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

**BookDTO**

package com.bookstore.api.dtos;

import com.fasterxml.jackson.annotation.JsonProperty;

public class BookDTO {

private Long id;

@JsonProperty("title")

private String title;

@JsonProperty("author")

private String author;

@JsonProperty("price")

private Double price;

@JsonProperty("isbn")

private String isbn;

// Getters and setters

}

**CustomerDTO**

package com.bookstore.api.dtos;

import com.fasterxml.jackson.annotation.JsonProperty;

public class CustomerDTO {

private Long id;

@JsonProperty("name")

private String name;

@JsonProperty("email")

private String email;

@JsonProperty("password")

private String password;

// Getters and setters

}

**BookMapper**

package com.bookstore.api.mappers;

import org.mapstruct.Mapper;

import org.mapstruct.Mapping;

import org.mapstruct.factory.Mappers;

import com.bookstore.api.entities.Book;

import com.bookstore.api.dtos.BookDTO;

@Mapper

public interface BookMapper {

BookMapper INSTANCE = Mappers.getMapper(BookMapper.class);

@Mapping(source = "id", target = "id")

@Mapping(source = "title", target = "title")

@Mapping(source = "author", target = "author")

@Mapping(source = "price", target = "price")

@Mapping(source = "isbn", target = "isbn")

BookDTO bookToBookDTO(Book book);

@Mapping(source = "id", target = "id")

@Mapping(source = "title", target = "title")

@Mapping(source = "author", target = "author")

@Mapping(source = "price", target = "price")

@Mapping(source = "isbn", target = "isbn")

Book bookDTOToBook(BookDTO bookDTO);

}

**CustomerMapper**

package com.bookstore.api.mappers;

import org.mapstruct.Mapper;

import org.mapstruct.Mapping;

import org.mapstruct.factory.Mappers;

import com.bookstore.api.entities.Customer;

import com.bookstore.api.dtos.CustomerDTO;

@Mapper

public interface CustomerMapper {

CustomerMapper INSTANCE = Mappers.getMapper(CustomerMapper.class);

@Mapping(source = "id", target = "id")

@Mapping(source = "name", target = "name")

@Mapping(source = "email", target = "email")

@Mapping(source = "password", target = "password")

CustomerDTO customerToCustomerDTO(Customer customer);

@Mapping(source = "id", target = "id")

@Mapping(source = "name", target = "name")

@Mapping(source = "email", target = "email")

@Mapping(source = "password", target = "password")

Customer customerDTOToCustomer(CustomerDTO customerDTO);

}

In the above code, I have defined DTOs BookDTO and CustomerDTO to transfer data between the client and server.

- I have used Jackson annotations to customize JSON serialization and deserialization.

- I have used MapStruct to map entities to DTOs and vice versa.

Note that I have used @JsonProperty to customize the JSON property names, and @Mapping to customize the mapping between entities and DTOs.