Searching for a Specific User and Updating the User Information

Source code

**index.html**

<html>

<body>

<h2>User verification</h2>

<form action="retrieve">

Enter User ID:<input type="text" name="id">

<input type="submit" name="sretrirve" value="Retrieve data">

</form>

</body>

</html>

**spring.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"

xsi:schemaLocation = "http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<!--creation of student bean -->

<bean id="student" class="com.example.User"></bean>

<!-- creation of dao bean -->

<bean id="dao" class="com.example.UserDAO">

<property name="temp" ref="jdbctemp"></property>

</bean>

<!--creation of jdbctemplate -->

<bean id="jdbctemp" class="org.springframework.jdbc.core.JdbcTemplate">

<property name="dataSource" ref="ds"></property>

</bean>

<!--create of datasource bean -->

<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="com.mysql.jdbc.Driver"></property>

<property name="url" value="jdbc:mysql://localhost:3306/db5"></property>

<property name="username" value="root"></property>

<property name="password" value="123456"></property>

</bean>

</beans>

**Initializer.html**

package com.example;

import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

public class Initializer extends AbstractAnnotationConfigDispatcherServletInitializer {

@Override

protected Class<?>[] getRootConfigClasses() {

// TODO Auto-generated method stub

return null;

}

@Override

protected Class<?>[] getServletConfigClasses() {

// TODO Auto-generated method stub

return new Class[] {Config.class};

}

@Override

protected String[] getServletMappings() {

// TODO Auto-generated method stub

return new String[] {"/"};

}

}

**Config.java**

package com.example;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan(basePackages = {"com.example"})

public class Config {

}

**UserDAO.java**

package com.example;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import org.springframework.dao.DataAccessException;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.ResultSetExtractor;

public class UserDAO {

private JdbcTemplate temp;

public void setTemp(JdbcTemplate temp) {

this.temp = temp;

}

public List<User> getallstudents(String s2){

String sql="select \* from user where id="+s2;

return temp.query(sql,new ResultSetExtractor<List<User>>() {

public List<User> extractData(ResultSet rs) throws SQLException, DataAccessException {

ArrayList<User> al=new ArrayList<User>();

while(rs.next()) {

User u=new User();

u.setId(rs.getInt(1));

u.setName(rs.getString(2));

u.setEmail(rs.getString(3));

al.add(u);

}

return al;

}

});

}

public int insert(User u) {

String sql="insert into User values("+u.getId()+",'"+u.getName()+"','"+u.getEmail()+"')";

return temp.update(sql);

}

}

**User.java**

package com.example;

public class User {

private int id;

private String name;

private String email;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

**display.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@page import="java.util.List,java.util.Scanner,org.springframework.context.ApplicationContext,

org.springframework.context.support.ClassPathXmlApplicationContext" %>

<%@page import="com.example.User,com.example.UserDAO" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Display here</title>

</head>

<body>

<form action="update.jsp">

<input type="submit" name="edit" value="Update data"></input>

</form>

<%

String id=(String)request.getAttribute("ID");

ApplicationContext ac=new ClassPathXmlApplicationContext("spring.xml");

User s=ac.getBean(User.class);

UserDAO dao=ac.getBean(UserDAO.class);

List<User> ss=dao.getallstudents(id);

%>

<table>

<tr><th>id</th><th>name</th><th>email</th></tr>

<%for(User us:ss){%>

<tr><td><%=us.getId()%></td><td><%=us.getName()%></td><td><%=us.getEmail()%></td></tr>

<%} %>

</table>

</body>

</html>

**insert.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@page import="java.util.List,java.util.Scanner,org.springframework.context.ApplicationContext,

org.springframework.context.support.ClassPathXmlApplicationContext" %>

<%@page import="com.example.User,com.example.UserDAO" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Update here</title>

</head>

<body>

<%

ApplicationContext ac=new ClassPathXmlApplicationContext("spring.xml");

User u=ac.getBean(User.class);

UserDAO dao=ac.getBean(UserDAO.class);

Scanner sc=new Scanner(System.in);

u.setId(Integer.parseInt(request.getParameter("id")));

u.setName(request.getParameter("name"));

u.setEmail(request.getParameter("email"));

int row = dao.insert(u);

if(row>0){ %>

Update is successful

<%}else{%>

Update failed

<%} %>

</body>

</html>

**update.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Update here</title>

</head>

<body>

<h3>Enter the user details</h3>

<form action="add">

Enter User ID:<input type="text" name="id">

Enter User Name:<input type="text" name="name">

Enter User Email:<input type="text" name="email">

<input type="submit" name="insert" value="Insert data">

</form>

</body>

</html>

**pom.xml**

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>Maven</groupId>

<artifactId>SpecificUser</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpecificUser Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.18</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.18</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.18</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>SpecificUser</finalName>

</build>

</project>

**SQL Commands**

create database db5;

use db5;

create table User(id int primary key, name varchar(20),email varchar(20));

select \* from User;

Github Link:

https://github.com/ashwinbalaji44/Assignment.git