BookDTO Class

package com.example.bookstoreapi.dto;

import lombok.Data;

@Data

public class BookDTO {

private Long id;

private String title;

private String author;

private Double price;

private String isbn;

}

CustomerDTO Class

package com.example.bookstoreapi.dto;

import lombok.Data;

@Data

public class CustomerDTO {

private Long id;

private String name;

private String email;

private String address;

}

pom.xml

<dependency>

<groupId>org.mapstruct</groupId>

<artifactId>mapstruct</artifactId>

<version>1.5.3.Final</version>

</dependency>

<dependency>

<groupId>org.mapstruct</groupId>

<artifactId>mapstruct-processor</artifactId>

<version>1.5.3.Final</version>

<scope>provided</scope>

</dependency>

package com.example.bookstoreapi.mapper;

import com.example.bookstoreapi.dto.BookDTO;

import com.example.bookstoreapi.model.Book;

import org.mapstruct.Mapper;

@Mapper(componentModel = "spring")

public interface BookMapper {

BookDTO toBookDTO(Book book);

Book toBook(BookDTO bookDTO);

}

package com.example.bookstoreapi.mapper;

import com.example.bookstoreapi.dto.CustomerDTO;

import com.example.bookstoreapi.model.Customer;

import org.mapstruct.Mapper;

@Mapper(componentModel = "spring")

public interface CustomerMapper {

CustomerDTO toCustomerDTO(Customer customer);

Customer toCustomer(CustomerDTO customerDTO);

}

package com.example.bookstoreapi.dto;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.Data;

@Data

public class BookDTO {

private Long id;

private String title;

private String author;

@JsonProperty("book\_price")

private Double price;

private String isbn;

}

package com.example.bookstoreapi.dto;

import com.fasterxml.jackson.databind.annotation.JsonDeserialize;

import lombok.Data;

@Data

public class BookDTO {

private Long id;

private String title;

private String author;

@JsonDeserialize(using = CustomPriceDeserializer.class)

private Double price;

private String isbn;

}

BookController Example

package com.example.bookstoreapi.controller;

import com.example.bookstoreapi.dto.BookDTO;

import com.example.bookstoreapi.mapper.BookMapper;

import com.example.bookstoreapi.model.Book;

import com.example.bookstoreapi.service.BookService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import java.util.List;

import java.util.stream.Collectors;

@RestController

@RequestMapping("/books")

public class BookController {

@Autowired

private BookService bookService;

@Autowired

private BookMapper bookMapper;

@GetMapping

public List<BookDTO> getAllBooks() {

return bookService.getAllBooks().stream()

.map(bookMapper::toBookDTO)

.collect(Collectors.toList());

}

@PostMapping

public ResponseEntity<BookDTO> addBook(@RequestBody BookDTO bookDTO) {

Book book = bookMapper.toBook(bookDTO);

Book savedBook = bookService.addBook(book);

return new ResponseEntity<>(bookMapper.toBookDTO(savedBook), HttpStatus.CREATED);

}

// Other CRUD operations using DTOs...

}