TP 3 : Marketplace dynamique avec RxJS

_____

1- Objectif de TP3 :

Ce TP a pour but de mettre en pratique les concepts d'Angular, RxJS, Observables et les Subjects tout en développant et améliorant le TP 2 pour gérer un panier d'achat en ligne.

Description : L'application doit permettre à l'utilisateur de voir une liste de produits, d'ajouter des produits à un panier d'achat, de voir les produits dans le panier, de modifier la quantité de chaque produit dans le panier et de supprimer un produit du panier.

2- Sujet à aborder :

1. RxJS / Observables : Ils peuvent être utilisés pour gérer le flux de données entre les composants et les services.
2. RxJS / Subjects : Ils peuvent être utilisés pour représenter le panier d'achat et permettre à plusieurs parties de l'application de s'y abonner et de réagir aux changements.
3. Les fonctions subscribe / next / error / complete : Elles sont utilisées pour manipuler les valeurs émises par les Observables et les Subjects.

3- Livrable sur place :

L'application doit permettre à l'utilisateur de :

- Voir une liste de produits.
- Ajouter des produits à un panier d'achat
- Visualiser le compteur des éléments ajoutés au panier
- Pouvoir ajouter un seul élément de produit
- Changement automatique de bouton “ajouter au panier” au “retirer du panier” si le produit est déjà ajouté.
- Changement de couleur de compteur si le chiffre est supérieur à 0.

TP2 : Marketplace

Categories Nouveautés Contactez-nous Panier 0

Welcome to Angular Training Marketplace

ADIDAR MAX
ADULT

$65

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.

+ Add to cart

LEBRON MAX AIR
KIDS

$65

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.

+ Add to cart

PUMA XS
ADULT

$4.99

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.

+ Add to cart

TP2 : Marketplace

Categories Nouveautés Contactez-nous Panier 2

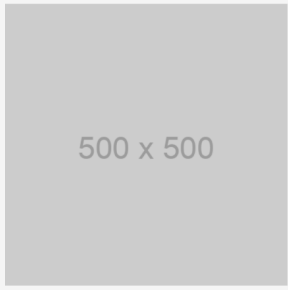
Welcome to Angular Training Marketplace

ADIDAR MAX
ADULT

$65

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.

- Remove from cart



LEBRON MAX AIR
KIDS

$65

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.

- Remove from cart

PUMA XS
ADULT

$4.99

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups.

+ Add to cart

3- Actions attendues :

1. Mettre à jour le type `MarketplaceItemType` (src/app/types/marketplace.type.ts):

```ts
export type MarketplaceItemType = {
    id: number,
    title: string,
    category: string,
    image: string,
    description: string,
    price: number,
    isSelected: boolean
}
```

2. Créer un service **`ProductService`** qui fournira une liste de produits à afficher (au lieu de `marketplace-item-list.component.ts`).

```
$ ng g s services/product
```

```ts
import { Injectable } from '@angular/core';
import { MarketplaceItemType } from '../types/marketplace.type';
import { of, Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ProductService {

  private _products: MarketplaceItemType[] = [...];

  constructor() { }

  getProducts = (): Observable<Array<MarketplaceItemType>> => {
    return of(this._products);
  }

  markProductAsSelected = (item: MarketplaceItemType) => {
    item.isSelected = true;
  }

  markProductAsUnselected = (item: MarketplaceItemType) => {
    item.isSelected = false;
  }
}
```

3. Créer un service **`CartService`** qui va gérer le panier utilisateur. On utilise un objet de type `BehaviorSubject` dans ce service pour représenter le panier d'achat. Le panier doit être un tableau d'objets, chaque objet représentant un produit et la quantité ajoutée au panier.

```
$ ng g s services/cart
```

```typescript
import { Injectable } from '@angular/core';
import { MarketplaceItemType } from '../types/marketplace.type';
import { BehaviorSubject } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class CartService {

  private _cartItems = new BehaviorSubject<Array<{ item: MarketplaceItemType, quantity: number }>>([]);

  constructor() { }

  getCartItems = () => this._cartItems;

  addItem = (item: MarketplaceItemType, quantity = 1) => {
    const currentCartItems = this._cartItems.getValue();
    const searchItem = currentCartItems.find(i => i.item.id === item.id);
    if (searchItem) {
      searchItem.quantity += quantity;
    } else {
      currentCartItems.push({
        item,
        quantity
      });
    }
    this._cartItems.next(currentCartItems);
  }

  removeItem = (item: MarketplaceItemType) => {
    let currentCartItems = this._cartItems.getValue();
    currentCartItems = currentCartItems.filter(i => i.item.id !== item.id);
    this._cartItems.next(currentCartItems);
  }
}
```

4. Dans le composant de la liste de produits **marketplace-item-list.component.ts**, chaque produit doit avoir un bouton "Ajouter au panier" qui, lorsqu'il est cliqué, ajoute le produit au panier et un bouton “Retirer du panier” qui retire le produit du panier, en utilisant le service **`CartService`**.

> *src/app/marketplace/marketplace-item-list/marketplace-item-list.component.ts*

```ts
import { Component, OnDestroy, OnInit } from '@angular/core';
import { MarketplaceItemType } from '../../types/marketplace.type';
import { CommonModule } from '@angular/common';
import { ProductService } from '../../services/product.service';
import { CartService } from '../../services/cart.service';
import { Subscription } from 'rxjs';

@Component({
  selector: 'app-marketplace-item-list',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './marketplace-item-list.component.html',
  styleUrl: './marketplace-item-list.component.scss'
})
export class MarketplaceItemListComponent implements OnInit, OnDestroy {

  productsSub!: Subscription;
  marketplaceItems: MarketplaceItemType[] = [];

  constructor(
    public productsService: ProductService,
    public cartService: CartService,
  ) { }

  ngOnInit(): void {
    this.productsSub = this.productsService.getProducts().subscribe(products => {
      this.marketplaceItems = products;
    });
  }

  addToCart = (item: MarketplaceItemType) => {
    this.productsService.markProductAsSelected(item);
    this.cartService.addItem(item);
  }

  removeFromCart = (item: MarketplaceItemType) => {
    this.productsService.markProductAsUnselected(item);
    this.cartService.removeItem(item);
  }

  ngOnDestroy(): void {
    this.productsSub.unsubscribe();
  }
}
```

> *src/app/marketplace/marketplace-item-list/marketplace-item-list.component.html*

```html
<div class="marketplace-list">
    <div class="single-item" *ngFor="let item of marketplaceItems">
        <div class="left-set">
            <img [src]="item.image" [alt]="item.title" />
        </div>
        <div class="right-set">
            <div class="name">{{ item.title | uppercase }}</div>
            <div class="subname">{{ item.category }}</div>
            <div class="price">${{ item.price }}</div>
            <div class="description">
                <p>
                    {{ item.description }}
                </p>
            </div>
            <div class="cart">
                <button class="add_to_cart" *ngIf="!item.isSelected" (click)="addToCart(item)">+ Add to cart</button>
                <button class="remove_from_cart" *ngIf="item.isSelected" (click)="removeFromCart(item)">- Remove from cart</button>
            </div>
        </div>
    </div>
</div>
```

> *src/app/marketplace/marketplace-item-list/marketplace-item-list.component.scss*

```scss
.left-set,
.right-set {
    position: relative;
    width: 50%;
    height: 100%;
    div.cart {
        position: absolute;
        bottom: 25px;
        button {
            cursor: pointer;
            border: 0;
            border-radius: 5px;
            padding: 10px;
            &.add_to_cart {
                background: #2aa200;
                color: white;
            }
            &.remove_from_cart {
                background: #e32525;
                color: white;
            }
        }
    }
}
```

5. Dans le composant du panier (**HeaderComponent**), Il faut synchroniser la liste des produits dans le panier pour pouvoir afficher le compteur.. Utilisez la fonction subscribe pour s'abonner au `BehaviorSubject` dans le service **CartService** et réagir aux changements dans le panier.

> *src/app/header/header.component.ts*

```ts
import { CommonModule } from '@angular/common';
import { Component, OnDestroy, OnInit } from '@angular/core';
import { CartService } from '../services/cart.service';
import { MarketplaceItemType } from '../types/marketplace.type';
import { Subscription } from 'rxjs';

@Component({
  selector: 'app-header',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './header.component.html',
  styleUrl: './header.component.scss'
})
export class HeaderComponent implements OnInit, OnDestroy {

  cartItems: { item: MarketplaceItemType, quantity: number }[] = [];
  cartItemsSub!: Subscription;

  constructor(
    public cartService: CartService,
  ) {}

  ngOnInit(): void {
    this.cartItemsSub = this.cartService.getCartItems().subscribe(cartItems => {
      this.cartItems = cartItems;
    });
  }

  ngOnDestroy(): void {
    this.cartItemsSub.unsubscribe();
  }
}
```

> *src/app/header/header.component.html*

```html
<header class="site-header">
    <div class="site-identity">
        <a href="#"><img src="https://via.placeholder.com/200" alt="TP2 : Marketplace" /></a>
        <h1><a href="#">TP2 : Marketplace</a></h1>
    </div>
    <nav class="site-navigation">
        <ul class="nav">
            <li><a href="#">Categories</a></li>
            <li><a href="#">Nouveautés</a></li>
            <li><a href="#">Contactez-nous</a></li>
            <li><a href="#">
                Panier <span class="badge" [ngClass]="{ 'badge-green': cartItems.length > 0, 'badge-yellow': cartItems.length === 0 }">{{ cartItems.length }}</span>
            </a></li>
        </ul>
    </nav>
</header>
```

> *src/app/header/header.component.scss*

```scss
...
span.badge {
    padding: 2px 6px;
    border-radius: 50%;
    font-size: 14px;
    &.badge-green {
        background: #2aa200;
        color: white;
    }
    &.badge-yellow {
        background: #e32525;
        color: white;
    }
}
```

3- Livrable à remettre avant la séance prochaine :

6. Dans le composant du **HeaderComponent**, lister les produits dans le panier (via un modal ou un right sidebar).
7. Dans le composant du **HeaderComponent** et depuis la liste des produits dans le panier, ajouter la possibilité de modifier la quantité de chaque produit et de supprimer un produit du panier.
8. Gérer les cas lors de l'ajout d'un produit existant dans le panier ou lors de la définition d'une quantité négative.