Spring Web MVC – III

(Spring Form Validation)

(Using Spring Validator Interface)

Table of Contents

Introduction

1. Spring provides a Validator interface that can be used for validation in all layers of an application.
2. In Spring MVC you can configure it for use as a global **Validator** instance.
3. To use this validator, we pass **@Valid** or **@Validated** annotation in controller method argument.
4. As a local validator within a controller, we can use an **@InitBinder** method and configure validator.
5. Global and local validator instances can be combined to provide composite validation.

Global Validator Configuration(Using Java Configuration)

```
public class WebMvcConfig extends WebMvcConfigurerAdapter {

        public Validator getValidator( ) ; {
                //return “global” validator
        }
}
```

Global Validator Configuration(Using XML Configuration)

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

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:mvc="http://www.springframework.org/schema/mvc"
        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.xsd
                http://www.springframework.org/schema/mvc
                http://www.springframework.org/schema/mvc/spring-mvc.xsd">

        <mvc:annotation-driven validator=”globalValidator” />

</beans>
```

Local Validator Configuration

```
@Controller
public class UserController {

        @InitBinder
        protected void initBinder(WebDataBinder binder)  {
             binder.addValidators(new UserValidator( ) );
        }
}
```

Validator Interface

The Validator Interface is not coupled with any tier of application. We can use this to validate the Objects in the web tier, the data-access tier or the whatever tier.

This interface consists only two methods, one for linking the object for validation and second for validating the properties of the object.

```
package org.springframework.validation;
public interface Validator  {
      boolean supports(Class clazz);
      void validate(Object target, Errors errors);
}
```

ValidationUtil Class

ValidationUtil is a utility class offering convenient methods for invoking a Validator and for rejecting empty fields.
We can check for an empty field in Validator implementation class using two methods.

ValidationUtils.rejectIfEmpty(Errors errors, String property, Strring errorCode);
ValidationUtils.rejectIfEmptyOrWhitespace(Errors errors, String property, Strring errorCode);

WebDataBinder

This is a **DataBinder** for data binding from web request parameters to JavaBean Objects. It is specially designed for web environments, but not dependent on Servlet API's.

To configure local **Validator** in Controller, we use **WebDataBinder** to bind the request parameters.

```
@Controller
public class UserController  {
        @InitBinder
        protected void initBinder(WebDataBinder binder)  {
              binder.addValidators(new UserValidator( ));
        }
}
```

BindingResult

It is an interface that represents binding results. It Extends the Errors Interface for error registrartion capabilities, allowing for a validator to be applied.

We pass BindingResult as an arguments in controller methods.

```
@RequestMapping(value="/register" method=RequestMethod.POST)
public ModelAndView  registerUser(@ModelAttribute User user, BindingResult result)
{
        if(result.hasErrors( ))
                return new ModelAndView("welcome");
        else
                return new ModelAndView("regSuccess", "user" , user);
}
```

ResourceBundleMessageSource

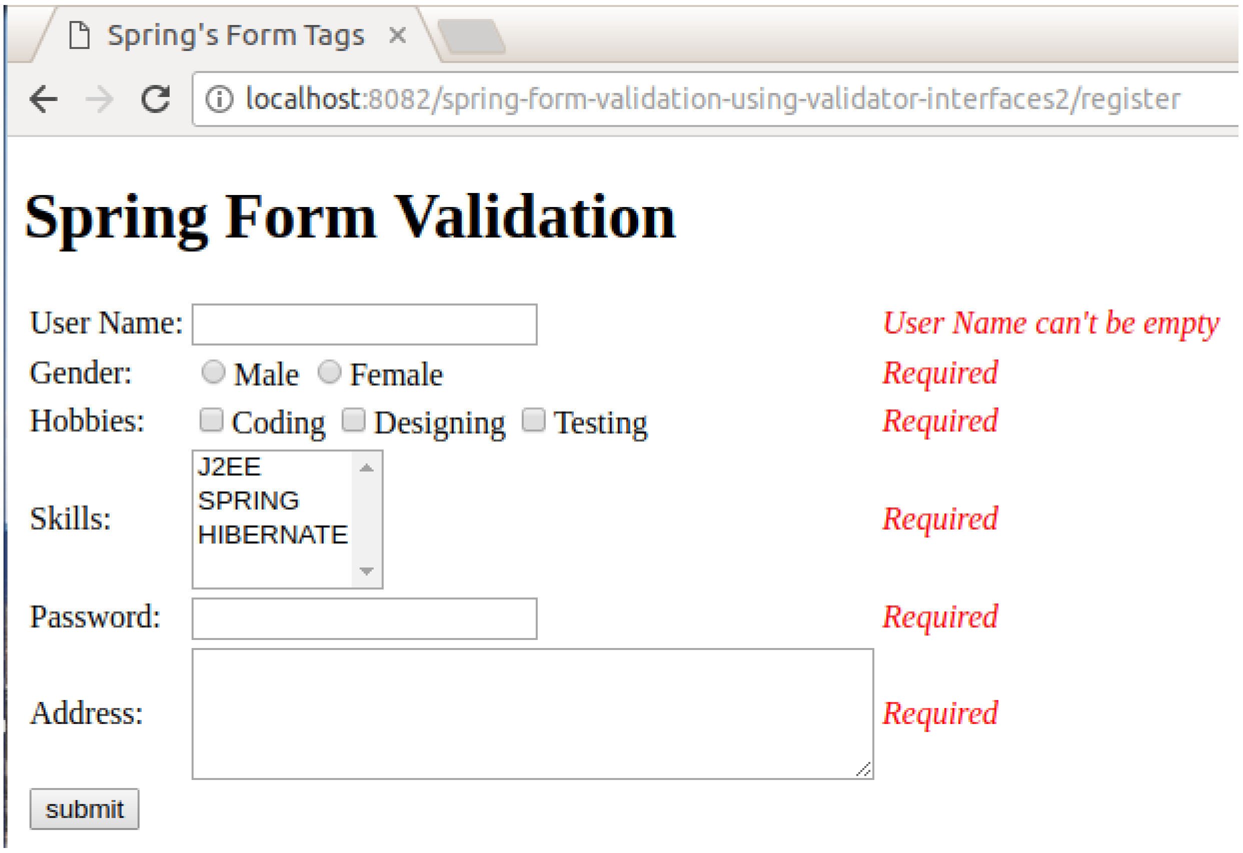
This is an implementation of org.springframework.context.MessageSource that access resource bundles using specified basenames. It loaded message-source from the properties file that is located in classpath.

To use ResourceBundMessageSource, first we configure it in WebApplicationContext

```
<bean id="messageSource"
            class="org.springframework.context.support.ResourceBundleMessageSource">
            <property name="basename" value="messages"></property>
</bean>
```

It will read **messages.properties** file from classpath and pass all error messages to view page.

Spring Form validation Application


Spring's Form Tags

localhost:8082/spring-form-validation-using-validator-interfaces2/register

Spring Form Validation

User Name:		User Name can't be empty
Gender:	○ Male ○ Female	Required
Hobbies:	☐ Coding ☐ Designing ☐ Testing	Required
Skills:	J2EE SPRING HIBERNATE	Required
Password:		Required
Address:		Required

pom.xml

```
<dependencies>
    <!-- JSTL -->
    <dependency>
            <groupId>jstl</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
    </dependency>

    <!-- Spring Web MVC -->
     <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>4.3.5.RELEASE</version>
    </dependency>
</dependencies>
```

web.xml

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

  <display-name>Spring-webmvc-app</display-name>
  <welcome-file-list>
          <welcome-file>/</welcome-file>
  </welcome-file-list>

  <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>/</url-pattern>
  </servlet-mapping>

</web-app>
```

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

        <context:annotation-config />
        <context:component-scan base-package="com.Biditvats.spring.*" />

        <bean id="viewResolver"
                class="org.springframework.web.servlet.view.InternalResourceViewResolver">
                <property name="prefix" value="/WEB-INF/views/" />
                <property name="suffix" value=".jsp" />
        </bean>
        <bean id="messageSource"
            class="org.springframework.context.support.ResourceBundleMessageSource">
                <property name="basename" value="messages"></property>
            </bean>

</beans>
```

welcome.jsp

```
<%@ page isELIgnored="false" language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="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=UTF-8">
<title>Spring's Form Tags</title>
<style type="text/css">
        .error{ color:#FF0000; font-style:italic; }
</style>
</head>
```

```
<body>
        <h1>Spring Form Validation</h1>
        <form:form method="POST" modelAttribute="user" action="register">
                <table>
                        <tr>
                                <td>User Name:</td>
                                <td><form:input path="userName"/></td>
                                <td><form:errors path="userName" cssClass="error" /></td>
                        </tr>
                        <tr>
                                <td>Gender:</td>
                                <td>
                                        <form:radiobutton path="gender" value="Male"/>Male
                                        <form:radiobutton path="gender" value="Female"/>Female
                                </td>
                                <td><form:errors path="gender" cssClass="error" /></td>
                        </tr>
                        <tr>
                                <td>Hobbies:</td>
                                <td>
                                        <form:checkbox path="hobbies" value="Coding"/>Coding
                                        <form:checkbox path="hobbies" value="Designing"/>Designing
                                        <form:checkbox path="hobbies" value="Testing"/>Testing
                                </td>
                                <td><form:errors path="hobbies" cssClass="error" /></td>
                        </tr>
                        <tr>
                                <td>Skills:</td>
                                <td>
                                        <form:select multiple="true" path="skills">
                                                <option value="J2EE">J2EE</option>
                                                <option value="Spring">SPRING</option>
                                                <option value="Hibernate">HIBERNATE</option>
                                        </form:select>
                                </td>
                                <td><form:errors path="skills" cssClass="error" /></td>
                        </tr>
                        <tr>
                                <td>Password:</td>
                                <td>
                                        <form:password path="password"/>
                                </td>
                                <td><form:errors path="password" cssClass="error" /></td>
                        </tr>
                        <tr>
                                <td>Address:</td>
                                <td><form:textarea path="address" rows="4" cols="40"/></td>
                                <td><form:errors path="address" cssClass="error"></form:errors></td>
                        </tr>
                        <tr>
                                <td colspan="2">
                                        <input type="submit" value="submit">
                                </td>
                        </tr>
                </table>
```

```
	</form:form>
</body>
</html>
```

regSuccess.jsp

```
<%@ page isELIgnored="false" language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!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=UTF-8">
<title>Show User Details</title>
</head>
<body>
	<h1>User Details</h1>
	<p>User Name: ${user.userName}</p>
	<p>Gender: ${user.gender}</p>
	<p>Hobbies:
			<c:forEach begin="0" items="${user.hobbies}" var="hob">
					<c:out value="${hob}"></c:out>
			</c:forEach>
	</p>
	<p>Skills:
		<c:forEach begin="0" items="${user.skills}" var="skill">
				<c:out value="${skill}"></c:out>
		</c:forEach>
	</p>
	<p>Password: ${user.password}</p>
	<p>Address: ${user.address}</p>
</body>
</html>
```

User.java

```
package com.Biditvats.spring.bean;

import org.springframework.stereotype.Component;

@Component("user")
public class User {
	private String userName;
	private String gender;
	private String[] hobbies;
	private String[] skills;
	private String password;
	private String address;

	public User() {
		System.out.println("User object is created");
	}

	//Getters and Setters
}
```

UserValidator.java

```
package com.Biditvats.spring.validator;

import org.springframework.stereotype.Component;
import org.springframework.validation.Errors;
import org.springframework.validation.ValidationUtils;
import org.springframework.validation.Validator;

import com.Biditvats.spring.bean.User;

@Component
public class UserValidator implements Validator {

	@Override
	public boolean supports(Class<?> clazz) {
		// TODO Auto-generated method stub
		return User.class.isAssignableFrom(clazz);
	}

	@Override
	public void validate(Object target, Errors errors) {
		// TODO Auto-generated method stub
		ValidationUtils.rejectIfEmpty(errors, "userName", "userName.required");
		ValidationUtils.rejectIfEmpty(errors, "gender", "gender.required");
		ValidationUtils.rejectIfEmpty(errors, "password", "password.required");
		ValidationUtils.rejectIfEmpty(errors, "address", "address.required");

		User user = (User) target;
		if(user.getHobbies().length == 0)
			errors.rejectValue("hobbies", "hobbies.required");
		if(user.getSkills().length ==0)
			errors.rejectValue("skills", "skills.required");
	}

}
```

messages.properties

```
userName.required = User Name can't be empty
gender.required = Required
hobbies.required = Required
skills.required = Required
address.required = Required
password.required = Required
```

UserController.java

```
package com.Biditvats.spring.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import com.Biditvats.spring.bean.User;
import com.Biditvats.spring.validator.UserValidator;

@Controller
public class UserController {

	@Autowired
	UserValidator userValidator;

	@InitBinder
	protected void initBinder(WebDataBinder binder) {
		binder.setValidator(userValidator);
	}

	@RequestMapping("/")
	public String userForm(Model model) {
		User user = new User();
		model.addAttribute("user", user);
		return "welcome";
	}

	@RequestMapping("/register")
	public ModelAndView registerUser(@ModelAttribute @Validated User user,
			BindingResult result) {
		if(result.hasErrors())
			return new ModelAndView("welcome");
		else
			return new ModelAndView("regSuccess", "user",user);
	}
}
```