

## ✓ Case Study: Product-Order Management System (With Mockito Testing) - Spring Test

### Product.java

```
package com.example.productordermanagementsystem.entity;
import jakarta.persistence.*;
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long productId;

    private String name;
    private double price;
    private int availableQuantity;

    public Product() {}
    public Product(Long productId, String name, double price, int availableQuantity) {
        this.productId = productId;
        this.name = name;
        this.price = price;
        this.availableQuantity = availableQuantity;
    }
    public Long getProductId() {
        return productId;
    }
    public void setProductId(Long productId) {
        this.productId = productId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public int getAvailableQuantity() {
        return availableQuantity;
    }
    public void setAvailableQuantity(int availableQuantity) {
        this.availableQuantity = availableQuantity;
    }
}
```

```
}
```

## Order.java

```
package com.example.productordermanagementsystem.entity;
import java.util.Date;
import jakarta.persistence.*;
@Entity
@Table(name = "orders")
public class Order {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long orderId;

    @ManyToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "product_id")
    private Product product;

    @Temporal(TemporalType.TIMESTAMP)
    private Date orderDate;

    private int quantityOrdered;
    public Order() {}
    public Order(Long orderId, Product product, Date orderDate, int quantityOrdered) {
        this.orderId = orderId;
        this.product = product;
        this.orderDate = orderDate;
        this.quantityOrdered = quantityOrdered;
    }
    public Long getOrderId() {
        return orderId;
    }
    public void setOrderId(Long orderId) {
        this.orderId = orderId;
    }
    public Product getProduct() {
        return product;
    }
    public void setProduct(Product product) {
        this.product = product;
    }
    public Date getOrderDate() {
        return orderDate;
    }
    public void setOrderDate(Date orderDate) {
        this.orderDate = orderDate;
    }
    public int getQuantityOrdered() {
```

```

        return quantityOrdered;
    }
    public void setQuantityOrdered(int quantityOrdered) {
        this.quantityOrdered = quantityOrdered;
    }
}

```

### ProductRepository.java

```

package com.example.productordermanagementsystem.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.productordermanagementsystem.entity.Product;
public interface ProductRepository extends JpaRepository<Product, Long>{
}

```

### OrderRepository.java

```

package com.example.productordermanagementsystem.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.productordermanagementsystem.entity.Order;
public interface OrderRepository extends JpaRepository<Order, Long>{
}

```

### ProductService.java

```

package com.example.productordermanagementsystem.service;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.productordermanagementsystem.entity.Product;
import com.example.productordermanagementsystem.repository.ProductRepository;
@Service
public class ProductService {

    @Autowired
    private ProductRepository productRepo;

    public Product addProduct(Product product) {
        return productRepo.save(product);
    }

    public List<Product> getAllProducts(){
        return productRepo.findAll();
    }

    public Product updateStock(Long productId, int qty) {
        Optional<Product> existingProduct = productRepo.findById(productId);
    }
}

```

```

        if (existingProduct.isPresent()) {
            Product product = existingProduct.get();
            product.setAvailableQuantity(qty);
            return productRepo.save(product);
        } else {
            throw new RuntimeException("Product not found with ID: " + productId);
        }
    }
}

```

## OrderService.java

```

package com.example.productordermanagementsystem.service;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.productordermanagementsystem.entity.Product;
import com.example.productordermanagementsystem.repository.ProductRepository;
@Service
public class ProductService {

    @Autowired
    private ProductRepository productRepo;

    public Product addProduct(Product product) {
        return productRepo.save(product);
    }

    public List<Product> getAllProducts(){
        return productRepo.findAll();
    }

    public Product updateStock(Long productId, int qty) {
        Optional<Product> existingProduct = productRepo.findById(productId);
        if (existingProduct.isPresent()) {
            Product product = existingProduct.get();
            product.setAvailableQuantity(qty);
            return productRepo.save(product);
        } else {
            throw new RuntimeException("Product not found with ID: " + productId);
        }
    }
}

```

## ProductOrderController.java

```
package com.example.productordermanagementsystem.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.example.productordermanagementsystem.entity.Order;
import com.example.productordermanagementsystem.entity.Product;
import com.example.productordermanagementsystem.service.OrderService;
import com.example.productordermanagementsystem.service.ProductService;
@RestController
@RequestMapping("/api")
public class ProductOrderController {

    @Autowired
    private ProductService productService;

    @Autowired
    private OrderService orderService;

    // PRODUCT
    @PostMapping("/products")
    public Product createProduct(@RequestBody Product product) {
        return productService.addProduct(product);
    }

    @GetMapping("/products")
    public List<Product> getAllProducts(){
        return productService.getAllProducts();
    }

    @PutMapping("/products/{id}")
    public Product updateStock(@PathVariable Long id, @RequestParam int
availableQuantity) {
        return productService.updateStock(id, availableQuantity);
    }

    // ORDER
    @GetMapping("/orders")
    public List<Order> getAllOrders(){
        return orderService.getAllOrders();
    }

    @PostMapping("/orders")
    public Order orderProduct(@RequestParam Long productId, @RequestParam int
quantityOrdered){
        return orderService.placeOrder(productId ,quantityOrdered);
    }
}
```