**Exercise 7: Online Bookstore - Introduction to Data Transfer Objects (DTOs)**

import org.modelmapper.ModelMapper;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import com.fasterxml.jackson.annotation.JsonProperty;

import com.fasterxml.jackson.annotation.JsonFormat;

import com.fasterxml.jackson.annotation.JsonIgnore;

import java.util.ArrayList;

import java.util.List;

// Entity classes

class Book {

private int id;

private String title;

private String author;

private double price;

private String isbn;

// Constructors, getters, and setters

public Book() {}

public Book(int id, String title, String author, double price, String isbn) {

this.id = id;

this.title = title;

this.author = author;

this.price = price;

this.isbn = isbn;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public String getIsbn() {

return isbn;

}

public void setIsbn(String isbn) {

this.isbn = isbn;

}

}

class Customer {

private int id;

private String name;

@JsonIgnore

private String password;

private String email;

@JsonFormat(pattern = "dd-MM-yyyy")

private String registrationDate;

// Constructors, getters, and setters

public Customer() {}

public Customer(int id, String name, String password, String email, String registrationDate) {

this.id = id;

this.name = name;

this.password = password;

this.email = email;

this.registrationDate = registrationDate;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getRegistrationDate() {

return registrationDate;

}

public void setRegistrationDate(String registrationDate) {

this.registrationDate = registrationDate;

}

}

// DTO classes

class BookDTO {

private int id;

private String title;

private String author;

@JsonProperty("cost")

private double price;

// Constructors, getters, and setters

public BookDTO() {}

public BookDTO(int id, String title, String author, double price) {

this.id = id;

this.title = title;

this.author = author;

this.price = price;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

}

class CustomerDTO {

private int id;

private String name;

private String email;

@JsonProperty("joined\_on")

@JsonFormat(pattern = "dd-MM-yyyy")

private String registrationDate;

// Constructors, getters, and setters

public CustomerDTO() {}

public CustomerDTO(int id, String name, String email, String registrationDate) {

this.id = id;

this.name = name;

this.email = email;

this.registrationDate = registrationDate;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getRegistrationDate() {

return registrationDate;

}

public void setRegistrationDate(String registrationDate) {

this.registrationDate = registrationDate;

}

}

// Controller using DTOs

@RestController

@RequestMapping("/api")

class BookstoreController {

private List<Book> books = new ArrayList<>();

private List<Customer> customers = new ArrayList<>();

private ModelMapper modelMapper = new ModelMapper();

// Initialize with some books and customers

public BookstoreController() {

books.add(new Book(1, "Java Programming", "John Smith", 29.99, "1234567890"));

books.add(new Book(2, "Python Essentials", "Jane Doe", 39.99, "0987654321"));

customers.add(new Customer(1, "Alice", "password123", "alice@example.com", "01-01-2020"));

customers.add(new Customer(2, "Bob", "password456", "bob@example.com", "15-03-2021"));

}

// GET books as DTOs

@GetMapping("/books")

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

List<BookDTO> bookDTOs = new ArrayList<>();

for (Book book : books) {

bookDTOs.add(modelMapper.map(book, BookDTO.class));

}

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

}

// GET customers as DTOs

@GetMapping("/customers")

public ResponseEntity<List<CustomerDTO>> getAllCustomers() {

List<CustomerDTO> customerDTOs = new ArrayList<>();

for (Customer customer : customers) {

customerDTOs.add(modelMapper.map(customer, CustomerDTO.class));

}

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

}

}