Searching for a Specific User and Updating the User Information

package com.example;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Student {

@Id

private int sid;

private String sname;

private String email;

public int getSid() {

return sid;

}

public void setSid(int sid) {

this.sid = sid;

}

public String getSname() {

return sname;

}

public void setSname(String sname) {

this.sname = sname;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

package com.example;

import java.util.List;

import org.springframework.orm.hibernate3.HibernateTemplate;

public class StudentDAO {

private HibernateTemplate temp;

public void setTemp(HibernateTemplate temp) {

this.temp = temp;

}

//Insert

public int insert(Student student) {

return (int) temp.save(student);

}

//update

public boolean updateStudentDetails(Student updatedStudent) {

int id = updatedStudent.getId();

if (studentMap.containsKey(id)) {

studentMap.put(id, updatedStudent);

System.out.println("Student details updated successfully for ID: " + id);

return true; // Update successful

} else {

System.out.println("Student with ID: " + id + " does not exist.");

return false; // Update failed

}

}

//retrieve

public List<Student> getall(){

return (List<Student>) temp.find("from Student");

}

}

package com.example;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

@Controller

public class StudentController {

@RequestMapping("/insert")

public ModelAndView insert(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

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

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

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

s.setSid(Integer.parseInt(request.getParameter("id")));

s.setSname(request.getParameter("name"));

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

if(dao.insert(s)>0) {

mv.setViewName("success.jsp");

}

return mv;

}

@RequestMapping("/update")

public ModelAndView update(HttpServletRequest request, HttpServletResponse response) {

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

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

int id = Integer.parseInt(request.getParameter("id"));

String name = request.getParameter("name");

String email = request.getParameter("email");

// Retrieve existing student

Student existingStudent = dao.getStudentById(id);

ModelAndView mv = new ModelAndView();

if (existingStudent != null) {

// Update details

existingStudent.setSname(name);

existingStudent.setEmail(email);

// Perform the update

boolean isUpdated = dao.updateStudentDetails(existingStudent);

if (isUpdated) {

mv.setViewName("success.jsp");

} else {

mv.setViewName("error.jsp");

}

} else {

// Handle if student with the given ID doesn't exist

mv.setViewName("error.jsp");

}

return mv;

}

@RequestMapping("/getall")

public ModelAndView getall(HttpServletRequest request,HttpServletResponse response) {

ModelAndView mv=new ModelAndView();

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

Student student=ac.getBean(Student.class);

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

List<Student> list=dao.getall();

mv.setViewName("displayall.jsp");

mv.addObject("list",list);

return mv;

}

}

package com.example;

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

public class DispatcherServlet 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[] {"/"};

}

}

package com.example;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

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

public class Config {

}

<**html**>

<**body**>

<**h2**>Student Records</**h2**>

<**form** action=*"insert"*>

SId<**input** type=*"text"* name=*"id"*></**br**>

SName<**input** type=*"text"* name=*"name"*></**br**>

SEmail<**input** type=*"text"* name=*"email"*></**br**>

<**input** type=*"submit"* value=*"insert"*></**br**>

</**form**>

</**body**>

</**html**>

<%@ **page** language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ **page** import=*"com.example.\*"* %>

<%@ **page** import=*"java.util.\*"* %>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"ISO-8859-1"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<%List<Student> list=(List<Student>)request.getAttribute("list"); %>

<**table** border=*"1"*>

<**tr**><**th**>Student Id</**th**><**th**>Student Name</**th**><**th**>Student Email</**th**></**tr**>

<%for(Student ss:list){ %>

<**tr**><**td**><%=ss.getSid() %></**td**><**td**><%=ss.getSname() %></**td**><**td**><%=ss.getEmail() %></**td**></**tr**>

<%} %>

</**table**>

</**body**>

</**html**>

<%@ **page** language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"ISO-8859-1"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**h1**>Student Details Added SuccessFully</**h1**>

<**form** action=*"getall"*>

<**input** type=*"submit"* value=*"click here to see all records"*>

</**form**>

</**body**>

</**html**>

<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>MphasisWebHibernate</groupId>

<artifactId>HibernateTeamplateWeb</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>HibernateTeamplateWeb Maven Webapp</name>

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

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>4.2.9.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-dbcp/commons-dbcp -->

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>3.6.9.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>3.6.9.Final</version>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

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

<version>8.0.29</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<finalName>HibernateTeamplateWeb</finalName>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>3.3.1</version>

</plugin>

</plugins>

</build>

</project>

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

<!-- collection of objects -->

<**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"*>

<**bean** id=*"id"* class=*"com.example.Student"*></**bean**>

<**bean** id=*"dao"* class=*"com.example.StudentDAO"*>

<**property** name=*"temp"* ref=*"tempref"*></**property**>

</**bean**>

<!-- HibernateTemplate Ref -->

<**bean** id=*"tempref"* class=*"org.springframework.orm.hibernate3.HibernateTemplate"*>

<**property** name=*"sessionFactory"* ref=*"Sessfref"*></**property**>

</**bean**>

<!-- session factory -->

<**bean** id=*"Sessfref"* class=*"org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"*>

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

<**property** name=*"packagesToScan"* value=*"com.example"*></**property**>

<**property** name=*"hibernateProperties"*>

<**props**>

<**prop** key=*"hibernate.dialect"*>org.hibernate.dialect.MySQLDialect</**prop**>

<**prop** key=*"hibernate.hbm2ddl.auto"*>update</**prop**>

<**prop** key=*"hibernate.show\_sql"*>true</**prop**>

</**props**>

</**property**>

</**bean**>

<!-- datasource -->

<**bean** id=*"ds"* class=*"org.apache.commons.dbcp.BasicDataSource"*>

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

<**property** name=*"url"* value=*"jdbc:mysql://localhost:3306/phase3"*></**property**>

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

<**property** name=*"password"* value=*"Gopi@123"*></**property**>

</**bean**>

</**beans**>