Register page and configure User DAO and DTO

1. Include register page inside the views folder of front end project. Add following code:

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

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

<div class=*"container"*>

<div class=*"row form-container"*>

<form:form action=*"userData.do"* method=*"POST"* commandName=*"user"* modelAttribute=*"user"* >

<div class=*"col-md-3"*></div>

<div class=*"panel col-md-6"*

style="padding-bottom: *10px*; border: *1px solid*; background-color: *#337ab7*; color: *white*; border-color: *#2e6da4*;">

<div class=*"panel-heading"*

style="text-align: *center*; font-weight: *bolder*">

<h2>Sign up now</h2>

<h4>Fill in the form below to get instant access</h4>

<hr />

</div>

<!-- Text input-->

<div class=*"row form-group"* style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:input path=*"userName"* id=*"userName"* type=*"text"* placeholder=*"User Name"* class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userName"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

</div>

<!-- Text input-->

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-6"*>

<form:input path=*"userPassword"* type=*"password"* placeholder=*"Password"*

class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userPassword"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

<div class=*"col-md-6"* style="padding-left: *2%*">

<form:input path=*"userConfirmPassword"* type=*"Password"*

placeholder=*"Confirm Password"* class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userConfirmPassword"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

</div>

<!-- Text input-->

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:input path=*"userFullName"* type=*"text"* placeholder=*"Name"*

class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userFullName"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

</div>

<!-- Text Select -->

<div class=*"row form-group"* style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:select path=*"userRole"* class=*"form-control"*>

<form:option value=*"0"* label=*"-- Select User Role --"* selected=*"true"*/>

<form:option value=*"customer"* label=*"Customer"*/>

<form:option value=*"supplier"* label=*"Supplier"*/>

</form:select>

</div>

</div>

<!-- Text input-->

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:input path=*"userEmail"* type=*"email"* placeholder=*"Email"*

class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userEmail"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

</div>

<!-- Text input-->

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:input path=*"userPhoneNumber"* type=*"tel"* placeholder=*"Phone Number"*

class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userPhoneNumber"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

</div>

<!-- Text input-->

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:input path=*"userQuestion"* type=*"text"* placeholder=*"Security Question"*

class=*"form-control input-md"*/>

</div>

</div>

<!-- Text input-->

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*">

<div class=*"col-md-12"*>

<form:input path=*"userAnswer"* type=*"text"* placeholder=*"Security Answer"*

class=*"form-control input-md"*/>

<div class=*"has-error"*>

<form:errors path=*"userAnswer"* class=*"help-inline"* style=*"color:red"*/>

</div>

</div>

</div>

<div class=*"row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*"">

<div class=*"col-md-12"*></div>

</div>

<div class=*" row form-group"*

style="padding-left: *10%*; padding-right: *10%*; padding-bottom: *2%*"">

<div class=*"col-md-12"*>

<input type=*"submit"* name=*"action"* value=*"Add"* class=*"btn btn-primary btn-sm"*>

</div>

</div>

</div>

</form:form>

</div>

</div>

1. Create a class inside the backend project named “User” inside the model package and add following code.

**package com.health.HealthMedicineQuestBackEnd.model;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**import javax.persistence.Transient;**

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

**@Entity**

**public class User{**

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**int** userId;

@Size(min=1, max=30,message="user name should be between 1 to 30 characters long")

String userName;

@Size(min=1, max=30,message="user password should be between 1 to 30 characters long")

String userPassword;

@Size(min=1, max=30,message="user full name should be between 1 to 30 characters long")

String userFullName;

**public** String getUserFullName() {

**return** userFullName;

}

**public** **void** setUserFullName(String userFullName) {

**this**.userFullName = userFullName;

}

String userRole;

@Transient

@Size(min=1, max=30,message="user confirm password should be between 1 to 30 characters long")

String userConfirmPassword;

@Size(min=1, max=30,message="user phone number should be between 1 to 30 characters long")

String userPhoneNumber;

@Size(min=1, max=30,message="user email should be between 1 to 50 characters long")

String userEmail;

String userQuestion;

@Size(min=1, max=30,message="user answer should be between 1 to 30 characters long")

String userAnswer;

**boolean enabled;**

**public boolean isEnabled() {**

**return enabled;**

**}**

**public void setEnabled(boolean enabled) {**

**this.enabled = enabled;**

**}**

**public User() {**

**super();**

**// TODO Auto-generated constructor stub**

**}**

**public User(int userId, String userName, String userPassword, String userFullName,String userRole, String userConfirmPassword,**

**String userPhoneNumber, String userEmail, String userQuestion, String userAnswer) {**

**super();**

**this.userId = userId;**

**this.userName = userName;**

**this.userPassword = userPassword;**

**this.userFullName=userFullName;**

**this.userRole = userRole;**

**this.userConfirmPassword = userConfirmPassword;**

**this.userPhoneNumber = userPhoneNumber;**

**this.userEmail = userEmail;**

**this.userQuestion = userQuestion;**

**this.userAnswer = userAnswer;**

**this.enabled=false;**

**}**

**public int getUserId() {**

**return userId;**

**}**

**public void setUserId(int userId) {**

**this.userId = userId;**

**}**

**public String getUserName() {**

**return userName;**

**}**

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

**this.userName = userName;**

**}**

**public String getUserPassword() {**

**return userPassword;**

**}**

**public void setUserPassword(String userPassword) {**

**this.userPassword = userPassword;**

**}**

**public String getUserRole() {**

**return userRole;**

**}**

**public void setUserRole(String userRole) {**

**this.userRole = userRole;**

**}**

**public String getUserConfirmPassword() {**

**return userConfirmPassword;**

**}**

**public void setUserConfirmPassword(String userConfirmPassword) {**

**this.userConfirmPassword = userConfirmPassword;**

**}**

**public String getUserPhoneNumber() {**

**return userPhoneNumber;**

**}**

**public void setUserPhoneNumber(String userPhoneNumber) {**

**this.userPhoneNumber = userPhoneNumber;**

**}**

**public String getUserEmail() {**

**return userEmail;**

**}**

**public void setUserEmail(String userEmail) {**

**this.userEmail = userEmail;**

**}**

**public String getUserQuestion() {**

**return userQuestion;**

**}**

**public void setUserQuestion(String userQuestion) {**

**this.userQuestion = userQuestion;**

**}**

**public String getUserAnswer() {**

**return userAnswer;**

**}**

**public void setUserAnswer(String userAnswer) {**

**this.userAnswer = userAnswer;**

**}**

2. **}**Create an interface named “IUserDAO” inside the dao package and add following code:

**package** com.health.HealthMedicineQuestBackEnd.dao;

**import** java.util.List;

**import** com.health.HealthMedicineQuestBackEnd.model.User;

**public** **class** IUserDAO {

**public** List<User> getAllUsers();

**public** User getUser(**int** id);

**public** **boolean** addUser(User u);

**public** **boolean** updateUser(User u);

**public** **boolean** deleteUser(**int** userId);

}

1. Create a class named UserDAOImpl inside daoimpl package and add following code.

**package com.health.HealthMedicineQuestBackEnd.daoimpl;**

**import java.util.ArrayList;**

**import java.util.List;**

**import org.hibernate.SessionFactory;**

**import org.springframework.beans.factory.annotation.Autowired;**

**import org.springframework.stereotype.Repository;**

**import org.springframework.transaction.annotation.Transactional;**

**import com.health.HealthMedicineQuestBackEnd.model.Product;**

**import com.health.HealthMedicineQuestBackEnd.model.User;**

**import com.health.HealthMedicineQuestBackEnd.dao.IUserDAO;;**

**@Repository("UserDAO")**

**@Transactional**

**public class UserDAOImpl implements IUserDAO{**

**@Autowired**

**SessionFactory sessionFactory;**

**public List<User> getAllUsers(){**

**return sessionFactory.getCurrentSession().createQuery("from User", User.class).getResultList();**

**}**

**public User getUser(int id){**

**return sessionFactory.getCurrentSession().get(User.class,Integer.valueOf(id));**

**}**

**public boolean addUser(User u){**

**try {**

**sessionFactory.getCurrentSession().persist(u);**

**return true;**

**} catch (Exception e) {**

**e.printStackTrace();**

**return false;**

**}**

**}**

**public boolean updateUser(User u){**

**try {**

**sessionFactory.getCurrentSession().update(u);**

**return true;**

**} catch (Exception e) {**

**e.printStackTrace();**

**return false;**

**}**

**}**

**public boolean deleteUser(int userId){**

**try {**

**sessionFactory.getCurrentSession().delete(getUser(userId));**

**return true;**

**} catch (Exception e) {**

**e.printStackTrace();**

**return false;**

**}**

**}**

**}**

1. Open page .jsp and add the following code

<!-- Loading only when user clicks register -->

<c:if test=*"*${userClickRegister==**true**}*"*>

<%@include file=*"register.jsp"* %>

</c:if>

1. Open FrontEndProductController and type the following code:

package com.health.HealthMedicineQuestFrontEnd.controller;

import java.io.File;

import java.util.List;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

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

import org.springframework.web.servlet.ModelAndView;

import com.health.HealthMedicineQuestBackEnd.dao.IUserDAO;

import com.health.HealthMedicineQuestBackEnd.model.Product;

import com.health.HealthMedicineQuestBackEnd.model.User;

@Controller

public class FrontEndUserController {

@Autowired

private IUserDAO userDAO;

@RequestMapping(value={"register"})

public ModelAndView register(){

ModelAndView model =new ModelAndView("page");

model.addObject("title","Register");

model.addObject("userClickRegister","true");

model.addObject("user",new User());

return model;

}

// @RequestMapping("/allusers")

// public @ResponseBody List<User> alluser(){

// return userDAO.getAllUser();

// }

// @RequestMapping(value={"userCRUD"})

// public ModelAndView usercrud(){

// ModelAndView model =new ModelAndView("page");

// model.addObject("title","User Management");

// model.addObject("userClickUserCRUD","true");

// model.addObject("user",new User());

// //model.addObject("userList",userDAO.getAllUser());

// return model;

// }

@RequestMapping(value="/userData.do", method=RequestMethod.POST)

private ModelAndView doActions(@ModelAttribute("user") @Valid User user,BindingResult result) {

if(result.hasErrors()) {

ModelAndView model1 =new ModelAndView("page");

model1.addObject("title","Register");

model1.addObject("userClickRegister","true");

return model1;

}

userDAO.addUser(user);

ModelAndView model1 =new ModelAndView("page");

model1.addObject("title","Home");

model1.addObject("userClickHome","true");

model1.addObject("user",new User());

return model1;

}

}

1. Create a class named “UserTestCase” inside test package of BackEnd project and add the following code to test UserDAO.

**package** com.health.HealthMedicineQuestBackEnd.test;

**import** **static** org.junit.Assert.\*;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**import** com.health.HealthMedicineQuestBackEnd.dao.IUserDAO;

**import** com.health.HealthMedicineQuestBackEnd.model.User;

**public** **class** UserTestCase {

**private** **static** AnnotationConfigApplicationContext *context*;

**private** **static** IUserDAO *userDAO*;

**private** User user;

@BeforeClass

**public** **static** **void** init(){

*context*=**new** AnnotationConfigApplicationContext();

*context*.scan("com.health.HealthMedicineQuestBackEnd");

*context*.refresh();

*userDAO*=(IUserDAO)*context*.getBean("userDAO");

}

@Test

**public** **void** testAddUser(){

user=**new** User();

user.setUserName("LopaB");

user.setUserPassword("lopab");

user.setUserConfirmPassword("lopab");

user.setUserEmail("pinkfriend@gmail.com");

user.setUserPhoneNumber("9339901411");

user.setUserRole("user");

user.setUserQuestion("What is your nick name?");

user.setUserAnswer("pinky");

*assertEquals*("Successfully added a user inside the table!", **true**, *userDAO*.addUser(user));

}

@Test

**public** **void** testGetUser(){

user=*userDAO*.getUser(1);

*assertEquals*("Successfully added a user inside the table!", "LopaB", user.getUserName());

}

@Test

**public** **void** testUpdateUser(){

user=*userDAO*.getUser(1);

//assertEquals("Successfully added a user inside the table!", "LopaB", user.getUserName());

//user=new User();

user.setUserName("LopaB");

user.setUserPassword("lopabm");

user.setUserConfirmPassword("lopabm");

user.setUserEmail("pinkfriend@gmail.com");

user.setUserPhoneNumber("9339901411");

user.setUserRole("user");

user.setUserQuestion("What is your nick name?");

user.setUserAnswer("pinky");

*assertEquals*("Successfully updated a user inside the table!", **true**, *userDAO*.updateUser(user));

}

@Test

**public** **void** testDeleteUser(){

*assertEquals*("Successfully deleted a user from the table!", **true**, *userDAO*.deleteUser(1));

}

}