

Refund Endpoint Throws Exception Instead of Returning HTTP 500

System Impact Summary

Uncaught RuntimeException in refund endpoint leads to test errors and unstructured 500 responses, causing inconsistent API behavior.

Description

When `orderService.requestRefundSingle(...)` throws a `RuntimeException`, the `OrderController.requestRefund(...)` method propagates the exception, resulting in a servlet error instead of an HTTP 500 response.

Steps to Reproduce

1. Mock `orderService.requestRefundSingle(...)` to throw new `RuntimeException("boom")` in `RefundTest`.
2. Perform `MockMvc.perform(put("/api/order/requestRefund/{orderId}", id))`.

Expected Behavior

The controller should catch the exception and return an HTTP **500 Internal Server Error** status.

Actual Behavior

The exception bubbles up, causing a `ServletException` and test error:

```
jakarta.servlet.ServletException: Request processing failed:  
java.lang.RuntimeException: boom
```

Severity

Medium – Causes unstable API behavior and test suite failures.

Affected Component: `OrderController.requestRefund(...)` endpoint

Root Cause: No exception handling in the controller or global advice to translate uncaught runtime exceptions into HTTP 500 responses.

Proposed Fix: Add a `ControllerAdvice` with `ExceptionHandler` to return HTTP 500 for uncaught exception.