Spring – JDBC

Create a source folder “spring\_jdbc”

Create a package “com.springs.jdbc” in the source folder

Create the following java files in above package

**Create table student(id int, name varchar2(20), age int);**

Student.java

**package** com.springs.jdbc;

**public** **class** Student {

**private** Integer age;

**private** String name;

**private** Integer id;

**public** **void** setAge(Integer age) {

**this**.age = age;

}

**public** Integer getAge() {

**return** age;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** Integer getId() {

**return** id;

}

}

StudentDao.java

**package** com.springs.jdbc;

**import** java.util.List;

**import** javax.sql.DataSource;

**public** **interface** StudentDao {

**public** **void** setDataSource(DataSource ds);

**public** **void** create(Integer id,String name, Integer age);

**public** Student getStudent(Integer id);

**public** List<Student> listStudents();

**public** **void** delete(Integer id);

**public** **void** update(Integer id, Integer age);

}

StudentMapper.java

**package** com.springs.jdbc;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** org.springframework.jdbc.core.RowMapper;

**public** **class** StudentMapper **implements** RowMapper<Student> {

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

Student student = **new** Student();

student.setId(rs.getInt("id"));

student.setName(rs.getString("name"));

student.setAge(rs.getInt("age"));

**return** student;

}

}

StudentJdbcTemplate.java

**package** com.springs.jdbc;

**import** java.util.List;

**import** javax.sql.DataSource;

**import** org.springframework.jdbc.core.JdbcTemplate;

**public** **class** StudentJdbcTemplate **implements** StudentDao {

**private** DataSource dataSource;

**private** JdbcTemplate jdbcTemplateObject;

**public** **void** setDataSource(DataSource dataSource) {

**this**.dataSource = dataSource;

**this**.jdbcTemplateObject = **new** JdbcTemplate(dataSource);

}

**public** **void** create(Integer id,String name, Integer age) {

String SQL = "insert into Student (id,name, age) values (?,?,?)";

jdbcTemplateObject.update( SQL, id,name, age);

System.*out*.println("Created Record ID = "+id+ "Name = " + name + " Age = " + age);

**return**;

}

**public** Student getStudent(Integer id) {

String SQL = "select \* from Student where id = ?";

Student student = jdbcTemplateObject.queryForObject(SQL,

**new** Object[]{id}, **new** StudentMapper());

**return** student;

}

**public** List<Student> listStudents() {

String SQL = "select \* from Student";

List <Student> students = jdbcTemplateObject.query(SQL,

**new** StudentMapper());

**return** students;

}

**public** **void** delete(Integer id){

String SQL = "delete from Student where id = ?";

jdbcTemplateObject.update(SQL, id);

System.*out*.println("Deleted Record with ID = " + id );

**return**;

}

**public** **void** update(Integer id, Integer age){

String SQL = "update Student set age = ? where id = ?";

jdbcTemplateObject.update(SQL, age, id);

System.*out*.println("Updated Record with ID = " + id );

**return**;

}

}

JdbcTest.java

**package** com.springs.jdbc;

**import** java.util.List;

**import** org.springframework.beans.factory.BeanFactory;

**import** org.springframework.beans.factory.xml.XmlBeanFactory;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** org.springframework.core.io.FileSystemResource;

**public** **class** JdbcTest {

**public** **static** **void** main(String[] args) {

BeanFactory factory =

**new** XmlBeanFactory(**new** FileSystemResource("jdbc.xml"));

StudentJdbcTemplate studentJDBCTemplate =

(StudentJdbcTemplate)factory.getBean("studentJDBCTemplate");

System.*out*.println("------Records Creation--------" );

studentJDBCTemplate.create(111,"Raj", 18);

studentJDBCTemplate.create(222,"Ravi", 12);

studentJDBCTemplate.create(333,"Satish", 25);

System.*out*.println("------Listing Multiple Records--------" );

List<Student> students = studentJDBCTemplate.listStudents();

**for** (Student record : students) {

System.*out*.print("ID : " + record.getId() );

System.*out*.print(", Name : " + record.getName() );

System.*out*.println(", Age : " + record.getAge());

}

System.*out*.println("----Updating Record with ID = 222 -----" );

studentJDBCTemplate.update(222, 20);

System.*out*.println("----Listing Record with ID = 111 -----" );

Student student = studentJDBCTemplate.getStudent(111);

System.*out*.print("ID : " + student.getId() );

System.*out*.print(", Name : " + student.getName() );

System.*out*.println(", Age : " + student.getAge());

}

}

Create jdbc.xml file in the SpringProj

jdbc.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 "*>

<!-- Initialization for data source -->

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"oracle.jdbc.driver.OracleDriver"*/>

<property name=*"url"* value=*"jdbc:oracle:thin:@localhost:1521:orcl"*/>

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

<property name=*"password"* value=*"tiger"*/>

</bean>

<!-- Definition for studentJDBCTemplate bean -->

<bean id=*"studentJDBCTemplate"*

class=*"com.springs.jdbc.StudentJdbcTemplate"*>

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

</bean>

</beans>

SPRING MVC

Create a dynamic web project “SpringMVC”

Add Tomcat 7 server to the workspace

Add the SpringMVC libraries to the classpath of the project

Copy the SpringMVC libraries to WEB-INF\lib folder

Create index.jsp in WebContent

index.jsp

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

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

<!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>Spring 3, MVC Examples</title>

</head>

<body>

<h1>Spring 3, MVC Examples</h1>

<ul>

<li><a href=*"forms/loginform.html"*>Login Form</a></li>

</ul>

</body>

</html>

Create LoginForm.java in src folder

LoginForm.java

**package** com.springs.mvc;

**import** javax.validation.constraints.Size;

**import** org.hibernate.validator.constraints.NotEmpty;

**public** **class** LoginForm {

@NotEmpty

@Size(min = 1, max = 50)

**private** String userName;

@NotEmpty

@Size(min = 1, max = 20)

**private** String password;

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getPassword() {

**return** password;

}

}

Create LoginController.java in src folder

LoginController.java

**package** com.springs.mvc;

**import** org.springframework.stereotype.Controller;

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

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.validation.BindingResult;

**import** java.util.Map;

**import** javax.validation.Valid;

@Controller

@RequestMapping("loginform.html")

**public** **class** LoginController {

@RequestMapping(method = RequestMethod.*GET*)

public String showForm(Map model) {

LoginForm loginForm = new LoginForm();

model.put("loginForm", loginForm);

return "loginform";

}

@RequestMapping(method = RequestMethod.*POST*)

**public** String processForm(@Valid LoginForm loginForm, BindingResult result,

Map model) {

String userName = "spring";

String password = "spring";

**if** (result.hasErrors()) {

**return** "loginform";

}

loginForm = (LoginForm) model.get("loginForm");

**if** (!loginForm.getUserName().equals(userName)

|| !loginForm.getPassword().equals(password)) {

**return** "loginform";

}

model.put("loginForm", loginForm);

**return** "loginsuccess";

}

}

Create a folder “views” in WEB-INF

Create “loginform.jsp” in views folder

loginform.jsp

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

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

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<!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>Spring3Example</title>

</head>

<body>

<h3>Login Form</h3>

<FONT color=*"blue"*>

<h6>User Name="spring" and password="spring"</h6>

</FONT>

<form:form action=*"loginform.html"* commandName=*"loginForm"*>

<table>

<tr>

<td>User Name:<FONT color=*"red"*><form:errors

path=*"userName"* /></FONT></td>

</tr>

<tr>

<td><form:input path=*"userName"* /></td>

</tr>

<tr>

<td>Password:<FONT color=*"red"*><form:errors

path=*"password"* /></FONT></td>

</tr>

<tr>

<td><form:password path=*"password"* /></td>

</tr>

<tr>

<td><input type=*"submit"* value=*"Submit"* /></td>

</tr>

</table>

</form:form>

</body>

</html>

Create loginsuccess.jsp in views folder

loginsuccess.jsp

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

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

<%@ taglib prefix=*"core"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<!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>Spring3Example</title>

</head>

<body>

<h3>Welcome <core:out value=*"*${loginForm.userName}*"* /></h3>

<table>

<tr>

<td><a href=*"loginform.html"*>Back</a></td>

</tr>

</table>

</body>

</html>

Copy the following code in web.xml

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

<web-app version=*"2.5"* xmlns=*"http://java.sun.com/xml/ns/j2ee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_5.xsd"*>

<display-name>Spring3Example</display-name>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/forms/\*</url-pattern>

</servlet-mapping>

<welcome-file-list>

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

</welcome-file-list>

</web-app>

Create “dispatcher-servlet.xml” under WEB-INF folder

dispatcher-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"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

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*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd*

*"*>

<!-- Enable annotation driven controllers, validation etc... -->

<mvc:annotation-driven />

<context:component-scan base-package=*"com.springs.mvc"* />

<bean id=*"viewResolver"*

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"*>

<value>/WEB-INF/views/</value>

</property>

<property name=*"suffix"*>

<value>.jsp</value>

</property>

</bean>

<bean id=*"messageSource"*

class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<property name=*"basename"* value=*"/WEB-INF/messages"* />

</bean>

</beans>

Create messages.properties file in WEB-INF

messages.properties

NotEmpty.loginForm.userName=must not be blank.

Size.loginForm.userName=size must between 1 to 50 characters.

NotEmpty.loginForm.password=must not be blank.

Size.loginForm.password=size must between 1 to 20 characters.

Run the application by running index.jsp

**Integrating Spring and Hibernate**

Create a Java Project “SpringHibernate” and add Spring libraries and Hibernate libraries to the classpath of the project

Create a package com.raj.springhibernate

Create the following Java files in com.raj.springhibernate package

//Employee1.java

**package** com.raj.springhibernate;

**public** **class** Employee1 {

**private** **int** id;

**private** String name;

**private** **float** salary;

**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** **float** getSalary() {

**return** salary;

}

**public** **void** setSalary(**float** salary) {

**this**.salary = salary;

}

}

//EmployeeDao.java

**package** com.raj.springhibernate;

**import** org.springframework.orm.hibernate3.HibernateTemplate;

**public** **class** EmployeeDao {

HibernateTemplate template;

**public** **void** setTemplate(HibernateTemplate template) {

**this**.template = template;

}

**public** **void** saveEmployee(Employee1 e){

template.save(e);

}

**public** **void** updateEmployee(Employee1 e){

template.update(e);

}

**public** **void** deleteEmployee(Employee1 e){

template.delete(e);

}

}

//InsertTest.java

**package** com.raj.springhibernate;

**import** org.springframework.beans.factory.BeanFactory;

**import** org.springframework.beans.factory.xml.XmlBeanFactory;

**import** org.springframework.core.io.ClassPathResource;

**import** org.springframework.core.io.Resource;

**public** **class** InsertTest {

**public** **static** **void** main(String[] args) {

Resource r=**new** ClassPathResource("applicationContext.xml");

BeanFactory factory=**new** XmlBeanFactory(r);

EmployeeDao dao=(EmployeeDao)factory.getBean("d");

Employee1 e=**new** Employee1();

e.setId(147);

e.setName("kumar");

e.setSalary(70000);

dao.saveEmployee(e);

//dao.updateEmployee(e);

}

}

Create applicationContext.xml in src folder of SpringHibernateProj and copy the following code

<?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"*

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

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

<property name=*"driverClassName"* value=*"oracle.jdbc.driver.OracleDriver"*></property>

<property name=*"url"* value=*"jdbc:oracle:thin:@localhost:1521:orcl"*></property>

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

<property name=*"password"* value=*"tiger"*></property>

</bean>

<bean id=*"mysessionFactory"* class=*"org.springframework.orm.hibernate3.LocalSessionFactoryBean"*>

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

<property name=*"mappingResources"*>

<list>

<value>employee.hbm.xml</value>

</list>

</property>

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.Oracle10gDialect</prop>

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

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

</props>

</property>

</bean>

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

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

</bean>

<bean id=*"d"* class=*"com.raj.EmployeeDao"*>

<property name=*"template"* ref=*"template"*></property>

</bean>

</beans>

Create employee.hbm.xml in src folder of SpringHibernate project and copy the following code

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

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name=*"com.raj.Employee1"* table=*"Employee1"*>

<id name=*"id"*>

<generator class=*"assigned"*></generator>

</id>

<property name=*"name"*></property>

<property name=*"salary"*></property>

</class>

</hibernate-mapping>

Run the application “InsertTest.java”