passe le module comme argument
4. C'est le contenu de « AppComponent »



```
src > app > TS app.component.ts > ...  
1  import { Component } from '@angular/core';  
2  
3  @Component({  
4      selector: 'app-root',  
5      templateUrl: './app.component.html',  
6      styleUrls: ['./app.component.css']  
7  })  
8  export class AppComponent {  
9      title = 'my-first-app';  
10  
11
```

5. Angular analyse le fichier « app.component.ts »
6. Ce fichier contient les informations nécessaires : selector (balise dans le fichier index.html, templateUrl (code en HTML), styleUrls (les feuilles de styles) et d'autres paramètres.

III. COMMENT ÇA FONCTIONNE ?



```
src > <> index.html > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>MyFirstApp</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-wi
8    <link rel="icon" type="image/x-icon" href="fav:
9  </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
14
```

« app-root » est le composant racine (par défaut).
Il va être remplacé par le app.component

III. COMMENT ÇA FONCTIONNE ?

> OPEN EDITORS

✓ MY-FIRST-APP

> e2e

> node_modules

✓ src

✓ app

app.component.css

<> app.component.html

TS app.component.spec.ts

TS app.component.ts

TS app.module.ts

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

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'my-first-app';
10 }
11
```

selector : contiendra le nom de
la balise dans le fichier
« index.html »



Welcome



http://localhost:4200/



star-phone-shop app is running!

Resources

Here are some links to help you get started:



[Learn Angular >](#)



[CLI Documentation >](#)



[Angular Blog >](#)

Next Steps

What do you want to do next with your app?

+ New Component

+ Angular Material

+ Add PWA Support

+ Add Dependency

+ Run and Watch Tests

+ Build for Production



```
ng generate component xyz
```



Love Angular? [Give our repo a star.](#)

