

Atelier 01 Angular :

Création d'un premier projet : Composant Web, Routing, Bootstrap

Step 3 - Install the latest version of Angular

With node . js and npm installed, the next step is to install the [Angular CLI](#) which provides tooling for effective Angular development.

From a Terminal window run the following command: `npm install -g @angular/cli`.

Step 4 - Install integrated development environment (IDE)

You are free to use any tool you prefer to build apps with Angular. We recommend the following:

1. [Visual Studio Code](#)
2. As an optional, but recommended step you can further improve your developer experience by installing the [Angular Language Service](#)

Objectifs :

1. Création d'un premier projet Angular,
2. Installation de Bootstrap,
3. Création d'un menu Navbar,
4. Création du Web Component Produits,
5. Création du Web Component add-produit,
6. Ajout d'un composant web par défaut,
7. Utiliser le « Data Binding » pour afficher la liste des produits.

Création d'un premier projet Angular

1. Créer un projet Angular nommé MesProduits :

ng new --no-standalone MesProduits

2. Editer le projet avec VS code, créer un terminal et lancer le serveur intégré :

ng serve

3. Tester votre travail [http://localhost :4200](http://localhost:4200)

Installation de Bootstrap

4. créer un terminal puis lancer : **npm install bootstrap@4.5.3**
5. éditer le fichier **angular.json** :

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

Création d'un menu Navbar

6. Editer le fichier app.component.html :

```
<!DOCTYPE html>
<html >
<head>
<meta charset="utf-8">
<title>Gestion des Produits</title>
</head>
<body>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <!-- Brand -->
  <a class="navbar-brand" href="#">Gestion Produits</a>

  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#" >Home</a>
    </li>

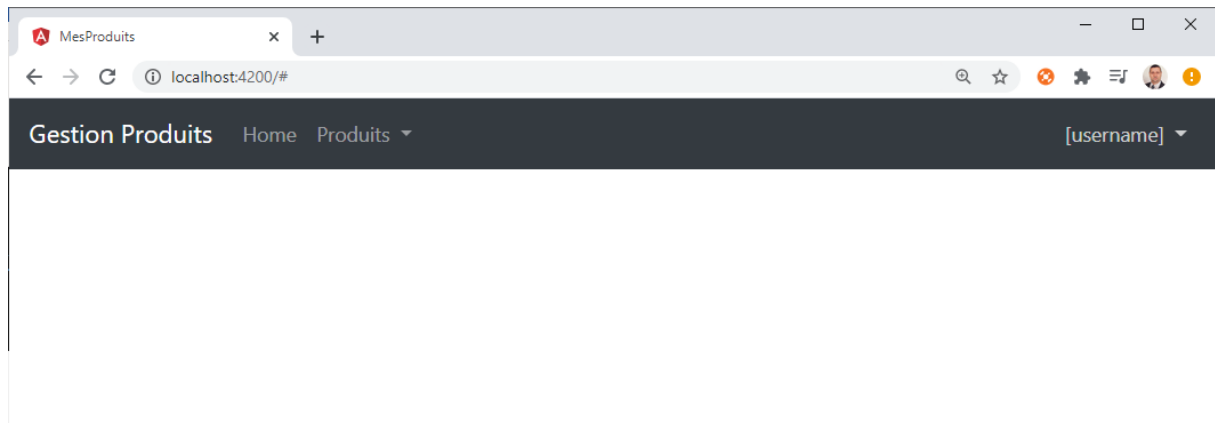
    <!-- Dropdown -->
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
        Produits
      </a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">Ajouter</a>
        <a class="dropdown-item" href="#">Lister</a>
      </div>
    </li>
  </ul>

  <ul class="navbar-nav ml-auto" >
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
        [username]
      </a>
      <div class="dropdown-menu">
        <a class="dropdown-item" href="#">login</a>
        <a class="dropdown-item" href="#">logout</a>
        <a class="dropdown-item" href="#">Profile</a>
      </div>
    </li>
  </ul>
</nav>
</body>
</html>
```

```

    </li>
  </ul>
</nav>
</body>
</html>
<router-outlet></router-outlet>

```



Vous pouvez remarquer que les dropdown listes ne fonctionnent pas car elles ont besoins de JQuery.js et de bootstrap.js

Installation de JQuery

7. Lancer la commande : **npm install jquery**
8. Ajouter jquery.js et bootstrap.min.js au fichier **angular.json** :

```

"scripts": [ "node_modules/jquery/dist/jquery.js",
              "node_modules/bootstrap/dist/js/bootstrap.min.js"
            ]

```

9. Redémarrer le serveur : CTRL+C puis **ng serve**
10. Tester les les dropdown listes.

Création du Web Component Produits et de son Routing

11. Créer le composant Web produits en tapant :

ng g c produits

12. Editer le fichier **app-routing.module.ts** et y ajouter la Route « produits » :

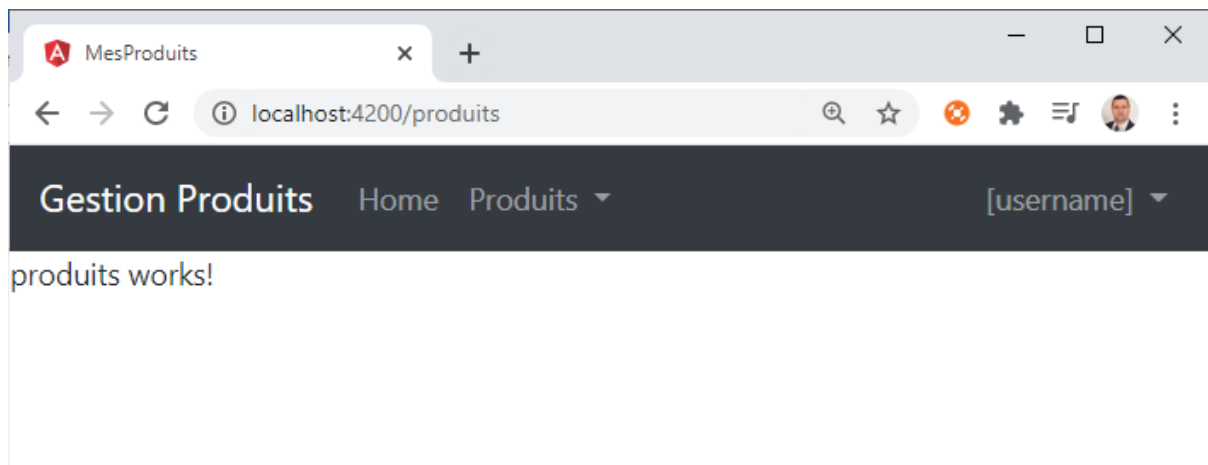
```
import { ProduitsComponent } from './produits/produits.component';

const routes: Routes = [
  {path: "produits", component : ProduitsComponent}
];
```

13. Editer le fichier app.component.html pour ajouter le lien vers le composant produits :

```
<li class="nav-item">
  <a class="nav-link" routerLink="/produits" >Home</a>
</li>
```

14. Tester le lien en cliquant sur « Home »



Création du composant Web add-produit

15. Créer le Web component add-produit permettant d'ajouter un nouveau produit :

ng g c add-produit

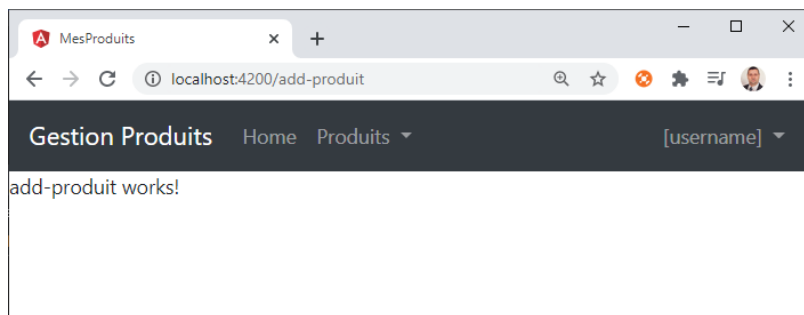
16. Editer le fichier **app-routing.module.ts** et y ajouter la Route « add-produit » :

```
import { AddProduitComponent } from './add-produit/add-produit.component';
```

```
const routes: Routes = [
  {path: "produits", component : ProduitsComponent},
  {path: "add-produit", component : AddProduitComponent}
];
```

17. Editer le fichier app.component.html pour ajouter le lien vers le composant add-produits :

```
<div class="dropdown-menu">
  <a class="dropdown-item" routerLink="/add-produit">Ajouter</a>
  <a class="dropdown-item" href="#">Lister</a>
</div>
```



Ajout d'un composant web par défaut

18. Ajouter un composant par défaut (ici : /produits) :

```
const routes: Routes = [
  {path: "produits", component : ProduitsComponent},
  {path: "add-produit", component : AddProduitComponent},
  { path: "", redirectTo: "produits", pathMatch: "full" }
];
```

19. Tapez dans l'url : <http://localhost:4200/> vous allez être redirigé vers <http://localhost:4200/produits>

Utiliser le « Data Binding » pour afficher la liste des produits

20. Au niveau de la classe *ProduitsComponent* créer l'attribut **produits** qui est un tableau de string

```
export class ProduitsComponent implements OnInit {  
  produits : string[]; //un tableau de chaînes de caractères
```

21. Au niveau du constructeur de la même classe affecter des valeurs de produits au tableau **produits** :

```
constructor() {  
  this.produits = ["PC Asus", "Imprimante Epson", "Tablette Samsung"];  
}
```

22. Afficher la liste des produits en modifiant le fichier *produits.component.html* comme suit :

```
<div class="container">  
  <div class="card shadow mb-4">  
    <div class="card-body">  
      <table >  
        <thead>  
          <tr>  
            <th>Liste de Produits</th>  
          </tr>  
        </thead>  
        <tbody *ngFor="let produit of produits">  
          <tr >  
            <td>{{produit}}</td>  
          </tr>  
        </tbody>  
      </table>  
    </div>  
  </div>  
</div>
```

