Atelier Angular :

La recherche dans Angular

Objectifs :

1. Rechercher les produits par catégorie,
2. Rechercher par nom des produits en utilisant une api,
3. Rechercher par nom des produits en utilisant l'event keyup,
4. Rechercher par nom des produits en utilisant un pipe.

Rechercher les produits par catégorie

1. Créer le Web component rechercheParCategorie (sans créer le fichier .spec ni le fichier .css):

   ng g c rechercheParCategorie --skip-tests -s

2. Editer le fichier **app.routes.ts** et y ajouter la route « rechercheParCategorie» :

```
{path: "rechercheParCategorie", component : RechercheParCategorieComponent}
```

3. Editer le fichier app.component.html pour ajouter le lien vers le composant rechercheParCategorie:

```
    <li> <a class="dropdown-item"  routerLink="/rechercheParCategorie" >Recherche Par Catégorie</a></li>
```

4. Tester le lien en cliquant sur « Recherche Par Catégorie »

5. Ajouter à la classe ProduitService la méthode rechercheParCategorie() :

```
rechercherParCategorie(idCat: number):Observable< Produit[]> {

          const url = `${this.apiURL}/prodscat/${idCat}`;
          return this.http.get<Produit[]>(url);
            }
```

6. Copier dans recherche-par-categorie.component.html le contenu du fichier produits.component.html et commentez le code des boutons « Modifier » et « Supprimer » :

```
<div class="container">
    <div class="card shadow mb-4">
      <div class="card-body">
       <table class="table table-striped">
          <tr> <th>Id</th><th>Nom</th> <th>Prix</th>  <th>Date Création</th>
<th>Catégorie</th></tr>
          @for (produit of produits; track produit.idProduit)
          {
           <tbody >
             <tr >
                 <td>{{produit.idProduit}}</td>
                <td>{{produit.nomProduit}}</td>
                <td>{{produit.prixProduit}}</td>
                <td>{{produit.dateCreation | date:'dd/MM/yyyy'}}</td>

                @if(produit.categorie){
                 <td>{{produit.categorie.nomCat}}</td>  }
                @else {
                 <td>PAS DE CATEGORIE</td>
                 }
  <!--
                <td><a class="btn btn-danger"
(click)="supprimerProduit(produit)">Supprimer</a></td>
                <td><a class="btn btn-success"
[routerLink]="['/updateProduit',produit.idProduit]">Modifier</a></td> -->
              </tr>
            </tbody>
          }
      </table>
      </div>
    </div>
  </div>
```

7. Ajouter, dans la div Container, la liste déroulant contenant les catégories au fichier rechercheParCategorie.component.html :

```
<div class="container">
    <div class="col-sm-4 col-md-4 col-lg-4 mb-4">
        <label>Categories </label>
        <select class="form-control form-control-lg" id="idCat" name="idCat"
[(ngModel)]="IdCategorie"    (change)="onChange()">
            @for(cat of categories;track cat.idCat)
            {
                <option value="{{cat.idCat}}"> {{cat.nomCat}}  </option>
            }
        </select>
    </div>
……..
```

8. Modifier la méthode ngOnInit() de la classe RechercheParCategorieComponent :

```
  produits! : Produit[];
  IdCategorie! : number;
  categories! : Categorie[];
```

```
ngOnInit(): void {
    this.produitService.listeCategories().
    subscribe(cats => {this.categories = cats._embedded.categories;
      console.log(cats);
  });
  }
```

9. Ajouter la méthode onChange() à la classe RechercheParCategorieComponent :

```
onChange() {
    this.produitService.rechercherParCategorie(this.IdCategorie).
      subscribe(prods =>{this.produits=prods});
    }
```

Au niveau du fichier recherche-par-categorie.component.ts, importer les modules : FormsModule et CommonModule :

```
  imports: [FormsModule, CommonModule],
```

Rechercher par nom des produits en utilisant une api

10. Au niveau du projet Spring boot, ajouter une api que recherche les produits dont le nom contient un texte donné :

```
@RequestMapping(value="/prodsByName/{nom}",method = RequestMethod.GET)
    public List<Produit> findByNomProduitContains(@PathVariable("nom") String nom) {
            return produitService.findByNomProduitContains(nom);
    }
```

11. Créer le composant Web rechercheParNom en tapant :

    ng g c rechercheParNom --skip-tests -s

12. Editer le fichier **app.routes.ts** et y ajouter la Route « rechercheParNom» :

```
{path: "rechercheParNom", component : RechercheParNomComponent},
```

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

```
<li><a class="dropdown-item"  routerLink="/rechercheParNom" >Recherche Par Nom</a></li>
```

14. Copier dans recherche-par-nom.component.html le contenu du fichier recherche-par-categorie.component.html et remplacer la liste Select des catégories par un input :

```
<div class="container">
    <form class="form-inline">
        <div class="form-group mx-sm-3 mb-2 mt-2 ">
            <label>Nom Produit</label>
            <input type="text" class="form-control" [(ngModel)]="nomProduit" name="nomProduit"
class="form-control">
<button type="submit" (click)="rechercherProds()" class="btn btn-primary mb-2 ml-2">Rechercher</button>
        </div>
    </form>
    <div class="card shadow mb-4">
        <div class="card-body">
            <table class="table table-striped">
                <tr>
                    <th>Id</th>
                    <th>Nom</th>
                    <th>Prix</th>
                    <th>Date Création</th>
                    <th>Catégorie</th>
                </tr>
                @for (produit of produits; track produit.idProduit)
                {
```

```
                <tbody>
                    <tr>
                        <td>{{produit.idProduit}}</td>
                        <td>{{produit.nomProduit}}</td>
                        <td>{{produit.prixProduit}}</td>
                        <td>{{produit.dateCreation | date:'dd/MM/yyyy'}}</td>

                        @if(produit.categorie){
                        <td>{{produit.categorie.nomCat}}</td> }
                        @else {
                        <td>PAS DE CATEGORIE</td>
                        }
                        <!--
                     <td><a class="btn btn-danger"
(click)="supprimerProduit(produit)">Supprimer</a></td>
                     <td><a class="btn btn-success"
[routerLink]="['/updateProduit',produit.idProduit]">Modifier</a></td> -->
                    </tr>
                </tbody>
                }
            </table>
        </div>
    </div>
</div>
```

15. Ajouter la méthode rechercherParNom() à la classe ProduitService :

```
rechercherParNom(nom: string):Observable< Produit[]> {
           const url = `${this.apiURL}/prodsByName/${nom}`;
             return this.http.get<Produit[]>(url);
          }
```

16. Ajouter la méthode rechercherProds () à la classe RechercheParNomComponent :

```
import { CommonModule } from '@angular/common';
import { Component, OnInit } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { Produit } from '../model/produit.model';
import { ProduitService } from '../services/produit.service';

@Component({
  selector: 'app-recherche-par-nom',
  standalone: true,
  imports: [FormsModule, CommonModule],
  templateUrl: './recherche-par-nom.component.html',
  styles: ``
})
```

```
export class RechercheParNomComponent implements OnInit {

  nomProduit! : string;
  produits!: Produit[];

  constructor(private produitService : ProduitService) { }

  ngOnInit(): void {
    this.produitService.listeProduit().subscribe(prods => {
      console.log(prods);
      this.produits = prods;
      });
  }

  rechercherProds(){
    this.produitService.rechercherParNom(this.nomProduit).
    subscribe(prods => {
      console.log(prods);
      this.produits=prods;});
  }


}
```

```
rechercherProds(){
   this.produitService.rechercherParNom(this.nomProduit).
       subscribe(prods => {
         this.produits = prods;
         console.log(prods)});
 }
```

17. Tester votre travail

Amélioration

Si la chaine de recherche entrée est vide on affiche tous les produits :

```
rechercherProds()
  {
    if (this.nomProduit)
      //ou bien (this.nomProduit!=="")
      this.produitService
        .rechercherParNom(this.nomProduit)
        .subscribe((prods) => {
          console.log(prods);
          this.produits = prods;
```

```
      });
  else
    this.produitService.listeProduit().subscribe((prods) => {
      console.log(prods);
      this.produits = prods;
    });
}
```

Rechercher par nom des produits en utilisant l'event keyup

18. Modifier le fichier recherche-par-nom.component.html comme suit :

```
<form class="form-inline">
       <div class="form-group mx-sm-3 mb-2 mt-2 ">
           <label>Nom Produit en tapant</label>
 <input class="form-control"  #searchTerm (keyup)="onKeyUp(searchTerm.value)">
       </div>
   </form>
```

19. Ajouter la méthode onKeyup à la classe RechercheParNomComponent :

```
allProduits! : Produit[];
searchTerm!: string;


ngOnInit(): void {

   this.produitService.listeProduit().subscribe(prods => {
     console.log(prods);
     this.allProduits = prods;
     });
 }

 onKeyUp(filterText : string){
  this.produits = this.allProduits.filter(item =>
item.nomProduit.toLowerCase().includes(filterText));
 }
```

20. Tester votre travail

Rechercher par nom des produits en utilisant un pipe

21. Créer le pipe `searchFilter` :

```
ng g pipe searchFilter
```

22. Modifier la méthode transform comme suit :

```
transform(list: any[], filterText: string): any {
    return list ? list.filter(item =>
item.nomProduit.toLowerCase().includes(filterText)) : [];
  }
```

23. Modifier le fichier recherche-par-nom.component.html comme suit :

```
<form class="form-inline">
        <div class="form-group mx-sm-3 mb-2 mt-2 ">
            <label>Nom Produit en tapant</label>
<input #nom [(ngModel)]="searchTerm"  name="searchTerm" class="form-control" >
        </div>
    </form>
```

```
 @for (produit of produits  | searchFilter: searchTerm ; track produit.idProduit)
```

24. Importer, au niveau du fichier recherche-par-nom.component.ts, le pipe SearchFilterPipe :

```
  imports: [FormsModule, CommonModule,SearchFilterPipe],
```

25. Modifier ngOninit() comme suit :

```
 ngOnInit(): void {
    this.produitService.listeProduit().subscribe(prods => {
      console.log(prods);
        this.produits = prods;
      });
  }
```

26. Tester votre travail