

Installation d'Angular

Angular est un puissant framework JavaScript permettant de créer des applications Web.

- 1- Si ce n'est pas déjà fait installez **Node.js**.
Pour vérifier s'il est déjà installé: ouvrez une **Invite de commande Windows (ICW)** et tapez la commande

```
node --version
```

Si la commande répond bien c'est que NodeJS est installé.

Sinon rendez-vous sur <https://nodejs.org/fr/> et procédez à son installation.

Au 1 juin 2024 la version LTS (Long Term Support) était 20.14.0

- 2- L'installation de Node.js nous donne accès au **npm** (Node.js Package Manager). Nous utiliserons npm pour installer le CLI (Command-Line Interface) d'Angular. Dans l'ICW, tapez:

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

Note : Vous frapperez possiblement une erreur git vous invitant à définir l'utilisateur et le courriel de votre compte git. Voici les commandes pour les définir :

```
git config --global user.name "Votre Nom"
```

```
git config --global user.email "votre.email@example.com"
```

- 3- Nous travaillerons avec la version 18 d'Angular. Pour vérifier l'installation tapez

```
ng v
```

```
C:\atelier\3D4\Aut2024>ng v

Angular CLI
-----
Angular CLI: 18.0.3
Node: 20.14.0
Package Manager: npm 10.7.0
OS: win32 x64

Angular:
...

Package      Version
-----
@angular-devkit/architect 0.1800.3 (cli-only)
@angular-devkit/core      18.0.3 (cli-only)
@angular-devkit/schematics 18.0.3 (cli-only)
@schematics/angular       18.0.3 (cli-only)

C:\atelier\3D4\Aut2024>
```

- 4- Positionnez-vous dans le dossier où vous voulez mettre vos projets Angular. Par exemple c:\atelier\3D4\Aut2024
- 5- Tapez

```
ng new nomProjet
```

Note : Remplacez nomProjet par le vrai nom de votre projet. Par exemple, ng new

jourvie24

- 6- Faut choisir le format de stylesheet désiré, prenez celui par défaut (CSS simple) en tapant « Enter »

```
C:\atelier\3D4\Aut2024>ng new jourvie24
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS [ https://developer.mozilla.org/docs/Web/CSS ]
  Sass (SCSS) [ https://sass-lang.com/documentation/syntax#scss ]
  Sass (Indented) [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]
```

- 7- L'environnement de développement de votre projet s'installe, ça peut prendre plusieurs secondes...

- 8- Tapez

cd jourvie24

Vous êtes maintenant sur c:\atelier\3D4\Aut24\jourvie24

- 9- Tapez

ng serve

Cette commande buildera votre app, lancera un serveur web de développement pour cette app

```
C:\atelier\3D4\Aut2024\jourvie24>ng serve
Initial chunk files | Names | Raw size
polyfills.js | polyfills | 88.34 kB |
main.js | main | 22.63 kB |
styles.css | styles | 95 bytes |

| Initial total | 111.06 kB

Application bundle generation complete. [4.044 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
→ Local: http://localhost:4200/
→ press h + enter to show help
```

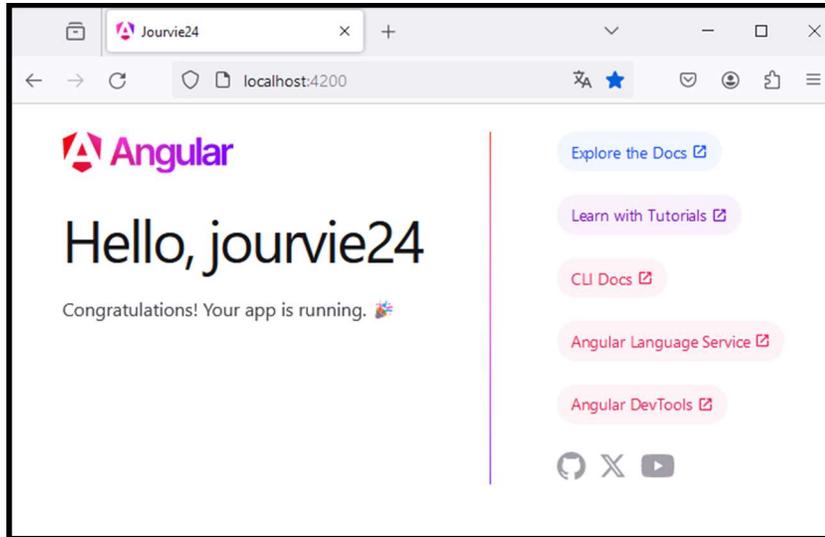
Le serveur lancé détectera tout changement à vos fichiers sources. De tels changements entraîneront un « rebuild » de votre app.

- 10- Par défaut le serveur web de développement d'Angular occupera le port 4200.

- 11- Ouvrez votre fureteur Web favori (Chrome, Firefox, Edge, etc) et saisissez l'URL

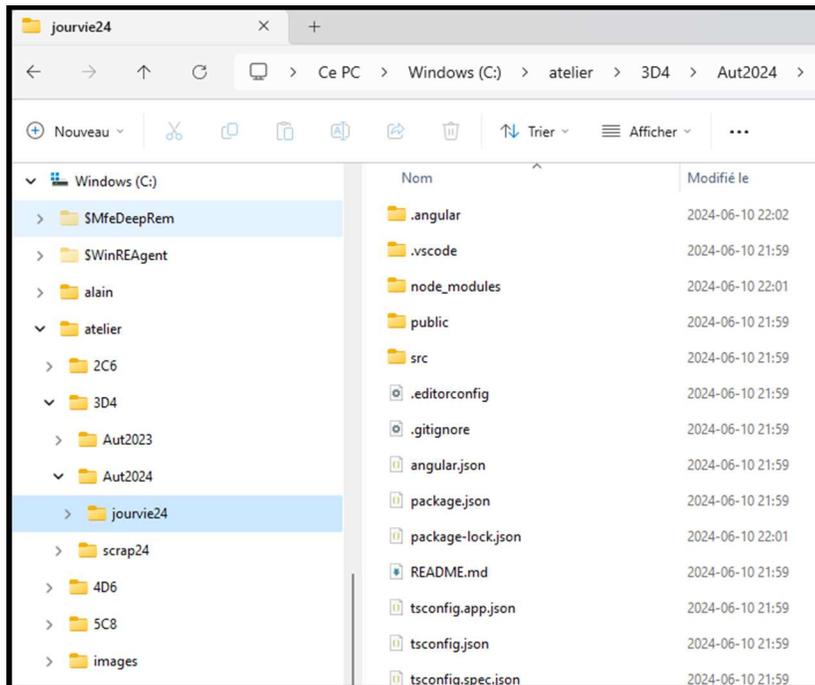
<http://localhost:4200>

vous devriez voir ceci :



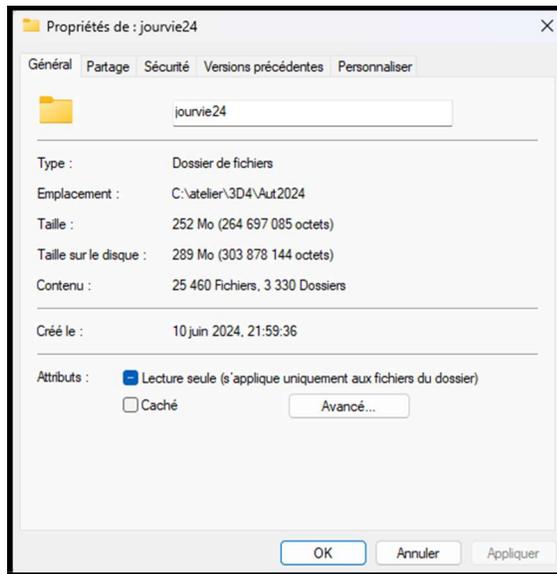
Ça correspond à l'état initial d'un projet Angular.

12- Ouvrons un explorateur Windows et analysons la structure du dossier racine du projet:

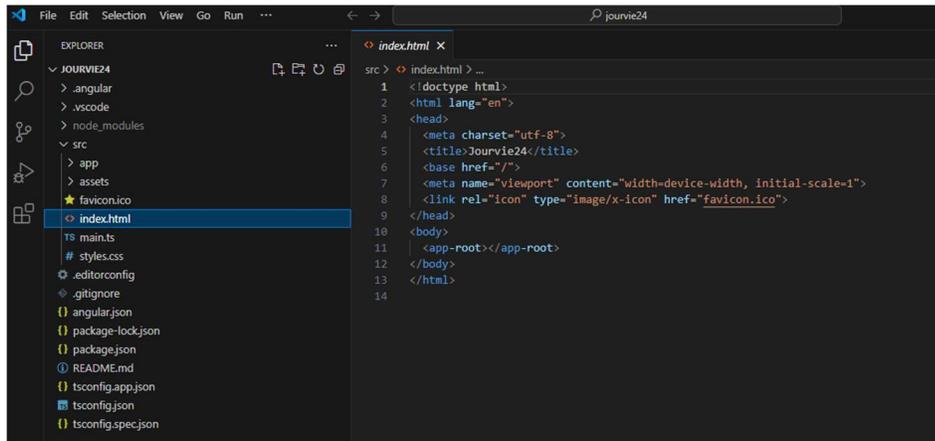


- La racine du projet est `c:\atelier\3D4\Aut2024\jourvie24\`

- Un clic droit sur le dossier racine nous donne les propriétés du projet :



- Plus de 25000 fichiers sur 289 Mo : un projet Angular c'est gros
- Dans cette racine, le dossier que nous utiliserons le plus est **src**
- Ouvrons le dossier du projet dans l'éditeur VisualCode, puis sélectionnons le fichier d'entrée de notre projet : **... \jourvie24\src\index.html**



- La balise **<body>** contient la balise **<app-root>** et c'est tout. **<app-root>** n'est pas une balise HTML standard, c'est une balise Angular qui sert de point d'entrée à notre application.
- On injectera dans cette balise le composant de base du projet. Le code du composant de base se trouve dans deux fichiers :
... \jourvie24\src\app\app.component.ts et
... \jourvie24\src\app\app.component.html

- Analysons le code TypeScript du component de base :
...\\jourvie24\\src\\app\\app.component.ts:

```

src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3  |
4  @Component({
5    selector: 'app-root',
6    standalone: true,
7    imports: [RouterOutlet],
8    templateUrl: './app.component.html',
9    styleUrls: ['./app.component.css']
10 })
11 export class AppComponent {
12   title = 'jourvie24';
13 }
14

```

- Les deux premières lignes :

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
```

indique les items nécessaires au component et où les trouver

- Suit le décorateur (@) du component :

```

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

```

Il comporte plusieurs attributs :

- selector : nom de la balise html qui accueillera le component sur la page web
- standalone : Si true, évite à l'app d'avoir à déclarer un module d'importation (nouvelle option depuis V18)
- imports : les composants externes nécessaires à l'app
- templateUrl : le fichier html du component
- styleUrls : fichier de styles du component (facultatif)
- Enfin, le code de la classe du component, qui se contente pour l'instant d'initialiser le title de l'app :

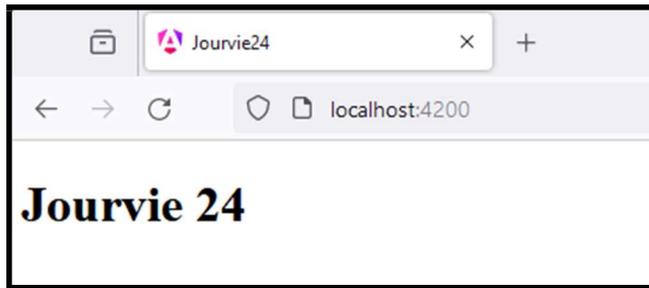
```

export class AppComponent {
  title = 'jourvie24';
}

```

- 13-** Ouvrez le fichier ...\\jourvie24\\src\\app\\app.component.html. Sélectionnez tout son contenu et effacez-le (n'ayez pas peur : ctrl-a + supprimer) et remplacez-le par :<h1>Jourvie24</h1>

- 14- Le serveur Web de développement réagit au changement du code et relance un nouveau build puis le fureteur se met à jour automatiquement:



- 15- On veut intégrer Bootstrap (framework CSS) à notre projet. Sur l'ICW arrêtez le serveur (**ctrl-c**) et tapez :

npm i bootstrap

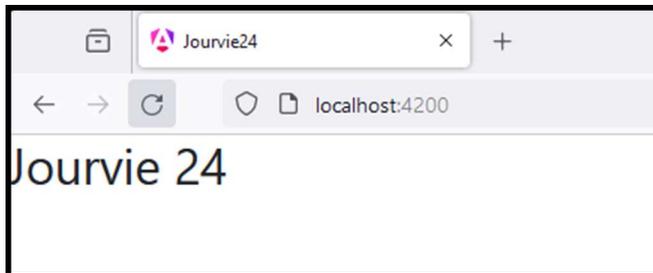
puis modifiez `jourvie24\angular.json` ainsi :

```
...  
"styles": [  
  "src/styles.css",  
  "node_modules/bootstrap/dist/css/bootstrap.min.css"  
],  
...
```

- 16- Relancer le serveur de développement Angular :

ng serve

rafraichissez l'URL **localhost:4200** dans votre fureteur :



On voit la belle police de **Bootstrap!**

- 17- Dans le fichier

`...\jourvie24\src\style.css` :

ajoutez ce contenu :

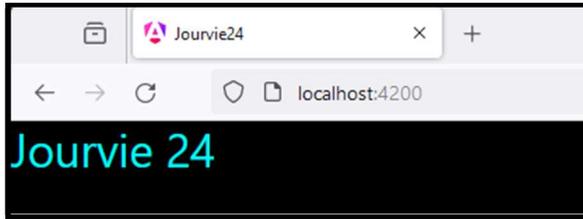
```
#fond{  
background-color: black;  
color: aqua;  
}
```

- 18- Éditez

`...\jourvie24\src\index.html`, modifiez la balise `body` :

```
<body id='fond'>
```

19- Votre fureteur se rafraichira :



20- Ça y est! Vous avez créé et modifié votre premier projet Angular! Le fun commence...

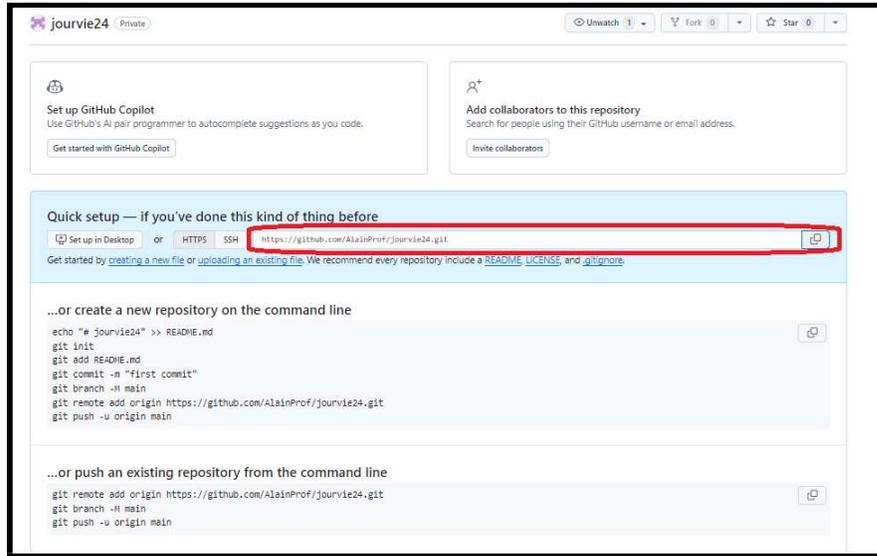
Git et Angular

1- Lors de la création d'un projet Angular (ng new ...) un dossier .git est créé à la racine du projet contenant un fichier **.gitignore** spécifiquement conçu pour Angular. Nous utiliserons git et github pour gérer les sources du projet

- Positionnez-vous à la racine du projet (...\\jourvie24\\)
- git status
- **git add .**
- **git commit -m"initial"**
- Allez sur github.com, connectez-vous et créez un repository nommé **jourvie24** (private, pas de readme, pas de gitignore)

A screenshot of the GitHub 'Create a new repository' form. The form is titled 'Create a new repository' and includes a link to 'Import a repository'. It has a section for 'Required fields are marked with an asterisk (*)' with 'Owner *' set to 'AlainProf' and 'Repository name *' set to 'jourvie24'. A green checkmark indicates 'jourvie24 is available'. There is a 'Description (optional)' field with the text 'Application jourvie pour l'aut 2024'. The 'Public' radio button is unselected, and the 'Private' radio button is selected. Under 'Initialize this repository with:', the 'Add a README file' checkbox is unselected. The 'Add .gitignore' section has a dropdown menu set to '.gitignore template: None'. The 'Choose a license' section has a dropdown menu set to 'License: None'. At the bottom right, there is a green 'Create repository' button.

- Copiez l'URL de ce repository :



- Retour sur l'ICW à la racine de votre projet (c:\atelier\3D4\Aut2024\jourvie24)
 - Configurez le serveur avec la commande **git remote**. Vous lui passez l'URL en paramètre :
git remote add origin https://github.com/alainprof/jourvie24.git
 - **Envoyez votre projet sur le serveur de github.com :**
git push origin master
- 2- Le projet est maintenant disponible sur GitHub. Pour l'installer sur un nouveau poste:
- Positionnez-vous sur le dossier où vous désirez installer le projet
 - **git clone https://github.com/alainprof/jourvie24.git**
 - **cd jourvie24**
 - **npm i**
 - À ce stade vous serez synchro avec github et vous pourrez lancer le serveur (**ng serve**) puis apporter des modifications à votre projet et conserver ces modifs dans github avec une série de git add, git commit, git push, etc