OrderHistoryServiceTest_Part3

Overview

This document describes additional test cases for OrderHistoryService. These cases expand coverage of features like history lookup, filtering, and edge cases.

Dependencies

- JUnit 5
- Mockito
- Spring's ResponseEntity

Mocked Components

- OrderHistoryRepository
- OrderRepository
- CartRepository
- ProductRepository

Test Cases

1. testViewActiveOrdersByUser_MixedStatuses

Purpose: Verify only 'Processing' and 'In-transit' orders are returned.

Test Scenario

- Arrange: Mock orderRepository to return a list with various status values.
- Act: Call viewActiveOrdersByUser.
- Assert: Response includes only orders with matching statuses.

Business Logic Verified

Filters orders based on business rules for active order states.

2. testGetProductsFromPreviousOrders_RefundedOrdersExcluded

Purpose: Ensure refunded orders are excluded from returned products.

Test Scenario

- Arrange: Mock orderRepository to return some 'Refunded' and some valid orders.
- Act: Call getProductsFromPreviousOrders.
- Assert: Only products from non-refunded orders are included in the response.

Business Logic Verified

Preserves correct product history by ignoring refunded items.

3. testRemoveOrderFromHistory_Success

Purpose: Verify that an order ID can be successfully removed from history.

Test Scenario

- Arrange: Mock orderHistoryRepository to return a history containing the target orderId.
- Act: Call removeOrderFromHistory.
- Assert: Response is 200 and order is removed from list.

Business Logic Verified

Supports removal of outdated or incorrect order entries.

Conclusion

These test cases ensure robust handling of order history features such as conditional retrieval, first-time use, cleanup, and filtering of order data.