import { CartItem } from './../../models/cart-item';

import { Product } from "./../../models/product";

import { ProductService } from "./../../services/product.service";

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

import { ActivatedRoute } from "@angular/router";

import { CartService } from "src/app/services/cart.service";

@Component({

selector: "app-product-list",

templateUrl: "./product-list.component.html",

styleUrls: ["./product-list.component.css"],

})

export class ProductListComponent implements OnInit {

products: Product[] = [];

currentCategory: number = 1;

previousCategory: number = 1;

searchMode: boolean = false;

// pagination

thePageNumber: number = 1;

thePageSize: number = 10;

theTotalElements: number = 0;

constructor(

private productService: ProductService,

private route: ActivatedRoute,

private cartService: CartService

) { }

ngOnInit() {

this.route.paramMap.subscribe(() => {

this.listProducts();

});

}

listProducts() {

this.searchMode = this.route.snapshot.paramMap.has("keyword");

if (this.searchMode) {

this.handleSearchProducts();

} else {

this.handleListProducts();

}

}

handleSearchProducts() {

const keyWord: string = this.route.snapshot.paramMap.get("keyword");

this.productService

.searchProductsList(keyWord)

.subscribe((data) => {

this.products = data;

});

}

handleListProducts() {

// check if "id" parameter is available

const hasCaregoryId: boolean = this.route.snapshot.paramMap.has("id");

if (hasCaregoryId) {

// get the "id" parameter string, convert string to number using the "+" symbol.

this.currentCategory = +this.route.snapshot.paramMap.get("id");

} else {

// set default category id 1

this.currentCategory = 1;

}

// check if the currentCategoryId is different of previousCategoryId

if(this.previousCategory !== this.currentCategory){

this.thePageNumber = 1;

}

this.previousCategory = this.currentCategory;

this.productService

.getProductsListPaginate(this.thePageNumber - 1, this.thePageSize, this.currentCategory)

.subscribe(this.processResult());

}

processResult() {

return data =>{

this.products = data.\_embedded.products;

this.thePageNumber = data.page.number + 1;

this.thePageSize = data.page.size;

this.theTotalElements = data.page.totalElements;

}

}

addToCart(theProduct: Product){

const theCartItem = new CartItem(theProduct);

this.cartService.addToCart(theCartItem);

}

}