

Source Code for Searching for a Specific User and Updating the User Information

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>Mphasis</groupId>
  <artifactId>SpringMVCExample1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>SpringMVCExample1 Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
    <spring.version>5.1.0.RELEASE</spring.version>
    <cglib.version>2.2.2</cglib.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/jstl/jstl -->
    <dependency>
      <groupId>jstl</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>
    <!-- Spring core & mvc -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.version}</version>
    </dependency>
```

```

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>${spring.version}</version>
        </dependency>
    </dependency>

        <groupId>org.springframework</groupId>
        <artifactId>spring-orm</artifactId>
        <version>${spring.version}</version>
        <type>jar</type>
        <scope>compile</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -
->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.28</version>
    </dependency>

    <dependency>
        <groupId>cglib</groupId>
        <artifactId>cglib-nodep</artifactId>
        <version>${cglib.version}</version>
        <scope>runtime</scope>
    </dependency>
</dependencies>

<build>
    <finalName>SpringMVCEXample1</finalName>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven
defaults (may be moved to parent pom) -->
    <plugins>
        <plugin>
            <artifactId>maven-clean-plugin</artifactId>
            <version>3.1.0</version>
        </plugin>
        <!-- see http://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin_bindings_for_war_packaging -->
        <plugin>
            <artifactId>maven-resources-plugin</artifactId>
            <version>3.0.2</version>
        </plugin>
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.0</version>
        </plugin>
        <plugin>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>2.22.1</version>
        </plugin>
        <plugin>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.2.2</version>
        </plugin>
        <plugin>
            <artifactId>maven-install-plugin</artifactId>
            <version>2.5.2</version>
        </plugin>
    </plugins>

```

```
<plugin>
  <artifactId>maven-deploy-plugin</artifactId>
  <version>2.8.2</version>
</plugin>
</plugins>
</pluginManagement>
</build>
</project>
```

Employee.java

```
package com.springcrud.beans;

public class Employee {
    private int eid;
    private String name, designation;
    private float salary;
    public int getEid() {
        return eid;
    }
    public void setEid(int eid) {
        this.eid = eid;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDesignation() {
        return designation;
    }
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    public float getSalary() {
        return salary;
    }
    public void setSalary(float salary) {
        this.salary = salary;
    }
}
```

EmployeeController.java

```
package com.springcrud.controller;

import java.util.List;

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

import com.springcrud.beans.Employee;
import com.springcrud.dao.EmployeeDao;

@Controller
public class EmployeeController {
    @Autowired
    EmployeeDao dao;
    @RequestMapping(value="/empform", method=RequestMethod.GET)
    public String showForm(Model m){
        m.addAttribute("emp", new Employee());
        //System.out.println("show Form");
        return "empform";
    }

    @RequestMapping(value="/save", method=RequestMethod.POST)
    public String save(@ModelAttribute("emp")Employee emp){
        //System.out.println(emp.getId());
        dao.insert(emp);
        return "redirect:/viewemp";
    }
    @RequestMapping("/viewemp")
    public String viewemp(Model m){
        List<Employee> empList= dao.getEmployeeDetails();
        /*for(Employee e : empList){
            System.out.println(e.getId());
        }*/
        m.addAttribute("empList",empList);
        return "viewemp";
    }

    /*@RequestMapping(value="/editemp/{eid}")
    public String edit(@PathVariable int eid, Model m){

        Employee emp=dao.getEmpId(eid);
        m.addAttribute("emp",emp);
        return "empeditform";
    }
    @RequestMapping(value="/editsave",method=RequestMethod.POST)
```

```

        public String editSave(@ModelAttribute("emp") Employee emp){

            System.out.println(emp.getId());
            dao.update(emp);
            return "redirect:/viewemp";
        }

        @RequestMapping(value="/deleteemp/{eid}")
        public String delete(@PathVariable int eid){
            dao.delete(eid);
            return "redirect:/viewemp";
        }

        @GetMapping("/searchEmployee")
        public String searchEmployee(@RequestParam("eid") int eid, Model model) {
            Employee employee = dao.getEmpId(eid);

            model.addAttribute("emp", employee);
            return "empeditform";
        }
        @PostMapping("/editsave")
        public String editSave(@ModelAttribute("emp") Employee emp) {
            dao.update(emp);
            return "redirect:/viewemp";
        }
        @GetMapping("/emp12")
        public String emp12() {
            // Code for handling the request for "/SpringMVCEXample1/emp12"
            return "emp12"; // The name of the JSP file to be displayed
        }
    }
}

```

EmployeeDao.java

```

package com.springcrud.dao;

import java.util.List;

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

import com.springcrud.beans.Employee;

public class EmployeeDao {

```

```

JdbcTemplate template;

    public void setTemplate(JdbcTemplate template) {
        this.template = template;
    }

    public int insert(Employee emp) {
        String sql="insert into Employee(eid, name, designation,salary)
values(?,?,?,?)";
        int
ans=template.update(sql,emp.getId(),emp.getName(),emp.getDesignation(),emp.getSal
ary());
        return ans;
    }

    public int update(Employee emp) {
        String sql="update Employee set  name=?, designation=?,salary=?
where eid=?";
        int
ans=template.update(sql,emp.getName(),emp.getDesignation(),emp.getSalary(),emp.get
Eid());
        return ans;
    }

    public int delete(int id) {
        String sql="delete from Employee where eid=?";
        int ans=template.update(sql,id);
        return ans;
    }

    public Employee getEmpId(int id){
        String sql="select * from Employee where eid=?";
        Employee emp=template.queryForObject(sql, new Object[] {id},new
EmployeeMapper() );

        return emp;
    }

    public List<Employee> getEmployeeDetails(){
        String sql="select * from Employee";
        List<Employee> emplist=template.query(sql, new EmployeeMapper());
        return emplist;
    }
}

```

EmployeeMapper.java

```
package com.springcrud.dao;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import com.springcrud.beans.Employee;

//RowMapper is pre defined interface to convert resultset records to object
public class EmployeeMapper implements RowMapper<Employee> {

    @Override
    public Employee mapRow(ResultSet rs, int rowNum) throws SQLException {
        // TODO Auto-generated method stub
        Employee employee=new Employee();
        employee.setEid(rs.getInt("eid"));
        employee.setName(rs.getString("name"));
        employee.setDesignation(rs.getString("designation"));
        employee.setSalary(rs.getFloat("salary"));

        return employee;
    }
}
```

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Employee Form</title>
</head>
<body>
    <h1>Employee Dashboard</h1>

    <a href="empform">Add Employee</a><br>
    <a href="emp12">Edit Employee</a><br>
    <a href="viewemp"> View Employee Details</a><br>
</body>
</html>
```

empeditform.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1> Edit Employee Information </h1>
<form:form method="post" action="${pageContext.request.contextPath}/editsave"
modelAttribute="emp">

id: <form:hidden path="eid"/> <br>
name: <form:input path="name" /> <br>
designation: <form:input path="designation"/> <br>
salary: <form:input path="salary"/> <br>
<input type="submit" value="Save changes">
</form:form>
</body>
</html>
```

empform.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ taglib uri="http://www.springframework.org/tags/form" prefix="fm"%>

<!DOCTYPE html>

<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Add New Employee</h1>
    <fm:form method="post" action="save" modelAttribute="emp">
        <table >
            <tr>
                <td>eid : </td>
                <td><fm:input path="eid" /></td>
            </tr>
            <tr>
                <td>Name : </td>
                <td><fm:input path="name" /></td>
            </tr>
```



```

        <tr>
            <td>Salary :</td>
            <td><fm:input path="salary" /></td>
        </tr>
        <tr>
            <td>Designation :</td>
            <td><fm:input path="designation" /></td>
        </tr>
        <tr>
            <td> </td>
            <td><input type="submit" value="Save" /></td>
        </tr>
    </table>
</fm:form>

</body>
</html>

```

viewemp.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" %>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<table border="2">
<tr>
<th>Id</th>
<th>Name</th>
<th>Designation</th>
<th>Salary</th>
</tr>
<c:forEach var="emp" items="${emplist}" >
<tr>

<td>${emp.eid}</td>
<td>${emp.name}</td>
<td>${emp.designation}</td>
<td>${emp.salary}</td>

</tr>
</c:forEach>
</table>

```

```
</body>
</html>
```

spring-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:p="http://www.springframework.org/schema/p"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-context-
3.0.xsd">

    <context:component-scan base-package="com.springcrud.controller" />
    <!-- Create Object of View Resolver -->
    <bean id="viewresolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>
    <!-- Create Object of Database Connection -->
    <bean id="ds"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">

        <property name="driverClassName" value="com.mysql.cj.jdbc.Driver"/>
        <property name="url" value="jdbc:mysql://localhost:3306/employee"/>
        <property name="username" value="root" />

        <property name="password" value="Satti@2001" />

    </bean>

    <!-- Create Object of JDBCTemplate -->
    <bean id="jt" class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="ds" />
    </bean>

    <!-- Create Object of Repository/CRUD -->

    <bean id="dao" class="com.springcrud.dao.EmployeeDao">
        <property name="template" ref="jt"/>
    </bean>
</beans>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                             http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
         version="4.0">

    <display-name>Spring Crud Operation</display-name>
    <welcome-file-list>

        <welcome-file>index.jsp</welcome-file>

    </welcome-file-list>

    <servlet>
        <!-- Provide a Servlet Name -->
        <servlet-name>spring</servlet-name>
        <!-- Provide a fully qualified path to the DispatcherServlet class -->
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <!-- Provide a Servlet Name that you want to map -->
        <servlet-name>spring</servlet-name>
        <!-- Provide a url pattern -->
        <url-pattern>/</url-pattern>
    </servlet-mapping>
</web-app>
```