

# DEVOIR DEVELOPPEMENT WEB: ANGULAR

Realisé par Ameni Lahouar.

## Partie 1:

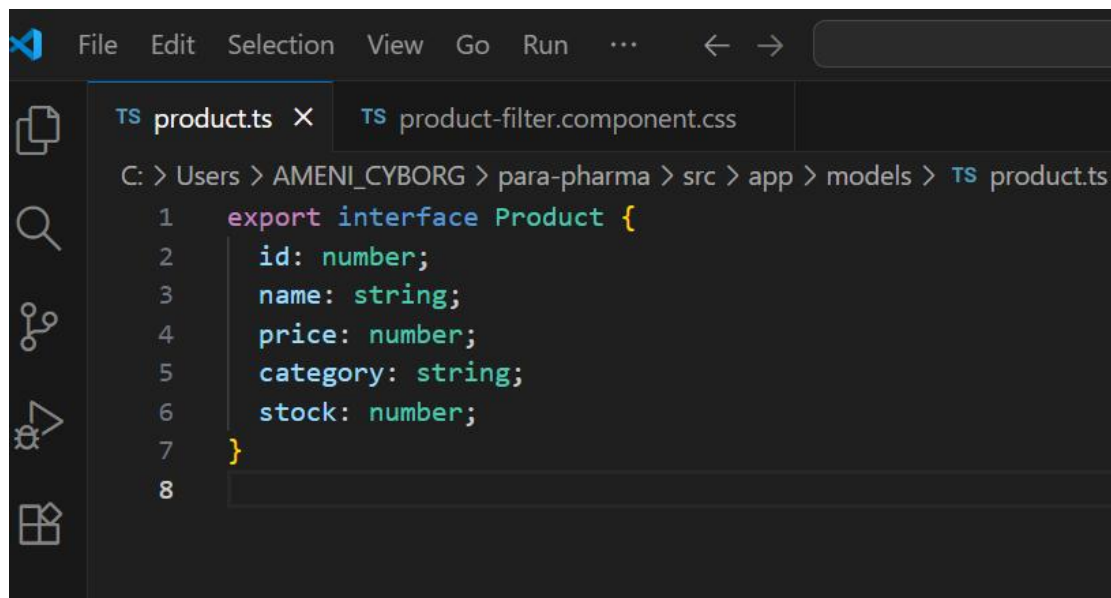
On a fait les commandes: `ng new para-pharma --standalone`  
`cd para-pharma`

## Partie 2:

On a fait la commande: `ng generate component components/product-filter --standalone`

```
✓ Packages installed successfully.
'git' is not recognized as an internal or external command,
operable program or batch file.
PS C:\Users\AMENI_CYBORG> cd para-pharma
PS C:\Users\AMENI_CYBORG\para-pharma> ng generate component components/product-filter --standalone
? Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.io/analytics. No
Global setting: enabled
Local setting: disabled
Effective status: disabled
CREATE src/app/components/product-filter/product-filter.component.html (29 bytes)
CREATE src/app/components/product-filter/product-filter.component.spec.ts (604 bytes)
CREATE src/app/components/product-filter/product-filter.component.ts (328 bytes)
CREATE src/app/components/product-filter/product-filter.component.css (0 bytes)
PS C:\Users\AMENI_CYBORG\para-pharma> |
```

## Partie3:

A screenshot of the Visual Studio Code editor interface. The top toolbar shows icons for File, Edit, Selection, View, Go, Run, and a search icon. Below the toolbar, there are two tabs: 'TS product.ts' (active) and 'TS product-filter.component.css'. The main editor area displays the content of 'product.ts' at the path 'C:\Users\AMENI\_CYBORG\para-pharma\src\app\models\product.ts'. The code defines an interface 'Product' with the following properties: 'id' (number), 'name' (string), 'price' (number), 'category' (string), and 'stock' (number).

```
1  export interface Product {
2      id: number;
3      name: string;
4      price: number;
5      category: string;
6      stock: number;
7  }
8
```

## Partie4:

```
File Edit Selection View Go Run ... Search
TS products TS product-filter.components
C:\Users\AMENI_CYBORG> para-pharma > src > app > components > product-filter > TS product-filter.components > ...
1 import { Component } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { Product } from '../models/product';
4 @Component({
5   selector: 'app-product-filter',
6   standalone: true,
7   imports: [FormsModule],
8   templateUrl: './product-filter.component.html',
9   styleUrls: ['./product-filter.component.css']
10 })
11 export class ProductFilterComponent {
12   searchText: string = '';
13   products: Product[] = [
14     { id: 1, name: 'Crème hydratante', price: 15, category: 'Soins', stock: 10 },
15     { id: 2, name: 'Gel douche', price: 5, category: 'Hygiène', stock: 0 },
16     { id: 3, name: 'Vitamine C', price: 12, category: ' Compléments', stock: 3 },
17     { id: 4, name: 'Shampooing antipelliculaire', price: 8, category: 'Cheveux', stock: 7 },
18     { id: 5, name: 'Masque visage', price: 20, category: 'Soins', stock: 2 },
19   ];
20   get filteredProducts(): Product[] {
21     const search = this.searchText.trim().toLowerCase();
22     if (!search) return this.products;
23     return this.products.filter(p => p.name.toLowerCase().includes(search));
24   }
25   stockColor(stock: number): string {
26     if (stock === 0) return 'red';
27     if (stock < 5) return 'orange';
28     return 'green';
29   }
30 }
```

## Partie5:

```
File Edit Selection View Go Run ... Search
TS products TS product-filter.components product-filter.component.html
C:\Users\AMENI_CYBORG> para-pharma > src > app > components > product-filter > product-filter.component.html > ...
1 <section class="product-filter">
2   <h1>Liste des produits</h1>
3
4   <!-- Barre de recherche -->
5   <input type="text" placeholder="Rechercher un produit..." [(ngModel)]="searchText" />
6
7   <!-- Liste des produits filtrés -->
8   <ul *ngIf="filteredProducts.length > 0; else noResult">
9     <li *ngFor="let p of filteredProducts; trackBy: p.id">
10       <span class="product-name">{{ p.name }}</span>
11       <span class="product-category">{{ p.category }}</span>
12       <span class="product-price">{{ p.price | currency:'EUR' }}</span>
13       <span class="product-stock" [style.color]="stockColor(p.stock)">
14         Stock: {{ p.stock }}
15       </span>
16     </li>
17   </ul>
18
19   <!-- Message état vide -->
20   <ng-template #noResult>
21     <p>Aucun produit correspondant.</p>
22   </ng-template>
23 </section>
24
25
```

## Partie6:

```
19 ];
20 get filteredProducts(): Product[] {
21   const search = this.searchText.trim().toLowerCase();
22   if (!search) return this.products;
23   return this.products.filter(p => p.name.toLowerCase().includes(search));
24 }
25 stockColor(stock: number): string {
```

## Partie7:

**La partie VII** concerne l’affichage des produits filtrés avec des **directives conditionnelles et de boucle**.

Utiliser `*ngIf` (ou `@if` en Angular 20) pour afficher soit la liste, soit le message “aucun résultat”.

Utiliser `*ngFor` (ou `@for`) pour parcourir `filteredProducts`.

Ajouter `trackBy: p.id` pour optimiser le rendu et éviter de recréer les éléments inutilement.

Colorer dynamiquement le stock avec  
`[style.color]="stockColor(p.stock) "`.

### Résultat attendu :

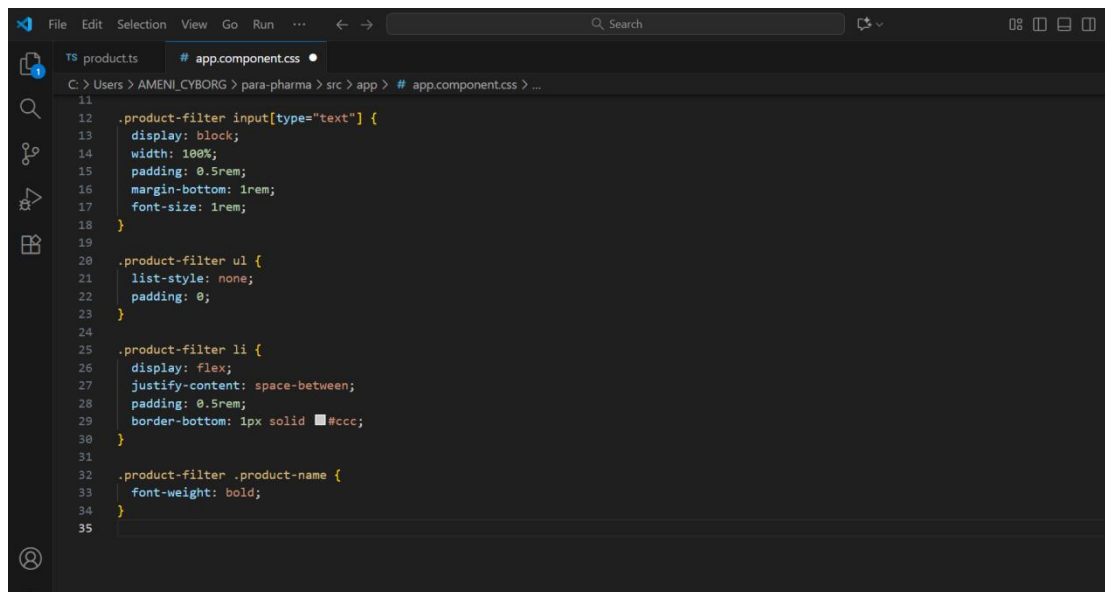
La liste des produits se met à jour en direct lors de la saisie dans le champ de recherche.

Si aucun produit ne correspond → affichage du message “Aucun produit correspondant”.

Les couleurs du stock s’appliquent automatiquement selon la quantité.

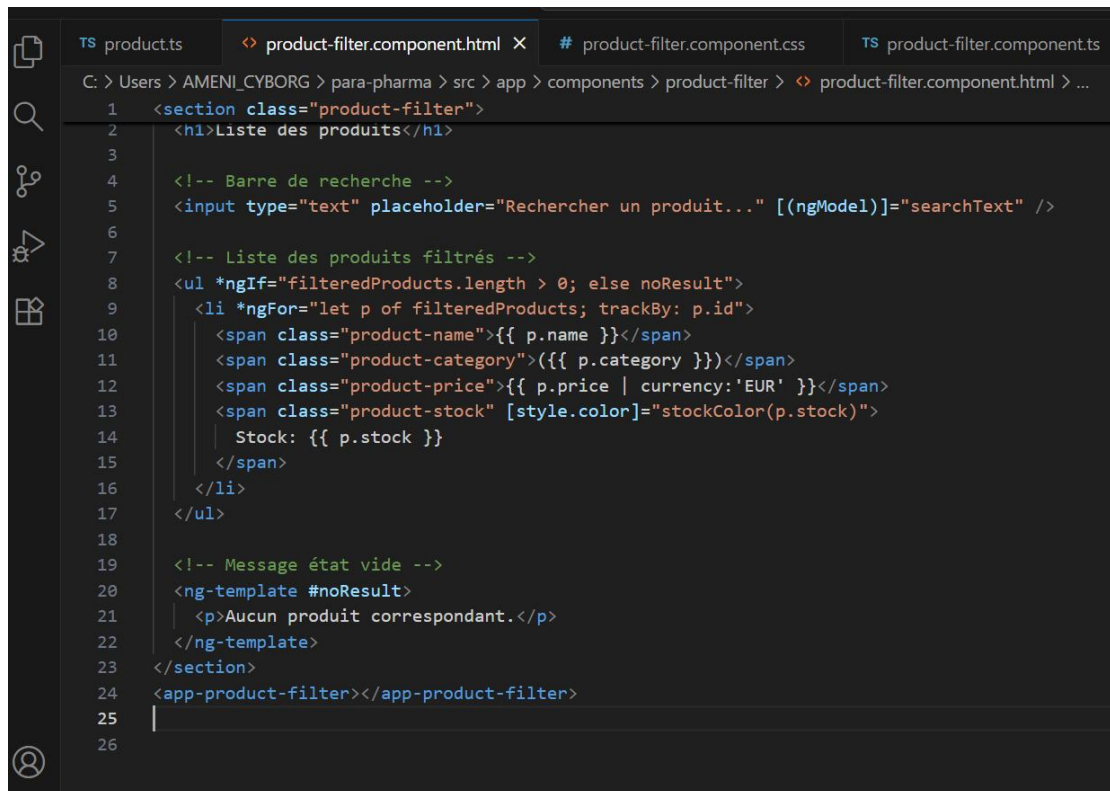
Il est aussi possible de mettre à jour le code pour utiliser la **nouvelle syntaxe Angular 20** `@if/@for`.

## Partie8:



```
11
12 .product-filter input[type="text"] {
13   display: block;
14   width: 100%;
15   padding: 0.5rem;
16   margin-bottom: 1rem;
17   font-size: 1rem;
18 }
19
20 .product-filter ul {
21   list-style: none;
22   padding: 0;
23 }
24
25 .product-filter li {
26   display: flex;
27   justify-content: space-between;
28   padding: 0.5rem;
29   border-bottom: 1px solid #ccc;
30 }
31
32 .product-filter .product-name {
33   font-weight: bold;
34 }
35
```

## Partie9:



```
1 <section class="product-filter">
2   <h1>Liste des produits</h1>
3
4   <!-- Barre de recherche -->
5   <input type="text" placeholder="Rechercher un produit..." [(ngModel)]="searchText" />
6
7   <!-- Liste des produits filtrés -->
8   <ul *ngIf="filteredProducts.length > 0; else noResult">
9     <li *ngFor="let p of filteredProducts; trackBy: p.id">
10      <span class="product-name">{{ p.name }}</span>
11      <span class="product-category">{{ p.category }}</span>
12      <span class="product-price">{{ p.price | currency:'EUR' }}</span>
13      <span class="product-stock" [style.color]="stockColor(p.stock)">
14        Stock: {{ p.stock }}
15      </span>
16    </li>
17  </ul>
18
19  <!-- Message état vide -->
20  <ng-template #noResult>
21    <p>Aucun produit correspondant.</p>
22  </ng-template>
23 </section>
24 <app-product-filter></app-product-filter>
25
26
```

on a ajouté la dernière ligne

## Partie10:

On execute maintenant et ouvre le site web:

