

## Atelier 02 Angular :

### Création de formulaire d'ajout des produits

#### Objectifs :

1. Création de la classe **Modèle** Produit,
2. Création d'un **formulaire** pour ajouter un nouveau produit,
3. Création du **Service** produit.

#### Création de la classe Modèle Produit

1. Créer sous le dossier app, le dossier model, puis y créer le fichier produit.model.ts :

```
export class Produit {  
  idProduit? : number;  
  nomProduit? : string;  
  prixProduit? : number;  
  dateCreation? : Date ;  
}
```

2. Au niveau de la classe ProduitsComponent, modifier la déclaration de l'attribut produits comme suit :

```
import { Produit } from '../model/produit.model';  
  
export class ProduitsComponent implements OnInit {  
  
  produits : Produit[]; //un tableau de Produit
```

3. Au niveau du constructeur de la classe, affecter trois produits au tableau produits :

```
constructor() {  
    this.produits = [  
        {idProduit : 1, nomProduit : "PC Asus", prixProduit : 3000.600, dateCreation : new Date("01/14/2011")},  
        {idProduit : 2, nomProduit : "Imprimante Epson", prixProduit : 450, dateCreation : new Date("12/17/2010")},  
        {idProduit : 3, nomProduit : "Tablette Samsung", prixProduit : 900.123, dateCreation : new Date("02/20/2020")}  
    ];  
}
```

4. Modifier le fichier produits.component.html comme suit :

```
<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> </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>  
            </tr>  
          </tbody>  
        }  
      </table>  
    </div>  
  </div>  
</div>
```

Id	Nom	Prix	Date Création
1	PC Asus	3000.6	14/01/2011
2	Imprimante Epson	450	17/12/2010
3	Tablette Samsung	900.123	20/02/2020

## Utiliser l'attribut @empty

```
@empty {  
  <h1>Aucun produit trouvé !</h1>  
}
```

## Utiliser l'attribut \$index

```
@for (produit of produits; track produit.idProduit; let index =$index)  
  
<td>{{index}}</td>
```

N°	Id	Nom	Prix	Date Création
0	1	PC Asus	3000.6	14/01/2011
1	2	Imprimante Epson	450	17/12/2010
2	3	Tablette Samsung	900.123	20/02/2020

## Utiliser l'attribut \$count

```
@for (produit of produits; track produit.idProduit; let index =$index; let nbr=$count)
```

```
@if(index==0)  
{ <h1>Nombre : {{nbr}}</h1>}
```

## Utiliser l'attribut \$first et \$last

```
@for (produit of produits; track produit.idProduit; let index =$index; let nbr=$count;  
let first = $first)
```

```
@if(first)  
{ <h1>Nombre : {{nbr}}</h1>}
```

## Création d'un formulaire pour ajouter un nouveau produit

5. Pour pouvoir utiliser le « Two way Binding » avec la directive **ngModel**, il faut importer le module **FormsModule** dans le fichier **add-produit.component.ts** :

```
import { FormsModule } from '@angular/forms';
...
imports: [
  FormsModule
]
...
```

6. Ajouter l'attribut **newProduit** à la classe **AddProduitComponent** :

```
import { Produit } from '../model/produit.model';
export class AddProduitComponent implements OnInit {

  newProduit = new Produit();
```

7. Ajouter la méthode **addProduit()** à la classe **AddProduitComponent** :

```
addProduit(){
  console.log(this.newProduit);
}
```

8. Modifier le fichier **add-produit.component.html** comme suit :

```
<div class="container">
  <div>
    <h2>Ajouter un Produit :</h2>
  </div>
  <form >

    <div class="col-sm-2 col-md-2 col-lg-2">
      <label >ID  Produit</label>
      <input  type="text" [(ngModel)]="newProduit.idProduit"
        name="idProduit" class="form-control">
    </div>

    <div class="col-sm-4 col-md-4 col-lg-4" >
      <label >Nom Produit</label>
      <input  type="text"[(ngModel)]= "newProduit.nomProduit"
        name="nomProduit" class="form-control">
    </div>
  </form>
</div>
```

```

</div>

<div class="col-sm-2 col-md-2 col-lg-2">
  <label >Prix Produit</label>
  <input type="number" [(ngModel)]= "newProduit.prixProduit"
    name="prixProduit" class="form-control">
</div>

<div class="col-sm-4 col-md-4 col-lg-4">
  <label >Date création</label>
  <input type="date" [(ngModel)]= "newProduit.dateCreation"
    name="dateCreation" class="form-control">
</div>

<div class="mt-2">
  <button type="submit" (click)="addProduit()" class="btn btn-success">Ajouter</button>
</div>
</form>
</div>

```

The screenshot shows a web browser window with the title 'MesProduits'. The address bar shows 'localhost:4200/add-produit'. The page has a dark header with 'Gestion Produits' and navigation links 'Home' and 'Produits'. A user profile dropdown shows '[username]'. The main content area is titled 'Ajouter un Produit :'. It contains four input fields: 'ID Produit', 'Nom Produit', 'Prix Produit', and 'Date création' (which has a calendar icon). Below these fields is a green button labeled 'Ajouter'.

## Création du Service produit

9. Créer sous app le dossier services, et y créer le service produit :

```
cd .\src\app\services\
```

**ng g service produit**

10. Modifier le fichier produit.service.ts comme suit :

```

import { Injectable } from '@angular/core';
import { Produit } from '../model/produit.model';

@Injectable({
  providedIn: 'root'
})
export class ProduitService {
  produits : Produit[]; //un tableau de Produit
  constructor() {
    this.produits = [
      { idProduit : 1, nomProduit : "PC Asus", prixProduit : 3000.600, dateCreation : new Date("01/14/2011") },
      { idProduit : 2, nomProduit : "Imprimante Epson", prixProduit : 450, dateCreation : new Date("12/17/2010") },
      { idProduit : 3, nomProduit : "Tablette Samsung", prixProduit : 900.123, dateCreation : new Date("02/20/2020") }
    ];
  }

  listeProduits():Produit[] {
    return this.produits;
  }

  ajouterProduit( prod: Produit){
    this.produits.push(prod);
  }
}

```

11. Modifier le fichier produits.component.ts comme suit :

```

import { Component, OnInit } from '@angular/core';
import { Produit } from '../model/produit.model';
import { ProduitService } from '../services/produit.service';
...
export class ProduitsComponent implements OnInit {

  produits : Produit[]; //un tableau de Produit
  constructor(private produitService: ProduitService ) {
    this.produits = produitService.listeProduits();
  }
}

```

```
...  
}
```

12. Modifier le fichier add-produit.component.ts comme suit :

```
import { Component, OnInit } from '@angular/core';  
import { Produit } from '../model/produit.model';  
import { ProduitService } from '../services/produit.service';  
  
export class AddProduitComponent implements OnInit {  
  newProduit = new Produit();  
  constructor(private produitService: ProduitService) { }  
  
  addProduit(){  
    // console.log(this.newProduit);  
    this.produitService.ajouterProduit(this.newProduit);  
  }  
}
```

Création d'une interface au lieu d'une classe :

```
export interface Produit {  
  idProduit? : number;  
  nomProduit? : string;  
  prixProduit? : number;  
  dateCreation? : Date ;  
}
```