ASSIGNMENT 5: RESTFUL WEB SERVICE IMPLEMENTATION + DOCKER

Project Name :- custProject

Participant :- Nikhil Bhosale

Project Explanation

This code implements a Spring Boot REST API for managing customers and their orders. The code defines the following components:

- **Customer**: A class that represents a customer with the following attributes: id, name, email, phone, and orders.
- **Order**: A class that represents an order with the following attributes: id, description, product, quantity, price, and orderDate.
- **Database**: A class that reads data from a JSON file and loads it into memory.
- **CustomerService**: A service that provides methods for getting all customers, getting a customer by ID, getting a customer's orders, and getting a customer's order by ID.
- **CustomerController**: A REST controller that defines endpoints for retrieving customer and order data.

The CustomerService class is responsible for interacting with the Database class to retrieve customer and order data. The CustomerController class defines REST endpoints for retrieving customer and order data by calling the appropriate methods on the CustomerService class. The API exposes the following endpoints:

- **GET /customers**: Retrieves all customers.
- **GET /customers/{id}**: Retrieves a customer by ID.
- **GET /customers/{customerId}/orders**: Retrieves a customer's orders by customer ID.
- **GET /customers/{customerId}/orders/{orderId}**: Retrieves a customer's order by customer ID and order ID.

The API returns JSON responses for each endpoint.

Data Model

Customer class:

- id (int): a unique identifier for the customer
- name (String): the name of the customer
- email (String): the email address of the customer
- phone (String): the phone number of the customer
- orders (List<Order>): a list of Order objects representing the orders placed by the customer

The Customer class represents a customer in the system and contains various properties that describe the customer. The id property is an integer value that uniquely identifies the customer. The name property is a String that contains the customer's name. The email property is a String that contains the email address of the customer. The phone property is a String that contains the phone number of the customer. The orders property is a List of Order objects that represent the orders placed by the customer.

Order class:

- id (int): a unique identifier for the order
- description (String): a description of the order
- product (String): the product being ordered
- quantity (int): the quantity of the product being ordered
- price (double): the price of the product being ordered
- orderDate (Date): the date the order was placed

The Order class represents an order in the system and contains various properties that describe the order. The id property is an integer value that uniquely identifies the order. The description property is a String that contains a description of the order. The product property is a String that contains the name of the product being ordered. The quantity property is an integer that represents the quantity of the product being ordered. The price property is a double that represents the price of the product being ordered. The orderDate property is a Date object that represents the date the order was placed.

t