



## **Spring Web MVC – III**

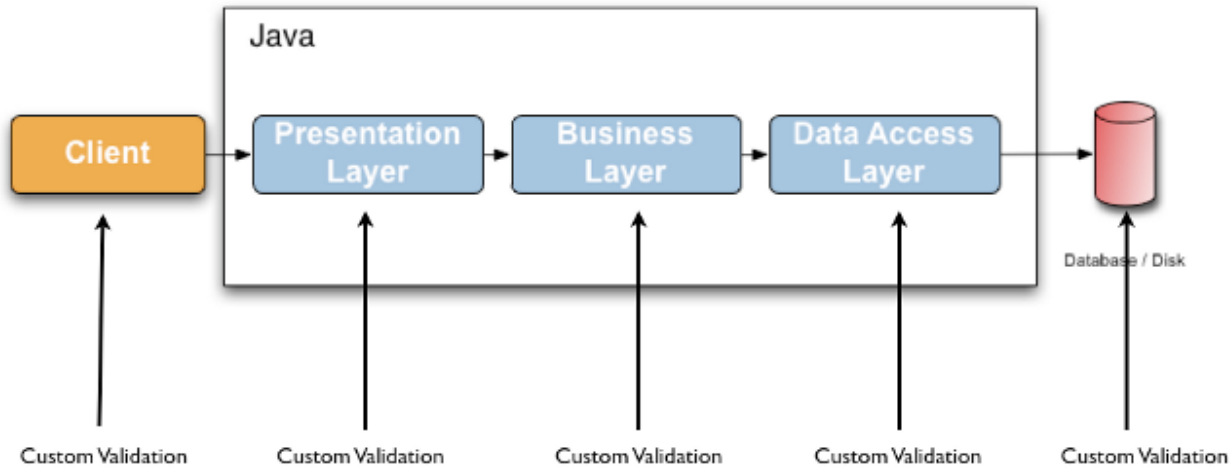
### **(Spring Bean Validation) (Using JSR-303)**

## **Table of Contents**

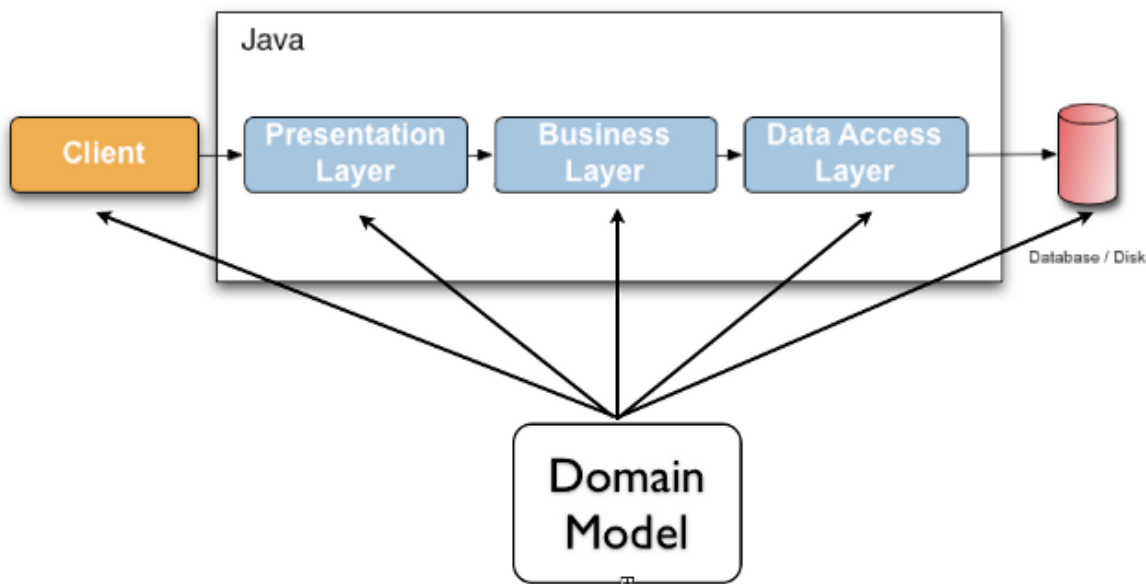
- 1.** Introduction
- 2.** Requirements
- 3.** JSR-303 Annotations
- 4.** JSR-303 Maven Dependencies
- 5.** Hibernate-Validator Annotations
- 6.** Hibernate-Validator Maven Dependencies
- 7.** Spring Bean Validation Application

## Introduction

Validating data is a common task that occurs throughout all application layers, from the presentation to the persistence layer. Often the same validation logic is implemented in each layer which is time consuming and error-prone. To avoid duplication of these validations in each layer, developers often bundle validation logic directly into the domain model with validation code which is really metadata about the class itself.



JSR 303 - Bean Validation - defines a metadata model and API for entity and method validation. The default metadata source are annotations, with the ability to override and extend the meta-data through the use of XML. The API is not tied to a specific application tier nor programming model. It is specifically not tied to either web or persistence tier, and is available both server-side application programming, as well as rich client Swing application developers.



JSR-303 is a validation framework available as part of Java EE 6 platform. The JSR-303 is supported by constraints in the form of annotations placed on a field, method, or class of a JavaBeans component, such as a managed bean.

**There are two types of constraints-**

1. Built-in constraints (available in the **javax.validation.constraints** package)
2. User-defined constraints (Created by user)

## Built-In Bean Validation Constraints

| Constraint/<br>Annotation | Description  | Example   |
|---------------------------|--|---|
| @AssertFalse              | The value of the field or property must be false.  |   |
|                           |  | @AssertFalse<br>boolean isUnsupported;              |
|                           |  |   |
| @AssertTrue               | The value of the field or property must be true.   |   |
|                           |  | @AssertTrue<br>boolean isActive;                    |
|                           |  |   |
| @DecimalMax               | The value of the field or property must be a decimal value lower than or equal to the number in the value element.   |   |
|                           |  | @DecimalMax("30.00")<br>BigDecimal discount;        |
|                           |  |   |
| @DecimalMin               | The value of the field or property must be a decimal value greater than or equal to the number in the value element.   |   |
|                           |  | @DecimalMin("5.00")<br>BigDecimal discount;         |
|                           |  |   |
| @Digits                   | The value of the field or property must be a number within a specified range. The integer element specifies the maximum integral digits for the number, and the fraction element specifies the maximum fractional digits for the number. |   |
|                           |  | @Digits(integer=6, fraction=2)<br>BigDecimal price; |
|                           |  |   |
| @Future                   | The value of the field or property must be a date in the future.   |   |
|                           |  | @Future<br>Date eventDate;                          |
|                           |  |   |
| @Max                      | The value of the field or property must be an integer value lower than or equal to the number in the value element.  |   |
|                           |  | @Max(10)<br>int quantity;                           |
|                           |  |   |

| Constraint/<br>Annotation | Description   | Example   |
|---------------------------|---|---|
| @Min                      | The value of the field or property must be an integer value greater than or equal to the number in the value element.   | @Min(5)<br>int quantity;  |
| @NotNull                  | The value of the field or property must not be null.  | @NotNull<br>String username;                                    |
| @Null                     | The value of the field or property must be null.  | @Null<br>String unusedString;                                   |
| @Past                     | The value of the field or property must be a date in the past.  | @Past<br>Date birthday;   |
| @Pattern                  | The value of the field or property must match the regular expression defined in the regex element.  | @Pattern(regex="^(\\d{3})\\d{3}-\\d{4}")<br>String phoneNumber; |
| @Size                     | The size of the field or property is evaluated and must match the specified boundaries. If the field or property is a String, the size of the string is evaluated. If the field or property is a Collection, the size of the Collection is evaluated. If the field or property is a Map, the size of the Map is evaluated. If the field or property is an array, the size of the array is evaluated. Use one of the optional max or min elements to specify the boundaries. | @Size(min=2, max=240)<br>String briefMessage;                   |

## Note: Validating Null and Empty Strings

The Java programming language distinguishes between null and empty strings. An **empty** string is a string instance of **zero** length, whereas as a **null** string has **no value** at all.

An empty string is represented as **""**. It is a character sequence of zero characters. A **null string** is represented by **null**. It can be described as the absence of a string instance.

## JSR-303 Maven Dependencies

**<!-- JSR-303- Maven Dependencies -->**

```
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>1.1.0.Final</version>
</dependency>
```

**OR**

```
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>1.1.0.GA</version>
</dependency>
```

In the following example, a constraint is placed on a field using the built-in @NotNull constraint:

```
public class Name {

    @NotNull

    private String firstname;

    @NotNull

    private String lastname;

    ...

}
```

You can also place more than one constraint on a single JavaBeans component object. For example, you can place an additional constraint for size of field on the firstname and the lastname fields:

```
public class Name {

    @NotNull

    @Size(min=1, max=16)

    private String firstname;

    @NotNull
```

```
@Size(min=1, max=16)
```

```
private String lastname;
```

```
...
```

```
}
```

## Hibernate Validator Framework (5.3.4.Final)

| Annotation Type                   | Description  |
|-----------------------------------|--|
| <b>ConstraintComposition</b>      | Boolean operator that is applied to all constraints of a composing constraint annotation.                  |
| <b>CreditCardNumber</b>           | The annotated element has to represent a valid credit card number.   |
| <b>CreditCardNumber.List</b>      | Defines several @CreditCardNumber annotations on the same element.   |
| <b>EAN</b>                        | Checks that the annotated character sequence is a valid EAN 13 number.                                     |
| <b>EAN.List</b>                   | Defines several @NotBlank annotations on the same element.   |
| <b>Email</b>                      | The string has to be well-formed email address.  |
| <b>Email.List</b>                 | Defines several @Email annotations on the same element.  |
| <b>Length</b>                     | Validate that the string is between min and max included.  |
| <b>Length.List</b>                | Defines several @Length annotations on the same element.   |
| <b>LuhnCheck</b>                  | Luhn algorithm check constraint.   |
| <b>LuhnCheck.List</b>             | Defines several @LuhnCheck annotations on the same element.  |
| <b>Mod10Check</b>                 | @Modulo 10 check constraint.   |
| <b>Mod10Check.List</b>            | Defines several @Mod10Check annotations on the same element.   |
| <b>Mod11Check</b>                 | Modulo 11 check constraint.  |
| <b>Mod11Check.List</b>            | Defines several @Mod11Check annotations on the same element.   |
| <b>ModCheck</b>                   | Deprecated<br>As of release 5.1.0, replaced by Mod10Check and Mod11Check                                   |
| <b>ModCheck.List</b>              | Defines several @ModCheck annotations on the same element.   |
| <b>NotBlank</b>                   | Validate that the annotated string is not null or empty.   |
| <b>NotBlank.List</b>              | Defines several @NotBlank annotations on the same element.   |
| <b>NotEmpty</b>                   | Asserts that the annotated string, collection, map or array is not null or empty.                          |
| <b>NotEmpty.List</b>              | Defines several @NotEmpty annotations on the same element.   |
| <b>ParameterScriptAssert</b>      | A method-level constraint, that evaluates a script expression against the annotated method or constructor. |
| <b>ParameterScriptAssert.List</b> | Defines several ParameterScriptAssert annotations on the same executable.                                  |
| <b>Range</b>                      | The annotated element has to be in the appropriate range.  |
| <b>Range.List</b>                 | Defines several @Range annotations on the same element.  |

|                          |  |
|--------------------------|--|
| <b>SafeHtml</b>          | Validate a rich text value provided by the user to ensure that it contains no malicious code, such as embedded <code>&lt;script&gt;</code> elements. |
| <b>SafeHtml.List</b>     | Defines several <code>@SafeHtml</code> annotations on the same element.  |
| <b>SafeHtml.Tag</b>      | Allows to specify whitelist tags with specified optional attributes.   |
| <b>ScriptAssert</b>      | A class-level constraint, that evaluates a script expression against the annotated element.  |
| <b>ScriptAssert.List</b> | Defines several <code>@ScriptAssert</code> annotations on the same element.  |
| <b>URL</b>               | Validates the annotated string is an URL.  |
| <b>URL.List</b>          | Defines several <code>@URL</code> annotations on the same element.   |

## Hibernate Validator Framework – Maven Dependencies

`<!-- Hibernate Validator -->`

```

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>5.3.4.Final</version>
</dependency>

```

## Spring Bean validation Application

# Spring Form Validation

|                                       |  |   |
|---------------------------------------|--|---|
| User Name:                            | <input type="text"/>   | <i>Required Field</i>   |
| Gender:                               | <input type="radio"/> Male <input type="radio"/> Female  | <i>Required Field</i>   |
| Hobbies:                              | <input type="checkbox"/> Coding <input type="checkbox"/> Designing <input type="checkbox"/> Testing          | <i>Required Field</i>   |
| Skills:                               | <input type="text" value="J2EE"/> <input type="text" value="SPRING"/> <input type="text" value="HIBERNATE"/> | <i>Required Field</i>   |
| Password:                             | <input type="password"/>   | <i>size must be between 6 and 15</i><br><i>Required Field</i> |
| Address:                              | <input type="text"/>   | <i>Required Field</i>   |
| <input type="button" value="submit"/> |  |   |



## pom.xml

```
<!-- 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>

<!-- JSR-303- Maven Dependencies -->
<dependency>
  <groupId>javax.validation</groupId>
  <artifactId>validation-api</artifactId>
  <version>1.1.0.Final</version>
</dependency>

<!-- Hibernate Validator -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>5.3.4.Final</version>
</dependency>
```

## 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: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/mvc
           http://www.springframework.org/schema/mvc/spring-mvc.xsd
           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">

    <mvc:annotation-driven />
    <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>

</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 javax.validation.constraints.Size;

import org.hibernate.validator.constraints.NotEmpty;
import org.springframework.stereotype.Component;

@Component("user")
public class User {
    @NotEmpty(message = "Required Field")
    private String userName;

    @NotEmpty(message = "Required Field")
    private String gender;

    @NotEmpty(message = "Required Field")
    private String[] hobbies;

    @NotEmpty(message = "Required Field")
    private String[] skills;

    @NotEmpty(message = "Required Field")
    @Size(min=6,max=15)
    private String password;

    @NotEmpty(message = "Required Field")
    private String address;

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

```

**//Getters and Setters**

}

## **UserController.java**

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

```
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.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;
```

```
import com.Biditvats.spring.bean.User;
```

```
@Controller
```

```
public class UserController {
```

```
    @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);
```

```
    }
```

```
}
```