



Angular 2+

Développer des applications Web

Support de cours

Réf. T44A-046





Module 2

Installation et premier projet

Contenu du module

- Installation de NodeJS
- Mise à jours de NPM **N**ode **P**ackage **M**anager
- Installation du CLI **C**ommand **L**ine **I**nterface
- Définir un répertoire pour nos projets Angular
- Création du premier projet
- Ajouter Bootstrap au projet
- Ajouter Font Awesome au projet
- Lancer le projet
- Architecture du projet

Installation NodeJS

- Installer NodeJS **LTS**
- Eviter d'installer chocolatey
- <https://nodejs.org/>



Installation NodeJS

- Vérifier la version de NodeJS déjà installer dans le terminal

```
node --version
```



Installation de NPM Node Package Manager

- Installer ou mettre à jours NPM **N**ode **P**ackage **M**anager
- NPM est un "*package manager*« qui permet l'installation d'énormément d'outils et de librairies dont vous aurez besoin pour tout type de développement.
- NPM permet l'installation de différents modules pour nodeJS

Vérifier la version de npm

```
npm --version
```



Installer la dernière version de npm

```
npm install -g@latest
```

OU

```
npm i -g @latest
```

Installation du CLI Angular

- Installer le CLI Angular **C**ommand **L**ine **I**nterface
- Le CLI d'Angular nous permettra de créer nos projets Angular et de générer des fichiers ou des composants pour notre projet

Vérifier la version de Angular CLI

```
ng --version
```

Installer la dernière version de npm

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

Définir un répertoire pour nos projet Angular

- Définir un emplacement où sera stocké les projets Angular

- Revenir à la racine C:

```
cd\
```

- Créer un répertoire (md : **m**ake **d**irectory)

```
md prj-angular
```

- Aller dans ce répertoire

```
cd prj-angular
```



Aperçu de la console

```
C:\prj-angular>
```




Module 2

Mon premier projet

Création d'un projet Angular

- Depuis notre emplacement nous allons créer un projet Angular

Aperçu de la console

```
C:\prj-angular>
```

- Création d'un projet

```
ng new premier-prj --style=scss --skip-tests=true
```

nom du projet
du futur répertoire

Active les fichiers SCSS
Au lieu de CSS

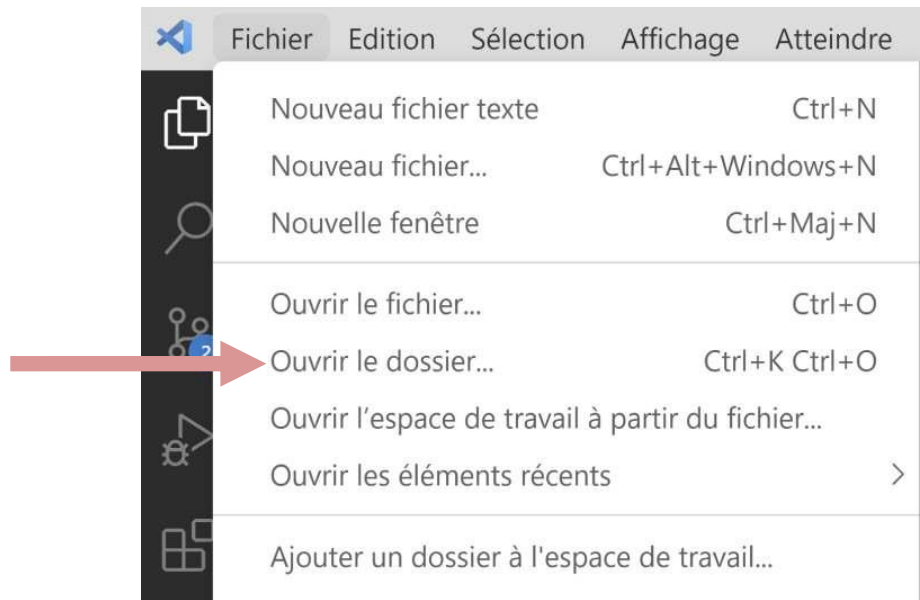
Enlève les fichiers de test

```
? Would you like to add Angular routing? Yes
```

Dites : y pour Yes

Ouvrir le projet Angular avec VS Code

- Sélectionner Fichier > Ouvrir le dossier ...



Ajouter Bootstrap au projet Angular

- Ouvrir le Terminal : Terminal > Nouveau terminal
 - *Command Prompt* (type de terminal)

Aperçu la console

```
C:\prj-angular\premier-prj>
```

```
npm i bootstrap
```

```
npm i bootstrap@3.3.7 --save
```

Ajouter Bootstrap au projet Angular

- Ouvrir le fichier **angular.json**
- Insérer cette ligne :

```
"node_modules/bootstrap/dist/css/bootstrap.css",
```

```
51      "assets": [  
52        "src/favicon.ico",  
53        "src/assets"  
54      ],  
55      "styles": [  
56        "src/styles.scss"  
57      ],  
58      "scripts": []
```



```
51      "assets": [  
52        "src/favicon.ico",  
53        "src/assets"  
54      ],  
55      "styles": [  
56        "node_modules/bootstrap/dist/css/bootstrap.css",  
57        "src/styles.scss"  
58      ],
```

Ajouter Font Awesome au projet Angular

- Ouvrir le Terminal : Terminal > Nouveau terminal
 - *Command Prompt* (type de terminal)
- Documentation :
 - <https://fontawesome.com/docs/web/setup/packages>

Aperçu la console

```
C:\prj-angular\premier-prj>
```

```
npm install --save @fontawesome/fontawesome-free
```

Ajouter Font Awesome au projet Angular

- Ouvrir le fichier **angular.json**
- Insérer cette ligne :

```
"node_modules/@fortawesome/fontawesome-free/css/all.min.css",
```

```
"assets": [  
  "src/favicon.ico",  
  "src/assets"  
],  
"styles": [  
  "node_modules/bootstrap/dist/css/bootstrap.css",  
  "node_modules/@fortawesome/fontawesome-free/css/all.min.css",  
  "src/styles.scss"  
],  
"scripts": []
```

Lancer le proger Angular

- Le projet est transcompilé
- Dès que l'on sauvegarde un fichier le code est transcompilé et la page du navigateur est rafraichie
- url de l'application **localhost:4200**

```
ng serve
```

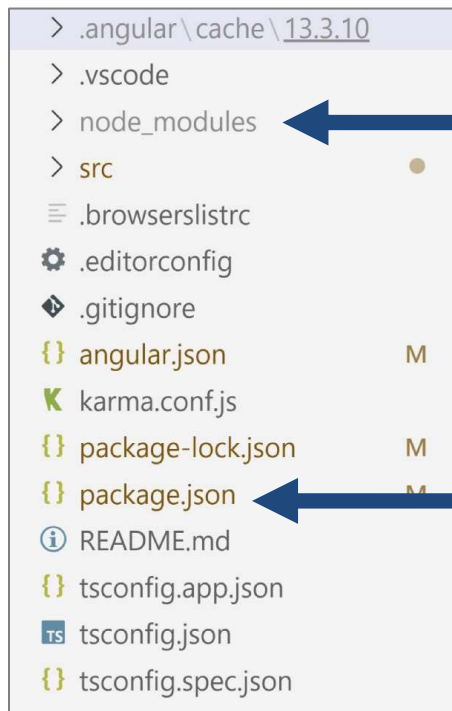
```
ng serve -o
```



Ouvre le navigateur web à l'adresse : localhost:4200

Architecture d'un projet Angular

- Fichiers et répertoires du projet Angular



Le répertoire **node_modules** peut être effacé

car le fichier **package.json** permet de le reconstruire facilement
Tout le volume du projet 321 Mo et 40 000 fichiers sont
contenus dans ce répertoire

Le fichier **package.json**

Contient tous les "node modules" installés
dans le répertoire **node_modules**

```
npm install
```

Pour reconstruire le répertoire
node_module

Architecture d'un projet Angular

- Fichiers et répertoire du projet Angular

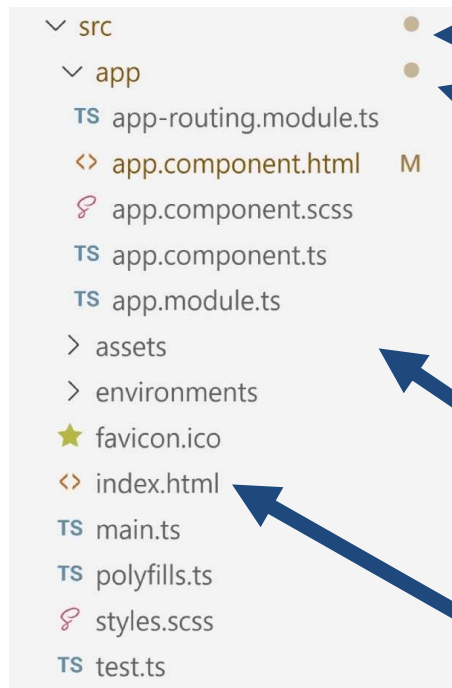


Le fichier **angular.json** permet de configurer le projet angular
C'est ici que l'on vient ajouter le path du css
pour **bootstrap** ou **Font Awesome**

Le 3 fichier **tsconfig** permettent de configurer TypeScript et sa compilation

Architecture d'un projet Angular

■ Fichiers et répertoire du projet Angular



Le répertoire **src** contient tout le code afin de créer ou modifier notre application

Le répertoire **app** contient

- le component "**root**" ou component principale (3 fichiers)
- Le fichier où sont définies les routes : **app-routing.module.ts**
- Le fichier où l'on définit les modules des composants : **app.module.ts**
Par exemple FormsModule pour utiliser les formulaires Angular

assets est le répertoire où sont stockés les images ou autres fichiers utiles à l'application, *par exemple : logo.jpg*

index.html est le point d'entrée de l'application
Il appelle le component principal (**root**)

Architecture d'un projet Angular

- Fichiers et répertoire du projet Angular

```
▼ src
  ▼ app
    TS app-routing.module.ts
    <> app.component.html
    ⚙ app.component.scss
    TS app.component.ts
    TS app.module.ts
  > assets
  > environments
  ★ favicon.ico
  <> index.html
  TS main.ts
  TS polyfills.ts
  ⚙ styles.scss
  TS test.ts
```

Les 3 fichiers du component principal : **root**

app.component.html

Fichier HTML : la vue

app.component.scss

Fichier CSS ou SCSS

app.component.ts

Fichier TS TypeScript le Model



un component