**Spring Form Validation Tutorial**

Here is our UserValidator class.

01.package com.tutorials4u.validator;

02.

03.import org.springframework.validation.Errors;

04.import org.springframework.validation.ValidationUtils;

05.import org.springframework.validation.Validator;

06.

07.import com.tutorials4u.domain.User;

08.

09.public class UserValidator implements Validator {

10.

11.    public boolean supports(Class<!--?--> clazz) {

12.        return User.class.isAssignableFrom(clazz);

13.    }

14.

15.    public void validate(Object target, Errors errors) {

16.        User user = (User) target;

17.        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "name", "name.required");

18.        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "email", "email.required");

19.        ValidationUtils.rejectIfEmpty(errors, "profession", "profession.required");

20.        if(user.getCountry().equals("select"))

21.        {

22.            errors.rejectValue("country", "country.required");

23.        }

24.        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "feedback", "feedback.required");

25.        if(user.getFrameworks().length == 0)

26.        {

27.            errors.rejectValue("frameworks","frameworks.required");

28.        }

29.    }

30.

31.}

Here you need to override the validate method. To check mandatory condition you can use the ValidationUtils class methods like rejectIfEmptyOrWhitespace or rejectIfEmpty. These methods takes three arguments the Errors object, the property name, and the error code. Here we have the error messages in a seperate properties file so we add the error code, you can even add the error messages directly. To do any other validation you have access to the domain object, using the domain object validate the properties, incase there are errors you can use the rejectValue() method of the Errors class to add errors. Here the first argument is the property name and the second argument is the error code.

The messages.properties file contains the following error keys and values.

1.name.required = User Name is required

2.email.required = Email is required

3.profession.required = Profession is required

4.country.required = Country is required

5.feedback.required = Feedback is required

6.frameworks.required = Select at least one framework

Now add the form:errors tags in the jsp page to display the errors.

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

02.    pageEncoding="ISO-8859-1"%>

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

04.<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

05.<html>

06.<head>

07.<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

08.<title>Feedback Form</title>

09.<style>

10..error {

11.color: #ff0000;

12.font-style: italic;

13.}

14.</style>

15.</head>

16.<body>

17.

18.<form:form method="POST" commandName="user">

19.    <table>

20.        <tr>

21.            <td>Name :</td>

22.            <td><form:input path="name" /></td>

23.            <td><form:errors path="name" cssClass="error" /></td>

24.        </tr>

25.        <tr>

26.            <td>Profession:</td>

27.            <td><form:radiobutton path="profession" value="Student" label="Student" />

28.                <form:radiobutton path="profession" value="Programmer" label="Programmer" /></td>

29.                <td><form:errors path="profession" cssClass="error" /></td>

30.        </tr>

31.        <tr>

32.            <td>Email:</td>

33.            <td><form:input path="email" /></td>

34.            <td><form:errors path="email" cssClass="error" /></td>

35.        </tr>

36.        <tr>

37.            <td>Country :</td>

38.            <td><form:select path="country">

39.                <form:option value="select" label="Select" />

40.                <form:option value="India" label="India" />

41.                <form:option value="United States" label="United States" />

42.                <form:option value="United Kingdom" label="United Kingdom" />

43.                <form:option value="Europe" label="Europe" />

44.                <form:option value="Canada" label="Canada" />

45.            </form:select></td>

46.            <td><form:errors path="country" cssClass="error" /></td>

47.        </tr>

48.        <tr>

49.            <td>Feedback:</td>

50.            <td><form:textarea path="feedback" /></td>

51.            <td><form:errors path="feedback" cssClass="error" /></td>

52.        </tr>

53.        <tr>

54.            <td>Frameworks:</td>

55.            <td>

56.                <form:checkbox path="frameworks" value="Spring" label="Spring" />

57.                <form:checkbox path="frameworks" value="Hibernate" label="Hibernate" />

58.                <form:checkbox path="frameworks" value="Struts" label="Struts" />

59.                <form:checkbox path="frameworks" value="Dojo" label="Dojo" />

60.            </td>

61.            <td><form:errors path="frameworks" cssClass="error" /></td>

62.        </tr>

63.        <tr>

64.            <td></td>

65.            <td><form:checkbox path="mailingList"

66.                label="Would you like to join our mailinglist?" /></td>

67.        </tr>

68.        <tr>

69.            <td colspan="2"><input type="submit" value="submit"></td>

70.        </tr>

71.    </table>

72.</form:form>

73.

74.</body>

75.</html>

The path attribute of the form:errors tag indicate the property for which the error should be displayed. Use path="\*" to display all the errors together.

In the Spring configuration file, link the UserValidator with the UserController using the validator-ref property and add the properties file.

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

02.<beans xmlns="http://www.springframework.org/schema/beans"

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

04.    xmlns:p="http://www.springframework.org/schema/p"

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

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

07.

08.    <bean id="viewResolver"

09.          class="org.springframework.web.servlet.view. InternalResourceViewResolver" p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />

10.

11.    <bean id="messageSource" class="org.springframework.context.support. ResourceBundleMessageSource" p:basename="messages" />

12.

13.    <bean id="userService" class="com.tutorials4u.service.UserServiceImpl" />

14.

15.    <bean id="userValidator" class="com.tutorials4u.validator.UserValidator" />

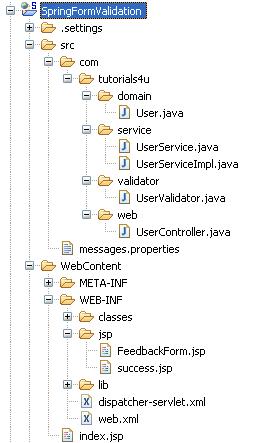
16.

17.    <bean name="/feedbackForm.htm" class="com.tutorials4u.web.UserController" p:userService-ref="userService" p:formView="FeedbackForm" p:successView="success" p:validator-ref="userValidator" />

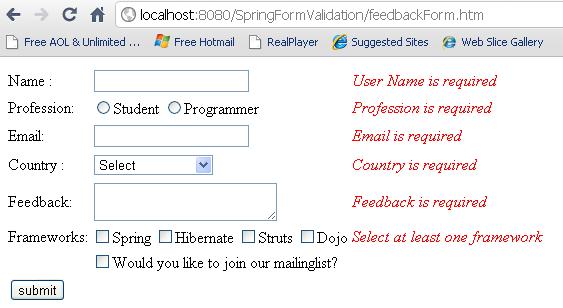
18.

19.</beans>

The directory structure of the example is show below. The **messages.properties** file should be placed in the root of the classpath.



Run the example. Click the **Submit** button without filling any details. You will see the following error messages.



**Spring Form Validation Using Annotation Tutorial**

Learn how to validate the Spring form using annotated controllers.

In this example you will learn the changes you need to make to the previous validation example to make it work with Spring annotations. When using annotated controller class the validate() method will not be automatically called. The onSubmit() method will be called when the form is submitted. Here you need to first call the validate() method of the UserValidator class, then check for any error using the hasErrors() method, if there are any errors then, forward the user to the user registration form and dispaly the errors, if there are no errors then display the success page. Here is the UserController class.

01.package com.tutorials4u.web;

02.

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

04.import org.springframework.stereotype.Controller;

05.import org.springframework.ui.ModelMap;

06.import org.springframework.validation.BindingResult;

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

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

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

10.import org.springframework.web.bind.annotation.SessionAttributes;

11.

12.import com.tutorials4u.domain.User;

13.import com.tutorials4u.service.UserService;

14.import com.tutorials4u.validator.UserValidator;

15.

16.@Controller

17.@RequestMapping("/userRegistration.htm")

18.@SessionAttributes("user")

19.public class UserController {

20.

21.    private UserService userService;

22.    private UserValidator userValidator;

23.

24.    @Autowired

25.    public UserController(UserService userService, UserValidator userValidator) {

26.        this.userService = userService;

27.        this.userValidator = userValidator;

28.    }

29.

30.    @RequestMapping(method = RequestMethod.GET)

31.    public String showUserForm(ModelMap model) {

32.        User user = new User();

33.        model.addAttribute("user", user);

34.        return "userForm";

35.    }

36.

37.    @RequestMapping(method = RequestMethod.POST)

38.    public String onSubmit(@ModelAttribute("user") User user,

39.            BindingResult result) {

40.        userValidator.validate(user, result);

41.        if (result.hasErrors()) {

42.            return "userForm";

43.        } else {

44.            userService.add(user);

45.            return "redirect:userSuccess.htm";

46.        }

47.    }

48.

49.}

Here the UserService and the UserValidator classes are injected using the @Autowired annotation. The Spring bean configuration file has the following entries.

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

02.<beans xmlns="http://www.springframework.org/schema/beans"

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

04.    xmlns:p="http://www.springframework.org/schema/p"

05.    xmlns:context="http://www.springframework.org/schema/context"

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

07.    http://www.springframework.org/schema/beans/spring-beans.xsd

08.    http://www.springframework.org/schema/context

09.    http://www.springframework.org/schema/context/spring-context.xsd">

10.

11.    <bean id="viewResolver" class="org.springframework.web.servlet.view. InternalResourceViewResolver" p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />

12.

13.    <bean id="messageSource" class="org.springframework.context.support. ResourceBundleMessageSource" p:basename="messages" />

14.

15.    <context:component-scan base-package="com.tutorials4u.web" />

16.

17.    <bean id="userService" class="com.tutorials4u.service.UserServiceImpl" />

18.

19.

20.    <bean id="userValidator" class="com.tutorials4u.validator.UserValidator" />

21.

22.</beans>