GlobalExceptionHandler.java

package com.example.bookstoreapi.exceptions;

import org.springframework.http.HttpHeaders;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.ControllerAdvice;

import org.springframework.web.bind.annotation.ExceptionHandler;

import org.springframework.web.context.request.WebRequest;

import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;

import java.util.Date;

@ControllerAdvice

public class GlobalExceptionHandler extends ResponseEntityExceptionHandler {

@ExceptionHandler(ResourceNotFoundException.class)

public ResponseEntity<ErrorResponse> handleResourceNotFoundException(ResourceNotFoundException ex, WebRequest request) {

ErrorResponse errorResponse = new ErrorResponse(new Date(), ex.getMessage(), request.getDescription(false));

return new ResponseEntity<>(errorResponse, HttpStatus.NOT\_FOUND);

}

@ExceptionHandler(BookAlreadyExistsException.class)

public ResponseEntity<ErrorResponse> handleBookAlreadyExistsException(BookAlreadyExistsException ex, WebRequest request) {

ErrorResponse errorResponse = new ErrorResponse(new Date(), ex.getMessage(), request.getDescription(false));

return new ResponseEntity<>(errorResponse, HttpStatus.CONFLICT);

}

@ExceptionHandler(Exception.class)

public ResponseEntity<ErrorResponse> handleGlobalException(Exception ex, WebRequest request) {

ErrorResponse errorResponse = new ErrorResponse(new Date(), ex.getMessage(), request.getDescription(false));

return new ResponseEntity<>(errorResponse, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

}

class ErrorResponse {

private Date timestamp;

private String message;

private String details;

public ErrorResponse(Date timestamp, String message, String details) {

this.timestamp = timestamp;

this.message = message;

this.details = details;

}

// getters and setters

}

ResourceNotFoundException.java

package com.example.bookstoreapi.exceptions;

public class ResourceNotFoundException extends RuntimeException {

public ResourceNotFoundException(String message) {

super(message);

}

}

BookAlreadyExistsException.java

package com.example.bookstoreapi.exceptions;

public class BookAlreadyExistsException extends RuntimeException {

public BookAlreadyExistsException(String message) {

super(message);

}

}

Here's what's new:

Global Exception Handler: The GlobalExceptionHandler class is created using @ControllerAdvice to handle exceptions globally for the bookstore RESTful services.

Exception Handling Methods: Methods are defined to handle various exceptions, such as ResourceNotFoundException, BookAlreadyExistsException, and a generic Exception. Each method returns a ResponseEntity with an ErrorResponse object and an appropriate HTTP status code.