**Exercise 9: Online Bookstore - Understanding HATEOAS**

**BookController**

package com.bookstore.api.controllers;

import com.bookstore.api.dtos.BookDTO;

import com.bookstore.api.entities.Book;

import com.bookstore.api.mappers.BookMapper;

import com.bookstore.api.services.BookService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.hateoas.EntityModel;

import org.springframework.hateoas.server.mvc.WebMvcLinkBuilder;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import javax.validation.Valid;

@RestController

@RequestMapping("/books")

public class BookController {

private final BookService bookService;

private final BookMapper bookMapper;

@Autowired

public BookController(BookService bookService, BookMapper bookMapper) {

this.bookService = bookService;

this.bookMapper = bookMapper;

}

@PostMapping

public ResponseEntity<EntityModel<BookDTO>> createBook(@RequestBody @Valid BookDTO bookDTO) {

Book book = bookMapper.bookDTOToBook(bookDTO);

book = bookService.createBook(book);

EntityModel<BookDTO> entityModel = EntityModel.of(bookMapper.bookToBookDTO(book));

entityModel.add(WebMvcLinkBuilder.linkTo(WebMvcLinkBuilder.methodOn(BookController.class).getBookById(book.getId())).withSelfRel());

return new ResponseEntity<>(entityModel, HttpStatus.CREATED);

}

@GetMapping

public ResponseEntity<List<EntityModel<BookDTO>>> getAllBooks() {

List<Book> books = bookService.getAllBooks();

List<EntityModel<BookDTO>> entityModels = books.stream()

.map(book -> EntityModel.of(bookMapper.bookToBookDTO(book)))

.peek(entityModel -> entityModel.add(WebMvcLinkBuilder.linkTo(WebMvcLinkBuilder.methodOn(BookController.class).getBookById(entityModel.getContent().getId())).withSelfRel()))

.toList();

return new ResponseEntity<>(entityModels, HttpStatus.OK);

}

@GetMapping("/{id}")

public ResponseEntity<EntityModel<BookDTO>> getBookById(@PathVariable Long id) {

Book book = bookService.getBookById(id);

EntityModel<BookDTO> entityModel = EntityModel.of(bookMapper.bookToBookDTO(book));

entityModel.add(WebMvcLinkBuilder.linkTo(WebMvcLinkBuilder.methodOn(BookController.class).getBookById(id)).withSelfRel());

return new ResponseEntity<>(entityModel, HttpStatus.OK);

}

// ...

}

In the above code, i have enhanced the REST API to follow HATEOAS principles by adding links to resources in the API responses.

- I have used Spring HATEOAS to add links to resources.

- I have built and consumed hypermedia-driven APIs.

Note that i have used EntityModel to wrap the response data and add links to it. We have also used WebMvcLinkBuilder to create links to resources.