|  |  |
| --- | --- |
|  | Année universitaire : 2025-2026  **UP-Web**  Mohammed Ali Bellili |
| **Workshop :**  **Angular 17 Sans NgRx** | |

📘 Guide Complet - Angular 17 Sans NgRx

# 1. Création de la base de données (JSON Server)

Créez un fichier nommé generate-db.js avec le contenu suivant :

const fs = require('fs');  
  
const types = [  
 { type: 'liquide', count: 90 },  
 { type: 'solide', count: 100 },  
 { type: 'vapeur', count: 150 }  
];  
  
let produits = [];  
let id = 1;  
  
types.forEach(group => {  
 for (let i = 0; i < group.count; i++) {  
 produits.push({  
 id: id++,  
 libelle: `${group.type.toUpperCase()} Produit ${i + 1}`,  
 image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRehTnAOptWY65gyJ0kgMvRa6Qmn9fvLeGZlg&s',  
 prix: (Math.random() \* 100).toFixed(2),  
 type: group.type  
 });  
 }  
});  
  
const data = { produits };  
fs.writeFileSync('db.json', JSON.stringify(data, null, 2));  
console.log('✅ Fichier db.json généré avec 340 produits.');

Ensuite, exécutez :

node generate-db.js

Installez JSON Server puis exécutez :

npm install -g json-server  
json-server --watch db.json --port 3000

# 2. Configuration du Projet Angular 17

Créez un module et un service :

ng g module produit --routing  
ng g component produit/produit-list  
ng g s produit/service/produit

## 3. app.routes.ts

{  
 path: 'produit',  
 loadChildren: () => import('./produit/produit.module').then(m => m.ProduitModule)  
},  
{ path: '', redirectTo: 'produit', pathMatch: 'full' }

## 4. produit-routing.module.ts

{ path: '', component: ProduitListComponent }

## 5. produit.module.ts

imports: [  
 CommonModule,  
 HttpClientModule,  
 ProduitRoutingModule,  
 ProduitListComponent  
]

# 6. Modèle Produit : produit/model/produit.ts :

export class Produit {  
 constructor(  
 public id: number,  
 public libelle: string,  
 public image: string,  
 public prix: number,  
 public type: string  
 ) {}  
}

# 7. produit.service.ts

import { HttpClient } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Observable } from 'rxjs';

import { Produit } from '../model/produit';

@Injectable({ providedIn: 'root' })  
export class ProduitService {  
 private API = 'http://localhost:3000/produits';  
 constructor(private http: HttpClient) {}  
 getProduits(): Observable<Produit[]> {  
 return this.http.get<Produit[]>(this.API);  
 }  
}

# 8. produit-list.component.ts

import { Component, OnInit } from '@angular/core';

import { ProduitService } from '../service/produit.service';

import { Produit } from '../model/produit'; // <-- importe Produit

import { CommonModule } from '@angular/common';

import { HttpClientModule } from '@angular/common/http';

@Component({

selector: 'app-produit-list',

standalone: true,

imports: [CommonModule, HttpClientModule],

templateUrl: './produit-list.component.html',

styleUrls: ['./produit-list.component.css']

})  
export class ProduitListComponent implements OnInit {  
 produits: Produit[] = [];  
 constructor(private produitService: ProduitService) {}  
 ngOnInit(): void {  
 this.produitService.getProduits().subscribe(data => this.produits = data);  
 }  
}

# 9. produit-list.component.html

Affichage des produits :

<div class="container mt-4">

<h2>Liste des Produits</h2>

<table class="table table-striped table-bordered">  
<thead>

<tr>

<th>Libellé</th>

<th>Image</th>

<th>Prix (€)</th>

<th>Type</th>

</tr>

</thead>  
<tbody>  
 <tr \*ngFor="let produit of produits">  
 <td>{{ produit.libelle }}</td>  
 <td><img [src]="produit.image" width="50" /></td>  
 <td>{{ produit.prix }}</td>  
 <td>{{ produit.type }}</td>  
 </tr>  
</tbody>  
</table>

</div>

# 10. Ajout de Bootstrap

npm install bootstrap

"styles": [  
 "node\_modules/bootstrap/dist/css/bootstrap.min.css",  
 "src/styles.css"  
]

# 11. app.config.ts

providers: [  
 provideHttpClient(withFetch())  
]