

Spring Boot CURD Thymeleaf Web + Data JPA + MySQL + Lombok

pom.xml:-

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.46</version><!--$NO-MVN-MAN-VER$-->
</dependency>

<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
```

application.properties:-

```
server.port=9898
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/boot
spring.datasource.username=root
spring.datasource.password=root

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL55Dialect
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
```

ProductReg.html:-

```
<html xmlns:th="https://www.thymeleaf.org/">
<body>
<h3> WELCOME TO PRODUCT REGISTER PAGE</h3>
<form th:action="@{/product/save}" method="POST" th:object="${product}">
<pre>
CODE : <input type="text" th:field="*{prodCode}"/>
COST : <input type="text" th:field="*{prodCost}"/>
  <input type="submit" value="Create"/>
</pre>
</form>
<span th:text="${message}"></span>
</body>
</html>
```

ProductsData.html:-

```
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h3>WELCOME TO PRODUCTS DATA PAGE</h3>
<table border="1">
  <tr>
    <th>ID</th>
    <th>CODE</th>
    <th>COST</th>
    <th>GST</th>
    <th>DISCOUNT</th>
    <th colspan="3">OPERATION</th>
  </tr>
  <tr th:each="ob:${list}">
    <td th:text="${ob.prodId}"> </td>
    <td th:text="${ob.prodCode}"> </td>
    <td th:text="${ob.prodCost}"> </td>
    <td th:text="${ob.gst}"> </td>
    <td th:text="${ob.discount}"> </td>
    <td>
      <a th:href="@{/product/delete/{id}{id=${ob.prodId}}}">DELETE</a>
    </td>
  </tr>
</table>
```

```
        <a th:href="@{/product/view/{id}{id=${ob.prodId}}}">VIEW</a>
    </td>
    <td>
        <a th:href="@{/product/edit/{id}{id=${ob.prodId}}}">EDIT</a>
    </td>
</tr>
</table>
<span th:text="${message}"></span>
</body>
</html>
```

ProductsOne.html:-

```
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<table border="1">
    <tr> <th>ID</th> <td th:text="${ob.prodId}"> </td> </tr>

    <tr> <th>CODE</th> <td th:text="${ob.prodCode}"> </td> </tr>

    <tr> <th>COST</th> <td th:text="${ob.prodCost}"> </td> </tr>

    <tr> <th>GST</th> <td th:text="${ob.gst}"> </td> </tr>

    <tr> <th>DISCOUNT</th> <td th:text="${ob.discount}"> </td> </tr>
</table>
</body>
</html>
```

ProductEdit.html:-

```
<html xmlns:th="https://www.thymeleaf.org/">
<body>
<h3> WELCOME TO PRODUCT EDIT PAGE</h3>
<form th:action="@{/product/update}" method="POST" th:object="${product}">
<pre>
ID : <input type="text" th:field="**{prodId}" readonly="readonly">
```

```
CODE : <input type="text" th:field="*{prodCode}"/>
COST : <input type="text" th:field="*{prodCost}"/>
<input type="submit" value="Update"/>
</pre>
</form>
</body>
</html>
```

Model class:-

```
package in.nit.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

import lombok.Data;

@Data
@Entity
public class Product {
    @Id
    @GeneratedValue
    private Integer prodId;
    private String prodCode;
    private Double prodCost;
    private Double gst;
    private Double discount;
}
```

Repository Interface:-

```
package in.nit.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import in.nit.model.Product;

public interface ProductRepository
    extends JpaRepository<Product, Integer>
{

}
```

IService:-

```
package in.nit.service;

import java.util.List;
import java.util.Optional;

import in.nit.model.Product;

public interface IProductService {

    List<Product> getAllProducts();
    void deleteProduct(Integer id);
    Optional<Product> getOneproduct(Integer id);

    Integer saveProduct(Product p);

}
```

ServiceImpl:-

```
package in.nit.service.impl;

import java.util.List;
import java.util.Optional;

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

import in.nit.model.Product;
import in.nit.repo.ProductRepository;
import in.nit.service.IProductService;

@Service
public class ProductServiceImpl
    implements IProductService
{
    @Autowired
    private ProductRepository repo; //HAS-A

    @Override
    public List<Product> getAllProducts() {
        List<Product> list=repo.findAll();
        return list;
    }
}
```

```
@Override
public void deleteProduct(Integer id) {
    repo.deleteByld(id);
}

@Override
public Optional<Product> getOneproduct(Integer id) {
    Optional<Product> opt=repo.findByld(id);
    return opt;
}

@Override
public Integer saveProduct(Product p) {
    double gst=p.getProdCost() * 8/100.0;
    double discount=p.getProdCost() * 12/100.0;
    p.setGst(gst);
    p.setDiscount(discount);
    Integer id=repo.save(p).getProdld();
    return id;
}
}
```

Controller:-

```
package in.nit.controller;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import in.nit.model.Product;
import in.nit.service.IProductService;

@Controller
@RequestMapping("/product")
public class ProductController {
    @Autowired
    private IProductService service;
```

```
/**
 * URL : /all , GET
 * showProducts() send data to UI
 * Page : ProductsData
 */
@RequestMapping("/all")
public String showProducts(Model model) {
    List<Product> list=service.getAllProducts();
    model.addAttribute("list", list);
    return "ProductsData";
}

/**
 * on click DELETE hyperLink this
 * method is called.
 * as: /delete/{id} , GET
 * deleteProduct
 */
@RequestMapping("/delete/{id}")
public String deleteProduct(
    @PathVariable("id")Integer id,
    Model model
)
{
    //deleting row
    service.deleteProduct(id);

    //fetching new data
    List<Product> list=service.getAllProducts();

    //send to ui
    model.addAttribute("list", list);
    model.addAttribute("message", id+" deleted");

    return "ProductsData";
    //return "redirect:../all";
}

/**
 * On click View HyperLink display
 * one row data at ProductOne HTML
 * Page, URL : /view/{id}
 * method: getOneProduct
 */
```

```
@RequestMapping("/view/{id}")
public String getOneProduct(
    @PathVariable("id") Integer id,
    Model model
)
{
    Optional<Product> opt=service.getOneproduct(id);
    if(opt.isPresent()) {
        model.addAttribute("ob", opt.get());
    }else {
        model.addAttribute("message", "NO DATA FOUND");
    }
    return "ProductOne";
}
```

```
/**
 * This method is used to display
 * Form with empty inputs
 * Page: ProductReg.html
 * URL : /reg, Type:GET
 * method: showRegPage()
 * Form Backing Obj: product
 */
@RequestMapping("/reg")
public String showRegPage(Model model) {
    model.addAttribute("product", new Product());
    return "ProductReg";
}
```

```
/**
 * On click Form submit
 * read ModelAttribute
 * call service layer save method
 * return ProduReg with success
 * message.
 * URL : /save, type:POST
 * method: saveProduct()
 */
@RequestMapping(value = "/save"
    ,method = RequestMethod.POST)
public String saveProduct(
```



```
        @ModelAttribute Product product,
        Model model)

    {
        Integer id=service.saveProduct(product);
        String message="Product saved with :"+id;
        model.addAttribute("message", message);
        //clear Form
        model.addAttribute("product", new Product());
        return "ProductReg";
    }

    /**
     * On click Edit HyperLink
     * load object data from DB
     * and display at UI using Model
     * URL : /edit/{id} , Type:GET
     * showEditPage() : ProductEdit.html
     */
    @RequestMapping("/edit/{id}")
    public String showEditPage(
        @PathVariable("id")Integer pid,
        Model model)
    {
        Optional<Product> ob=service.getOneproduct(pid);
        if(ob.isPresent()) {
            model.addAttribute("product", ob.get());
        }else {
            model.addAttribute("product", new Product());
        }

        return "ProductEdit";
    }

    /**
     * On click Update button,
     * Read Form Data using @ModelAttribute
     * call service layer save()
     * Return to UI : ProductData.html
     * updateProduct() , URL /update, POST
     * Also fetch new data
     */
    @RequestMapping(value = "/update",
        method = RequestMethod.POST)
    public String updateProduct(
        @ModelAttribute Product product,
```

```
        Model model = new Model();  
        {  
            Integer id=service.saveProduct(product);  
            model.addAttribute("message", "Product updated:"+id);  
  
            //fetch new data from DB  
            List<Product> list=service.getAllProducts();  
            model.addAttribute("list", list);  
            return "ProductsData";  
        }  
    }  
}
```

FB: <https://www.facebook.com/groups/thejavatemple/>