INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL

DEPARTEMENT TECHNOLOGIE DE L'INFORMATIQUE

TP4 : ROUTING

Matière : Atelier Framework Coté Client **Enseignante : Sonia Guerbouj**

Classes : DSI2, MDW2 **Durée : 3h**

Objectif

Le but de ce TP est de comprendre la mise en place du mécanisme de routing dans une application Angular pour assurer la navigation entre les vues.

1- RoutingModule

Pour pouvoir naviguer entre les vues des différents composants d'une application avec Angular, il faut intégrer le module RoutingModule lors de la création du projet. Soit en l'acceptant lors d'une des étapes de création du projet, soit en le forçant dès le début grâce à l'option ajoutée à la commande : `ng new nom_projet --routing`. Dans les 2 cas, RoutingModule est importé dans le fichier « app.component.ts ».
Ensuite, il faut ajouter toutes les routes de l'application (les URLs) au niveau du fichier « app-routing.module.ts » dans la constante « routes » en indiquant le chemin (path) et le composant correspondant.
Enfin, il faut s'assurer de l'existence de la balise <router-outlet> à la fin du fichier « app.component.html ».

2- Travail à faire

Ouvrez l'éditeur de code (VS Code) et créez les projets Angular sous le dossier « DSI21 ».

Exercice 1

Dans cet exercice, vous allez créer un nouveau projet « **magasin** », dans lequel sera affiché la liste d'articles à vendre. Pour rendre cette application attrayante, vous commencerez par installer Bootstrap et jQuery afin de créer la barre de menu.

1) Créez le projet **magasin** : `ng new magasin – routing --skipTests=true`
2) Ouvrez le projet avec VSCode, puis ouvrez un terminal et installez Bootrap et jQuery en exécutant ces commandes dans le projet :

```
npm install bootstrap@4.5.3
npm install jquery
```

3) Dans le fichier « **angular.json** », cherchez et éditez le paramètre "**styles**" ainsi :

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

4) Dans le même fichier « **angular.json** », éditez le paramètre "**scripts**" pour ajouter jquery.js et bootstrap.min.js comme suit :

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

5) Remplacez le contenu du fichier « **app.component.html** », par le code suivant :

```
<html >
 <head>
  <meta charset="utf-8">
  <title>Mon magasin</title>
 </head>
<body>
 <nav class="navbar navbar-expand-sm bg-info navbar-light">
  <a class="navbar-brand" href="">Mon magasin</a>
  <ul class="navbar-nav">
   <li class="nav-item">
    <a class="nav-link" href="" >Accueil</a>
   </li>
   <li class="nav-item dropdown">
    <a  class="nav-link  dropdown-toggle"  href=""  id="navbardrop"
data-toggle="dropdown">Articles</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">[Utilisateur]</a>
    <div class="dropdown-menu">
     <a class="dropdown-item" href="">Connexion</a>
     <a class="dropdown-item" href="">Déconnexion</a>
     <a class="dropdown-item" href="">Profile</a>
    </div>
   </li>
  </ul>
 </nav>
</body>
</html>
<router-outlet></router-outlet>
```

6) Lancez le projet avec la commande `ng serve --open`. Testez le menu.

7) Créez les 2 composants « **articles** » et « **add-article** » en tapant :

`ng g c articles --skipTests=true` puis : `ng g c add-article --skipTests=true`

8) Dans « **app-routing.module.ts** », importez les 2 composants et insérerez leurs routes ainsi qu'une redirection par défaut vers le composant « articles » :

```
import { ArticlesComponent } from './articles/articles.component';
import { AddArticleComponent } from './add-article/add-article.component';

const routes: Routes = [
  {path: "articles", component : ArticlesComponent},
  {path: "add-article", component : AddArticleComponent},
  {path: "", redirectTo: "articles", pathMatch: "full"}
];
```

9) Modifiez « **app.component.html** » pour indiquer les liens vers les 2 composants :

```
...
<div class="dropdown-menu">
 <a class="dropdown-item"  href="/add-article">Ajouter</a>
 <a class="dropdown-item"  href="/articles">Lister</a>
</div>
...
```

10) Assurez-vous que tous les liens du menu à gauche marchent bien (Accueil, Ajouter, Lister). Testez aussi l'URL « 127.0.0.1:4200/magasin/ », qu'affiche-t-elle ?

Exercice 2

Dans le projet **magasin** de l'exercice 1, on utilisera une classe modèle pour afficher la liste des articles vendus.

1) Sous le dossier « **app** » du projet, ajoutez un dossier « **model** » dans lequel créez un fichier « **article.model.ts** ». Dans ce dernier, exportez et décrire une classe Article caractérisée par un code (codea), un libellé (libelle), un prix, une quantité (qte) et une date d'ajout (dateAjout).

NB :

- Pensez à bien choisir les types des propriétés (attributs).
- Ajoutez « ? » à la fin de leurs noms pour éviter de les initialiser.

2) Dans le fichier « **articles.components.ts** » importez la classe Article et son fichier. Puis, déclarez « **tab_art** » un tableau d'objets Article, à initialiser par 3 exemples d'articles dans le constructeur :

```
import { Article } from '../model/article.model';

@Component({
```

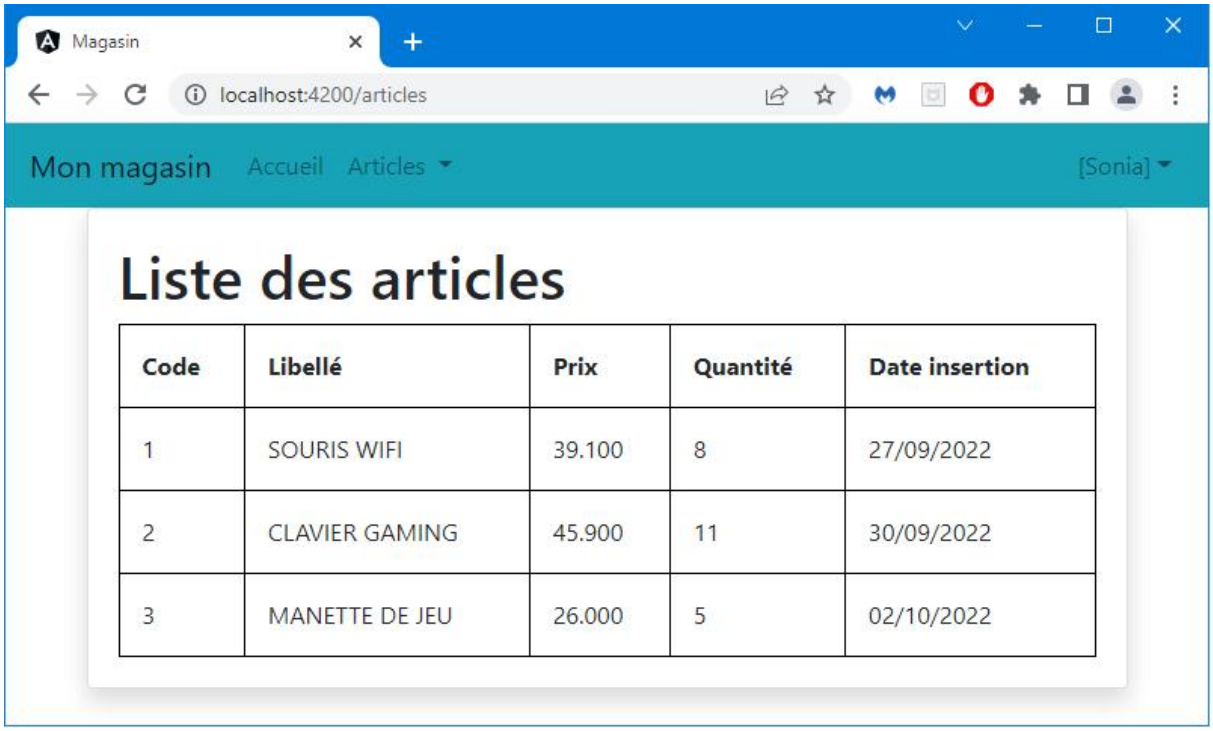
```
  selector: 'app-articles',
  templateUrl: './articles.component.html',
  styleUrls: ['./articles.component.css']
})

export class ArticlesComponent implements OnInit {

  tab_art : Article[];

  constructor() {
    this.tab_art = [
     {codea : 1,  libelle : "Souris Wifi", prix : 39.100, qte : 8,
dateAjout : new Date("09/27/2022")},
      {codea : 2,  libelle : "Clavier Gaming", prix : 45.900, qte :
11, dateAjout : new Date("09/30/2022")},
      {codea : 3,  libelle : "Manette de jeu", prix : 26, qte : 5,
dateAjout : new Date("10/02/2022")}
    ];
  }

  ngOnInit(): void {
  }

}
```

3) Remplacez le contenu du fichier « **articles.component.html** » par ce qui suit :

```
<div class="container">
    <div class="card shadow mb-4">
        <div class="card-body">
            <h1>Liste des articles</h1>
            <table width="100%">
                <thead>
                    <tr>
                        <th>Code</th>
                        <th>Libellé</th>
                        <th>Prix</th>
                        <th>Quantité</th>
                        <th>Date insertion</th>
                    </tr>
                </thead>
                <tbody *ngFor="let a of tab_art">
                    <tr >
                        <td>{{a.codea}}</td>
                        <td>{{a.libelle | uppercase}}</td>
                        <td>{{a.prix}}</td>
                        <td>{{a.qte}}</td>
                        <td>{{a.dateAjout }}</td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>
```

4) On remarque que la date s'affiche en anglais. Pour modifier son format, on ajoute un filtre : `{{a.dateAjout | date:'dd/MM/yyyy'}}`
5) Pour que les prix s'affichent toujours avec 3 chiffres après la virgule, on ajoute un filtre : `{{a.prix | number : '1.3'}}`
6) Pour améliorer l'affichage du tableau, ajoutez les styles suivants dans « **articles.component.css** » :

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 15px;
}
```



Exercice 3 : Travail à la maison (noté)

A la maison, créez un nouveau projet en reprenant les 2 exercices (avec des noms de composants différents), pour réaliser une application de gestion de votre choix. L'enseignante validera votre travail à la prochaine séance.

NB : Chaque étudiant doit fournir une application de gestion différente de ses camarades, sinon, une note de zéro lui sera attribuée ainsi qu'à son camarade.