

## Source Code for UpdatingUserDetails:

### 1. MainController.java:

```
package com.simplilearn.controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import com.simplilearn.dao.EProductDao;
import com.simplilearn.entity.EProductEntity;
import org.apache.log4j.Logger;
@Controller
public class MainController {

    @Autowired
    EProductDao dao;

    private Logger logger = Logger.getLogger(MainController.class);

    @GetMapping("/productid")
    public String getProduct(HttpServletRequest request, ModelMap map, HttpServletResponse response) {
        long id=Long.parseLong(request.getParameter("id"));
        if(id>0)
        {
            EProductEntity entity=dao.getProductById(id);

            logger.info("This Is An Info Log Entry .....!");
            logger.info(entity);
            map.addAttribute("obj",entity);
            return "details";
        }
        else
```

```

    {
        try {
            try {
                request.getRequestDispatcher("/index.jsp").include(request, response);
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                out.println("<center><span style='color:red'>Invalid Credentials!!</span></center>");

            } catch (ServletException e) {
                e.printStackTrace();
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return null;
}

```

```

@GetMapping("/update")
public String updateProduct(HttpServletRequest request, ModelMap map) {
    long id=Long.parseLong(request.getParameter("id"));
    String name=request.getParameter("name");
    BigDecimal price = new BigDecimal(request.getParameter("price"));
    String date1 = request.getParameter("date");
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    Date date;
    try {
        date = formatter.parse(date1);
        EProductEntity entity= new EProductEntity();
        entity.setId(id);
        entity.setName(name);
        entity.setPrice(price);
        entity.setDateAdded(date);
        dao.updateProduct(entity);
        map.addAttribute("obj",entity);

    } catch (ParseException e) {
        e.printStackTrace();
    }
    return "details";
}

```

```

}

```

## 2. EProductDao.java:

```
package com.simplilearn.dao;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.ResultSetExtractor;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.stereotype.Repository;
import com.simplilearn.entity.EProductEntity;

@Repository
public class EProductDao {

    @Autowired
    JdbcTemplate jdbcTemplate;

    public List<EProductEntity> getAllProducts() {

        return jdbcTemplate.query("select * from eproduct", new
RowMapper<EProductEntity>() {

            public EProductEntity mapRow(ResultSet rs, int rowNum) throws
SQLException {

                EProductEntity obj = new EProductEntity();
                obj.setId(rs.getLong(1));
                obj.setName(rs.getString(2));
                obj.setPrice(rs.getBigDecimal(3));
                obj.setDateAdded(rs.getTimestamp(4));
                return obj;
            }
        });
    }

    public EProductEntity getProductById(long id) {

        return jdbcTemplate.query("select * from eproduct where id=" + id, new
ResultSetExtractor<EProductEntity>() {

            public EProductEntity extractData(ResultSet rs) throws SQLException, DataAccessException {
                if (rs.next()) {
                    EProductEntity obj = new EProductEntity();
```

```

        obj.setId(rs.getLong(1));
        obj.setName(rs.getString(2));
        obj.setPrice(rs.getBigDecimal(3));
        obj.setDateAdded(rs.getTimestamp(4));
        return obj;
    }

    return null;
}

});

}

public void updateProduct(EProductEntity obj) {
    if (obj.getId() > 0)
    {
        String sql = "update eproduct set name=?, price=?,date_added=? where id=?";
        jdbcTemplate.update(sql, obj.getName(), obj.getPrice(), obj.getDateAdded(), obj.getId());

    } else
    {
        String sql = "insert into eproduct (name,price,date_added) values (?,?,:)";
        jdbcTemplate.update(sql, obj.getName(), obj.getPrice(), obj.getDateAdded());
    }
}

}

```

### 3. EProductEntity.java:

```

package com.simplilearn.entity;
import java.math.BigDecimal;
import java.util.Date;
public class EProductEntity {

    private long id;
    private String name;
    private BigDecimal price;
    private Date dateAdded;

    public EProductEntity() {
    }

    public EProductEntity(String name, BigDecimal price, Date dateAdded) {
        super();
        this.name = name;
    }
}

```

```

        this.price = price;
        this.dateAdded = dateAdded;
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public BigDecimal getPrice() {
        return price;
    }

    public void setPrice(BigDecimal price) {
        this.price = price;
    }

    public Date getDateAdded() {
        return dateAdded;
    }

    public void setDateAdded(Date dateAdded) {
        this.dateAdded = dateAdded;
    }

    @Override
    public String toString() {
        return "EProductEntity [id=" + id + ", name=" + name + ", price=" + price + ", dateAdded=" +
dateAdded + "]\n";
    }
}

```

#### 4. Details.jsp:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

```

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h3>Products Details Page</h3>
<c:set var="product" value="{obj }"></c:set>
Id: ${product.id } <br>
Name: ${product.name } <br>
Price: ${product.price } <br>
Date: ${product.dateAdded } <br><br>
<a href="edit.jsp">Click here to edit</a> <br><br>
<a href="index.jsp">Click here go to home page</a>
</body>
</html>

```

## 5. test-servlet.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config></context:annotation-config>

<!-- jsp output -->
<context:component-scan base-package="com.simplilearn.controller"></context:component-scan>
<bean id="jspViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<property name="suffix" value=".jsp"></property>
<property name="prefix" value="/WEB-INF/view/"></property>
</bean>

<!-- database -->

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="com.mysql.cj.jdbc.Driver"></property>
<property name="url" value="jdbc:mysql://localhost:3306/ecommerce1"></property>
<property name="username" value="root"></property>
<property name="password" value="Vyshi@1432"></property>

</bean>

```

```

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="dataSource"></property>
</bean>

<bean id="dao" class="com.simplilearn.dao.EProductDao"></bean>

</beans>

```

## 6. web.xml:

```

<web-app>
    <display-name>Archetype Created Web Application</display-name>
    <servlet>
        <servlet-name>test</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>test</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/test-servlet.xml</param-value>
    </context-param>

    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>
</web-app>

```

## 7. Edit.jsp:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h2>Please Update Details !</h2>
<form action="update">
<input type="number" name="id" placeholder="ID"><br>

```

```
<input type="text" name="name" placeholder="Name"><br>
<input type="number" name="price" placeholder="Price"><br>
<input type="date" name="date" placeholder="Date"><br>
<input type="submit" value="Confirm">
</form>
</body>
</html>
```

## 8. Index.jsp:

```
<html>
<body>
<h2>Product Details Console !</h2>

    <form action="productid">
        <input type="number" name="id" placeholder="ID"> <input
            type="submit" value="Enter">
    </form>
</body>
</html>
```