

Assignment: Spring Boot JDBC Template for Console Application

Objectives of this assignment

1. Understand how to use Spring Boot's JDBC Template to interact with a MySQL database.
2. Learn how to set up a Spring Boot project with the necessary dependencies and configurations.
3. Practice creating a DAO class that uses Spring Boot's JDBC Template to perform database operations.
4. Create a menu-driven console application that allows the user to interact with the database.
5. Implement CRUD (Create, Read, Update, Delete) operations on the database.
6. Learn how to handle errors and display helpful error messages to the user.
7. Implement input validation to ensure that the user's input is valid.
8. Create a working application that meets the requirements of the assignment.

Overall, this assignment provides a practical exercise in using Spring Boot's JDBC Template to perform database operations and building a console application to interact with the database.

In this assignment, you will create a simple console application using Spring Boot JDBC Template to perform CRUD (Create, Read, Update, Delete) operations on a MySQL database.

Requirements

Your application should have the following features:

1. A MySQL database with a table named "products" that has the following fields:
 - id (INT, primary key)
 - name (VARCHAR(50))
 - description (VARCHAR(200))
 - price (DECIMAL(10,2))
 - quantity (INT)

```
CREATE TABLE products (  
    id INT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    description TEXT NOT NULL,  
    price DECIMAL(10,2) NOT NULL,  
    quantity INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

add following data into above created table

```
INSERT INTO products (id, name, description, price, quantity) VALUES (1, 'iPhone 13 Pro',  
'6.1-inch Super Retina XDR display, A15 Bionic chip', 999.00, 100);
```

```
INSERT INTO products (id, name, description, price, quantity) VALUES (2, 'Samsung  
Galaxy S21 Ultra', '6.8-inch Dynamic AMOLED 2X display, Exynos 2100 chip', 1199.00,  
50);
```

```
INSERT INTO products (id, name, description, price, quantity) VALUES (3, 'Google Pixel 6  
Pro', '6.7-inch Fluid AMOLED display, Tensor chip', 1099.00, 25);
```

```

INSERT INTO products (id, name, description, price, quantity) VALUES (4, 'MacBook Pro',
'13-inch Retina display, Apple M1 chip', 1299.00, 20);
INSERT INTO products (id, name, description, price, quantity) VALUES (5, 'Microsoft
Surface Laptop 4', '13.5-inch PixelSense display, Intel Core i5 chip', 999.00, 15);
INSERT INTO products (id, name, description, price, quantity) VALUES (6, 'Dell XPS 13',
'13.4-inch InfinityEdge display, Intel Core i7 chip', 1199.00, 10);
INSERT INTO products (id, name, description, price, quantity) VALUES (7, 'Sony
PlayStation 5', 'AMD Zen 2 CPU, AMD RDNA 2 GPU', 499.00, 5);
INSERT INTO products (id, name, description, price, quantity) VALUES (8, 'Microsoft Xbox
Series X', 'AMD Zen 2 CPU, AMD RDNA 2 GPU', 499.00, 2);
INSERT INTO products (id, name, description, price, quantity) VALUES (9, 'Bose
QuietComfort 35 II', 'Noise-cancelling headphones with Alexa voice control', 299.00, 1);
INSERT INTO products (id, name, description, price, quantity) VALUES (10, 'Apple AirPods
Pro', 'Active noise cancellation, spatial audio, transparency mode', 249.00, 0);

```

2. A menu that allows the user to choose from the following options:
 1. Display all products
 2. Display a single product by id
 3. Add a new product
 4. Update an existing product
 5. Delete a product
 6. Exit the program
3. Use Spring Boot's JDBC Template to perform all database operations.

Instructions

1. Create a new Spring Boot project using Spring Initializr or your preferred IDE.
2. Add the following dependencies to your pom.xml file:
 - **spring-jdbc**
 - **mysql-connector-java**
3. Configure your MySQL database connection in **application.properties**.
4. Create a model class for the "products" table.
5. Create a DAO (Data Access Object) class that uses Spring Boot's JDBC Template to perform database operations.
6. Create a menu-driven console application that allows the user to choose from the following options:
 1. Display all products
 2. Display a single product by id
 3. Add a new product
 4. Update an existing product
 5. Delete a product
 6. Exit the program
7. Test your console application by running it and interacting with the menu.

Bonus

If you finish the requirements above, try implementing the following bonus features:

- Input validation: validate user input and display helpful error messages if the input is invalid.
- Search functionality: add a search option to the menu that allows the user to search for products by name or description.
- Error handling: implement error handling for database errors and display helpful error messages to the user.

Below is detailed information about this assignment

In this assignment, you are tasked with creating a console application that allows the user to perform CRUD (Create, Read, Update, Delete) operations on a MySQL database using Spring Boot's JDBC Template.

First, you'll need to set up a MySQL database with a table named "products" that has the fields "id", "name", "description", "price", and "quantity". You can use a tool like MySQL Workbench to create the database and table, or create it using the MySQL command line. Next, you'll need to create a Spring Boot project with the necessary dependencies. You'll need to include the "spring-jdbc" and "mysql-connector-java" dependencies in your project's pom.xml file. You'll also need to configure your MySQL database connection in the application.properties file.

After setting up the project, you'll need to create a model class for the "products" table. This class should have fields that correspond to the fields in the table.

Next, you'll need to create a DAO (Data Access Object) class that uses Spring Boot's JDBC Template to perform database operations. You'll need to create methods for retrieving all products, retrieving a single product by id, adding a new product, updating an existing product, and deleting a product.

Once you've created the DAO class, you'll need to create a menu-driven console application that allows the user to interact with the database. The menu should display the following options:

1. Display all products
2. Display a single product by id
3. Add a new product
4. Update an existing product
5. Delete a product
6. Exit the program

The application should accept user input, perform the corresponding database operation using the DAO class, and display the results to the user.

For the bonus features, you can implement input validation to ensure that the user's input is valid, add search functionality to the menu to allow the user to search for products by name or description, and handle database errors and display helpful error messages to the user.

After completing the assignment, you should zip your project and email it to your instructor along with a brief summary of the features you implemented and any challenges you faced during development.

here are the Java classes for the Spring Boot JDBC Template console application assignment, without the business logic:

```

public class Product {
    private int id;
    private String name;
    private String description;
    private BigDecimal price;
    private int quantity;

    public Product() {}

    public Product(int id, String name, String description, BigDecimal price, int quantity) {
        this.id = id;
        this.name = name;
        this.description = description;
        this.price = price;
        this.quantity = quantity;
    }

    // Getters and setters for fields
}

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

@Repository
public class ProductDAO {
    @Autowired
    private JdbcTemplate jdbcTemplate;

    // Method to retrieve all products from the database
    public List<Product> getAllProducts() {
        // TODO: Implement method
        return null;
    }

    // Method to retrieve a single product by id from the database
    public Product getProductById(int id) {
        // TODO: Implement method
        return null;
    }

    // Method to add a new product to the database
    public void addProduct(Product product) {
        // TODO: Implement method
    }

    // Method to update an existing product in the database

```

```

    public void updateProduct(Product product) {
        // TODO: Implement method
    }

    // Method to delete a product from the database
    public void deleteProduct(int id) {
        // TODO: Implement method
    }
}

import java.math.BigDecimal;
import java.util.Scanner;

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

@Component
public class Menu {
    @Autowired
    private ProductDAO productDAO;

    private Scanner scanner;

    public Menu() {
        scanner = new Scanner(System.in);
    }

    public void displayMenu() {
        System.out.println("1. Display all products");
        System.out.println("2. Display a single product by id");
        System.out.println("3. Add a new product");
        System.out.println("4. Update an existing product");
        System.out.println("5. Delete a product");
        System.out.println("6. Exit");

        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                displayAllProducts();
                break;
            case 2:
                displayProductById();
                break;
            case 3:
                addProduct();
                break;
            case 4:
                updateProduct();

```

```

        break;
    case 5:
        deleteProduct();
        break;
    case 6:
        System.exit(0);
        break;
    default:
        System.out.println("Invalid choice");
        break;
    }

    displayMenu();
}

private void displayAllProducts() {
    // TODO: Implement method
}

private void displayProductById() {
    // TODO: Implement method
}

private void addProduct() {
    // TODO: Implement method
}

private void updateProduct() {
    // TODO: Implement method
}

private void deleteProduct() {
    // TODO: Implement method
}
}

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ConfigurableApplicationContext;

@SpringBootApplication
public class Main {
    public static void main(String[] args) {
        ConfigurableApplicationContext context = SpringApplication.run(Main.class, args);

        Menu menu = context.getBean(Menu.class);
        menu.displayMenu();
    }
}

```

