PaymentServiceTest_Part2

Overview

This document details additional unit test cases for PaymentService in our e-commerce platform. It covers alternative validation logic, payment lookup, and deletion scenarios.

Dependencies

- JUnit 5
- Mockito
- Spring's ResponseEntity
- Java time/date libraries

Mocked Components

- OrderRepository
- PaymentRepository
- UserRepository
- ProductRepository
- CartRepository
- InvoiceService

Test Cases

1. testGetBankInformation_InvalidCardNumber

Purpose: Verify rejection of an invalid card number format.

Test Scenario

- Arrange: Pass a card number with fewer than 16 digits.
- Act: Call getBankInformation with the short card number.
- Assert: Expect HTTP 400 with 'Invalid card number' message.

Business Logic Verified

Ensures only 16-digit card numbers are accepted for validation.

2. testGetBankInformation_ExpiredCard

Purpose: Ensure expired cards are rejected during bank info validation.

Test Scenario

- Arrange: Provide a past date as expiry.
- Act: Call getBankInformation with expired date.
- Assert: Response should be 400 with message about expiration.

Business Logic Verified

Defensive check to block expired cards from proceeding.

3. testDeletePayment_Successful

Purpose: Ensure deletion of an existing payment works as intended.

Test Scenario

- Arrange: Mock paymentRepository to return true for existsById.
- Act: Call deletePayment with valid paymentId.
- Assert: Expect 200 status and deletion call verification.

Business Logic Verified

Allows admins to remove stale or refunded payment entries.

Conclusion

These additional tests help ensure PaymentService robustly validates payment inputs, handles deletions, and properly retrieves transactional records.