**Exercise 4: Online Bookstore - Processing Request Body and Form Data**

**Business Scenario:**

Create endpoints to accept and process JSON request bodies and form data for customer registrations.

**Instructions:**

1. **Request Body:**
   * Implement a POST endpoint to create a new customer by accepting a JSON request body.
2. **Form Data:**
   * Implement an endpoint to process form data for customer registrations.

**Solution**

**BookstoreAPIApplication.java**

package com.example.bookstoreapi;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class BookstoreAPIApplication {

public static void main(String[] args) {

SpringApplication.run(BookstoreAPIApplication.class, args);

}

}

**Customer.java**

package com.example.bookstoreapi.model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Customer {

private Long id;

private String name;

private String email;

private String address;

}

**CustomerController.java**

package com.example.bookstoreapi.controller;

import com.example.bookstoreapi.model.Customer;

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

import java.util.ArrayList;

import java.util.List;

@RestController

@RequestMapping("/api/customers")

public class CustomerController {

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

// POST: Create a new customer (accepts JSON request body)

@PostMapping

public Customer createCustomer(@RequestBody Customer customer) {

customers.add(customer);

return customer;

}

// POST: Process form data for customer registration

@PostMapping("/register")

public Customer registerCustomer(@RequestParam String name,

@RequestParam String email,

@RequestParam String address) {

Customer customer = new Customer((long) (customers.size() + 1), name, email, address);

customers.add(customer);

return customer;

}

}