@Entity

@Table(name = "category")

public class ProductCategory {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id")

private Long id;

@Column(name = "category\_name")

private String categoryName;

@JsonManagedReference

@OneToMany(cascade = CascadeType.ALL, mappedBy = "category")

private Set<Product> products;

public ProductCategory() {

}

public ProductCategory (Long id) {

super();

this.id = id;

}

public Set<Product> getProducts() {

return products;

}

public void setProducts(Set<Product> products) {

this.products = products;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getCategoryName() {

return categoryName;

}

public void setCategoryName(String categoryName) {

this.categoryName = categoryName;

}

}

**Product.java:**

@Entity

@Table(name = "product")

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id")

private Long id;

@Column(name = "brand")

private String brand;

@Column(name = "name")

private String name;

@ManyToOne

@JoinColumn(name = "category\_id", nullable = false)

private ProductCategory category;

public Product(Long id, String brand, String name, ProductCategory category) {

this.id = id;

this.brand = brand;

this.name = name;

this.category = category;

}

public Product(){}

public ProductCategory getCategory() {

return category;

}

public void setCategory(ProductCategory category) {

this.category = category;

}

public Long getCategoryId() {

return category.getId();

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

{

"brand": "hp",

"name": "envy",

"category": [{

"id": 2

},{

"id": 2

}

]

}

Product.ts

export class Product {

id: string;

brand: string;

name: string;

category: ProductCategory;

}

Product-category.ts

export class ProductCategory {

id: number;

categoryName: Object;

}

Product-form.component.html

<div class="card my-5">

<div class="card-body">

<form (ngSubmit)="onSubmit()" #productForm="ngForm">

<div class="form-group">

<label for="name">Brand</label>

<input type="text" [(ngModel)]="product.brand"

class="form-control"

id="brand"

name="brand"

placeholder="Enter your name"

required #name="ngModel">

</div>

<div class="form-group">

<label for="email">Name</label>

<input type="text" [(ngModel)]="product.name"

class="form-control"

id="name"

name="name"

placeholder="Enter your name"

required #name="ngModel">

<div [hidden]="!name.pristine" class="alert alert-danger">name is required</div>

</div>

<div class="form-group">

<label for="category">Category id</label>

<select [(ngModel)]="product.category" name="category.id">

<option \*ngFor="let category of productCategories">{{ category.id }}</option>

</select>

</div>

<button type="submit" [disabled]="!productForm.form.valid"

class="btn btn-info">Submit</button>

</form>

</div>

Product-form.component.ts

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

import { Product } from 'src/app/common/product';

import { ActivatedRoute, Router } from '@angular/router';

import { ProductService } from 'src/app/services/product.service';

import { ProductCategory } from 'src/app/common/product-category';

@Component({

selector: 'app-product-form',

templateUrl: './product-form.component.html',

styleUrls: ['./product-form.component.css']

})

export class ProductFormComponent implements OnInit {

product: Product;

productCategories: ProductCategory[];

constructor(private route: ActivatedRoute,

private router: Router,

private productService: ProductService) {

this.product = new Product();

}

ngOnInit(): void {

this.listProductCategories();

}

onSubmit(){

this.productService.save(this.product).subscribe(

result => this.gotoProductList()

);

}

gotoProductList(): void {

this.router.navigate(['/products']);

}

listProductCategories() {

this.productService.getProductCategories().subscribe(

data => {

console.log('Categories' + JSON.stringify(data));

this.productCategories = data;

}

);

}

}

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

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

import { Product } from '../common/product';

import { Observable } from 'rxjs';

import { ProductCategory } from '../common/product-category';

@Injectable({

providedIn: 'root'

})

export class ProductService {

private productsUrl: string;

private categoryUrl: string;

constructor(private http: HttpClient) {

this.productsUrl= 'http://localhost:8080/products';

this.categoryUrl = 'http://localhost:8080/categories';

}

public findAll(): Observable<Product[]>{

return this.http.get<Product[]>(this.productsUrl);

}

public save(product: Product){

return this.http.post<Product>(this.productsUrl, product);

}

getProductCategories(): Observable<ProductCategory[]> {

return this.http.get<ProductCategory[]>(this.categoryUrl);

}

}