InvoiceServiceEmailTest Documentation

Overview

This document details the unit tests implemented for the InvoiceService class in our e-commerce application backend. The InvoiceService is responsible for generating and emailing PDF invoices to customers after a purchase is completed.

Test Class Structure

Dependencies

- JUnit 5: For test execution and assertions
- Mockito: To mock dependencies and simulate interactions
- Spring Boot Test: For integration with the Spring testing framework

Mocked Components

- JavaMailSender: Used to simulate email sending
- OrderRepository: Database access for order records
- UserRepository: Database access for user records
- ProductRepository: Fetches ordered products
- PdfInvoiceBuilder: Utility for generating PDF invoice byte stream

Test Setup

Before each test, the following setup is performed:
1. Initialize mocks using MockitoAnnotations
2. Create a test order with sample data
3. Create a test user with associated details
4. Create a test product representing a purchased item

Test Cases

1. testEmailPdfInvoice_EmailStructure

Purpose: Verify that the email message is prepared and sent correctly with the PDF.

Test Scenario:

- Arrange:
  Mocks return valid order, user, and product. PDF builder returns byte array.

- Act:
  Call invoiceService.emailPdfInvoice.

- Assert:
  Verify mailSender.send() is invoked with appropriate parameters.

Business Logic Verified:

Email content must be constructed and delivered properly.

2. testInvoiceSentDateIsSaved

Purpose: Ensure that the invoice sent date is recorded after sending the email.

Test Scenario:

- Arrange:
  Mocks return all valid data, PDF builder returns sample data.

- Act:
  Call invoiceService.emailPdfInvoice.

- Assert:
  Assert testOrder.getInvoiceSentDate() is not null after execution.

Business Logic Verified:

Invoice records should track when the invoice was issued.

Mocking Strategy

The tests use a consistent mocking strategy to isolate the InvoiceService from its dependencies:
1. Mock Responses: Return prepared test objects when repository methods are called
2. Behavior Verification: Verify that the service calls dependent methods as expected
3. State Verification: Validate the responses and outcomes from the service

Test Coverage

These tests cover the core functionality of the InvoiceService:
- Successful generation and sending of invoices
- Validation of order and user existence
- PDF generation failure handling
- Invoice sent date persistence
- Proper structure and dispatch of email

Conclusion

The InvoiceService test classes verify critical aspects of the invoice generation and dispatch process. This ensures reliable communication with customers and auditability of financial transactions in the application backend.