Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №1

по дисциплине "распределенные программные системы"

Тема работы «Обеспечение безопасности веб-приложений.»

Выполнил:

студент гр. ИСТ-120

Барынкин В.Д.

Принял:

Спирин И.В.

Владимир 2023

**1. Цель работы**

Реализовать представления, сервис и дао.

**2. Ход работы**

1. DAO

Код ProductInfoDAO

public interface ProductInfoDAO {

ProductInfo FindOne(String productid);

ProductInfo update(ProductInfo productInfo);

ProductInfo save(ProductInfo productInfo);

void delete(String productId);

}

Код ProductInfoDAOImpl

public class ProductInfoDAOimpl

{

@Autowired

ProductInfoRepository productInfoRepository;

@Autowired

CategoryService categoryService;

@Override

public ProductInfo findOne(String productId) {

ProductInfo productInfo = productInfoRepository.findFirstByProductId(productId);

return productInfo;

}

@Override

public ProductInfo update(ProductInfo productInfo) {

// if null throw exception

categoryService.findByCategoryType(productInfo.getCategoryType());

if(productInfo.getProductStatus() > 1) {

throw new MyException(ResultEnum.PRODUCT\_STATUS\_ERROR);

}

return productInfoRepository.save(productInfo);

}

@Override

public ProductInfo save(ProductInfo productInfo) {

return update(productInfo);

}

@Override

public void delete(String productId) {

ProductInfo productInfo = findOne(productId);

if (productInfo == null) throw new MyException(ResultEnum.PRODUCT\_NOT\_EXIST);

productInfoRepository.delete(productInfo);

}

}

Код ProductService

public interface ProductService {

private final ProductInfoDAO productInfoDAO;

public ProductService(ProductInfoDAO productInfoDAO)

{

this.productInfoDAO = productInfoDAO;  
 }

ProductInfo findOne(String productId)  
 {

return productInfoDAO.FindOne(productid);

}

// All selling products

Page<ProductInfo> findUpAll(Pageable pageable);

// All products

Page<ProductInfo> findAll(Pageable pageable);

// All products in a category

Page<ProductInfo> findAllInCategory(Integer categoryType, Pageable pageable);

// increase stock

void increaseStock(String productId, int amount);

//decrease stock

void decreaseStock(String productId, int amount);

ProductInfo offSale(String productId);

ProductInfo onSale(String productId);

ProductInfo update(ProductInfo productInfo)

{

return productInfoDAO.update(productInfo);

}

ProductInfo save(ProductInfo productInfo)

{

return productInfoDAO.save(productInfo);

}

void delete(String productId)

{

productInfoDAO.delete(productId);

}

}

Код ProductServiceImpl

@Service

public class ProductServiceImpl implements ProductService {

@Autowired

ProductInfoDAO productInfoRepository;

@Autowired

CategoryService categoryService;

@Override

public ProductInfo findOne(String productId) {

return productInfoDAO.FindOne(productid);

}

@Override

public Page<ProductInfo> findUpAll(Pageable pageable) {

return productInfoRepository.findAllByProductStatusOrderByProductIdAsc(ProductStatusEnum.UP.getCode(),pageable);

}

@Override

public Page<ProductInfo> findAll(Pageable pageable) {

return productInfoRepository.findAllByOrderByProductId(pageable);

}

@Override

public Page<ProductInfo> findAllInCategory(Integer categoryType, Pageable pageable) {

return productInfoRepository.findAllByCategoryTypeOrderByProductIdAsc(categoryType, pageable);

}

@Override

@Transactional

public void increaseStock(String productId, int amount) {

ProductInfo productInfo = findOne(productId);

if (productInfo == null) throw new MyException(ResultEnum.PRODUCT\_NOT\_EXIST);

int update = productInfo.getProductStock() + amount;

productInfo.setProductStock(update);

productInfoRepository.save(productInfo);

}

@Override

@Transactional

public void decreaseStock(String productId, int amount) {

ProductInfo productInfo = findOne(productId);

if (productInfo == null) throw new MyException(ResultEnum.PRODUCT\_NOT\_EXIST);

int update = productInfo.getProductStock() - amount;

if(update <= 0) throw new MyException(ResultEnum.PRODUCT\_NOT\_ENOUGH );

productInfo.setProductStock(update);

productInfoRepository.save(productInfo);

}

@Override

@Transactional

public ProductInfo offSale(String productId) {

ProductInfo productInfo = findOne(productId);

if (productInfo == null) throw new MyException(ResultEnum.PRODUCT\_NOT\_EXIST);

if (productInfo.getProductStatus() == ProductStatusEnum.DOWN.getCode()) {

throw new MyException(ResultEnum.PRODUCT\_STATUS\_ERROR);

}

productInfo.setProductStatus(ProductStatusEnum.DOWN.getCode());

return productInfoRepository.save(productInfo);

}

@Override

@Transactional

public ProductInfo onSale(String productId) {

ProductInfo productInfo = findOne(productId);

if (productInfo == null) throw new MyException(ResultEnum.PRODUCT\_NOT\_EXIST);

if (productInfo.getProductStatus() == ProductStatusEnum.UP.getCode()) {

throw new MyException(ResultEnum.PRODUCT\_STATUS\_ERROR);

}

productInfo.setProductStatus(ProductStatusEnum.UP.getCode());

return productInfoRepository.save(productInfo);

}

@Override

public ProductInfo update(ProductInfo productInfo) {

// if null throw exception

categoryService.findByCategoryType(productInfo.getCategoryType());

if(productInfo.getProductStatus() > 1) {

throw new MyException(ResultEnum.PRODUCT\_STATUS\_ERROR);

}

return productInfoDAO.update(productInfo);

}

@Override

public ProductInfo save(ProductInfo productInfo) {

return productInfoDAO.save(productInfo);

}

@Override

public void delete(String productId) {

productInfoDAO.delete(productId);

}

}

Index.ftl

<html>

<#include "../partials/\_header.ftl">

<body>

<div class="container ">

<#include "../partials/\_nav.ftl">

<#include "../partials/\_cards.ftl">

<#include "../partials/\_pagination.ftl">

</div>

<#include "../partials/\_footer.ftl">

</body>

</html>

\_cards.ftl

<div class="row card-deck text-center">

<#list products.content as productInfo>

<div class="col-lg-4 ">

<div class="card mb-4 ">

<img height="10%" class="card-img-top" src="${productInfo.getProductIcon()}">

<div class="card-body">

<h4 class="card-title ">${productInfo.getProductName()}</h4>

<div class="text-left">

<p class="card-test"><strong>Описание: </strong>${productInfo.getProductDescription()}</p>

<p class="card-text"><strong>Цена: </strong>${productInfo.getProductPrice()?string.currency}</p>

<p class="card-text"><strong>Количество: </strong>${productInfo.getProductStock()}</p>

</div>

<a class="btn btn-primary btn-lg <#if productInfo.getProductStatus()==1>disabled</#if>"

href="/product/${productInfo.getProductId()}">Добавить в корзину</a>

</div>

</div>

</div>

</#list>

</div>

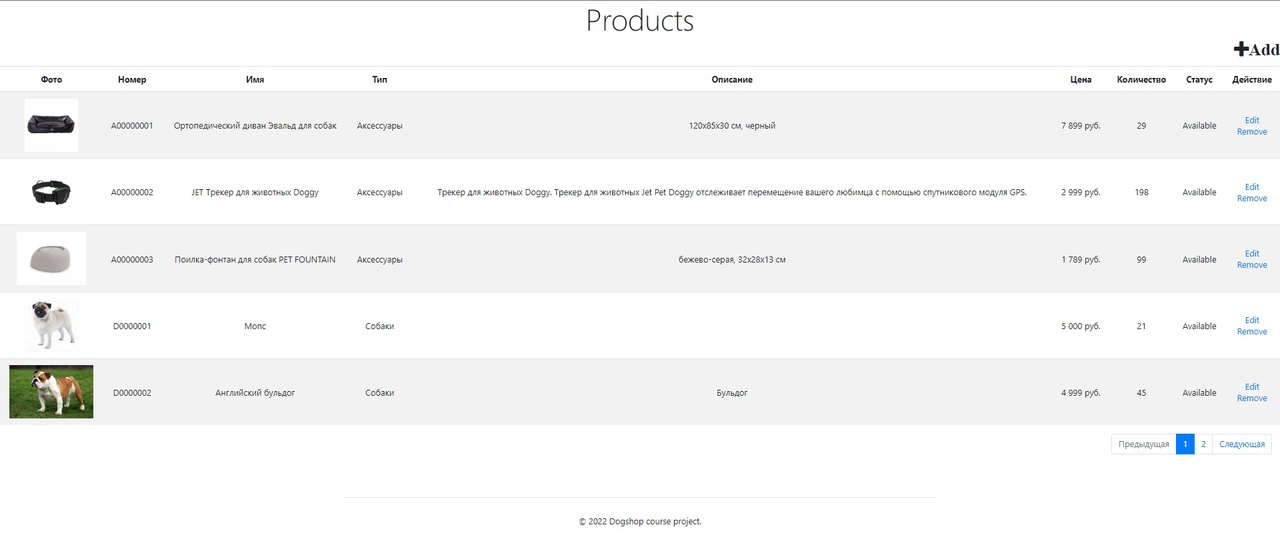
Скриншоты работы:  


Рисунок 1. Список продуктов

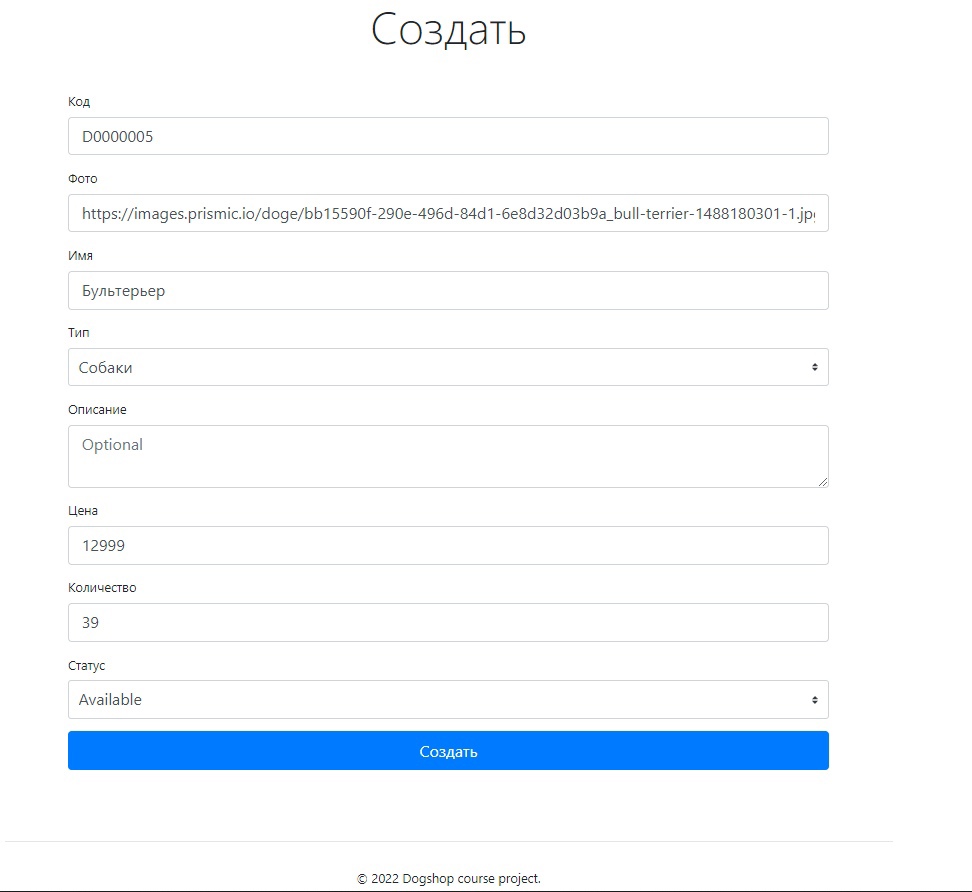


Рисунок 2. Создать

**3. Вывод**

Были реализованы сервис, представление и DAO.