Spring MVC Custom Validation

The Spring MVC framework allows us to perform custom validations. In such case, we declare own annotations. We can perform validation on the basis of own business logic.

Spring MVC Custom Validation Example

In this example, we use both pre-defined annotations as well as custom annotations to validate user input.

Add dependencies to pom.xml file.

**pom.xml**

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

**<dependency>**

**<groupId>**org.springframework**</groupId>**

**<artifactId>**spring-webmvc**</artifactId>**

**<version>**5.1.1.RELEASE**</version>**

**</dependency>**

<!-- https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-jasper -->

**<dependency>**

**<groupId>**org.apache.tomcat**</groupId>**

**<artifactId>**tomcat-jasper**</artifactId>**

**<version>**9.0.12**</version>**

**</dependency>**

    <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

**<dependency>**

**<groupId>**javax.servlet**</groupId>**

**<artifactId>**servlet-api**</artifactId>**

**<version>**3.0-alpha-1**</version>**

**</dependency>**

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->

**<dependency>**

**<groupId>**javax.servlet**</groupId>**

**<artifactId>**jstl**</artifactId>**

**<version>**1.2**</version>**

**</dependency>**

 <!-- https://mvnrepository.com/artifact/org.hibernate.validator/hibernate-validator -->

**<dependency>**

**<groupId>**org.hibernate.validator**</groupId>**

**<artifactId>**hibernate-validator**</artifactId>**

**<version>**6.0.13.Final**</version>**

**</dependency>**

Create the bean class

**Employee.java**

**import** javax.validation.constraints.Max;

**import** javax.validation.constraints.Min;

**import** com.javatpoint.customvalidation.Password;

**public** **class** Employee {

**private** String name;

    //Custom annotation

    @Password

**private** String password;

    //Predefined annotation

    @Min(value=18, message="must be equal or greater than 18")

    @Max(value=45, message="must be equal or less than 45")

**private** **int** age;

**public** String getName() {

**return** name;

    }

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

**this**.name = name;

    }

**public** String getPassword() {

**return** password;

    }

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

**this**.password = password;

    }

**public** **int** getAge() {

**return** age;

    }

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

**this**.age = age;

    }

}

Create the controller class

**EmployeeController.java**

**package** com.javatpoint;

**import** javax.validation.Valid;

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

**import** org.springframework.ui.Model;

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

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

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

@Controller

**public** **class** EmployeeController {

    @RequestMapping("/hello")

**public** String showForm(Model theModel) {

        theModel.addAttribute("emp", **new** Employee());

**return** "viewpage";

    }

    @RequestMapping("/helloagain")

**public** String processForm(

            @Valid @ModelAttribute("emp") Employee emp,

            BindingResult br) {

**if** (br.hasErrors()) {

**return** "viewpage";

        }

**else** {

**return** "final";

        }

    }

}

Create the validator annotation

**Password.java**

**package** com.javatpoint.customvalidation;

**import** java.lang.annotation.ElementType;

**import** java.lang.annotation.Retention;

**import** java.lang.annotation.RetentionPolicy;

**import** java.lang.annotation.Target;

**import** javax.validation.Constraint;

**import** javax.validation.Payload;

@Constraint(validatedBy = PasswordConstraintValidator.**class**)

@Target( { ElementType.METHOD, ElementType.FIELD } )

@Retention(RetentionPolicy.RUNTIME)

**public** **@interface** Password {

    //error message

**public** String message() **default** "must contain jtp";

    //represents group of constraints

**public** Class<?>[] groups() **default** {};

    //represents additional information about annotation

**public** Class<? **extends** Payload>[] payload() **default** {};

}

Create the validator class

**PasswordConstraintValidator.java**

**package** com.javatpoint.customvalidation;

**import** javax.validation.ConstraintValidator;

**import** javax.validation.ConstraintValidatorContext;

**public** **class** PasswordConstraintValidator **implements** ConstraintValidator<Password,String> {

**public** **boolean** isValid(String s, ConstraintValidatorContext cvc) {

**boolean** result=s.contains("jtp");

**return** result;

    }

}

Provide the entry of controller in the web.xml file

**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>**SpringMVC**</display-name>**

**<servlet>**

**<servlet-name>**spring**</servlet-name>**

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

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

**</servlet>**

**<servlet-mapping>**

**<servlet-name>**spring**</servlet-name>**

**<url-pattern>**/**</url-pattern>**

**</servlet-mapping>**

**</web-app>**

Define the bean in the xml file

**spring-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"

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

    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

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

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

    <!-- Provide support for component scanning -->

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

    <!--Provide support for conversion, formatting and validation -->

**<mvc:annotation-driven/>**

    <!-- Define Spring MVC view resolver -->

**<bean** id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>**

**<property** name="prefix" value="/WEB-INF/jsp/"**></property>**

**<property** name="suffix" value=".jsp"**></property>**

**</bean>**

**</beans>**

Create the requested page

**index.jsp**

**<html>**

**<body>**

**<a** href="hello"**>**Click here...**</a>**

**</body>**

**</html>**

Create the other view components

**viewpage.jsp**

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

**<html>**

**<head>**

**<style>**

        .error {color:red}

**</style>**

**</head>**

**<body>**

**<form:form** action="helloagain" modelAttribute="emp"**>**

        Username: **<form:input** path="name" **/>**

**<br><br>**

        Password (\*): **<form:password** path="password" **/>**

**<form:errors** path="password" cssClass="error" **/>**

**<br><br>**

        Age (\*): **<form:input** path="age" **/>**

**<form:errors** path="age" cssClass="error" **/>**

**<br><br>**

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

**</form:form>**

**</body>**

**</html>**

**final.jsp**

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

<!DOCTYPE html**>**

**<html>**

**<body>**

Username: ${emp.name}**<br><br>**

Password: ${emp.password}**<br><br>**

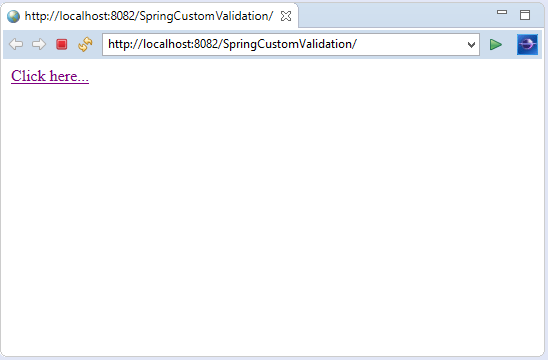
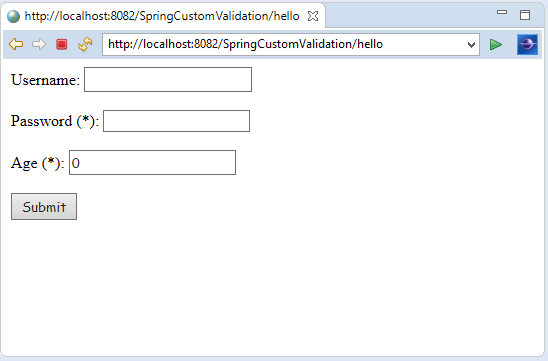
Age: ${emp.age}

**<br><br>**

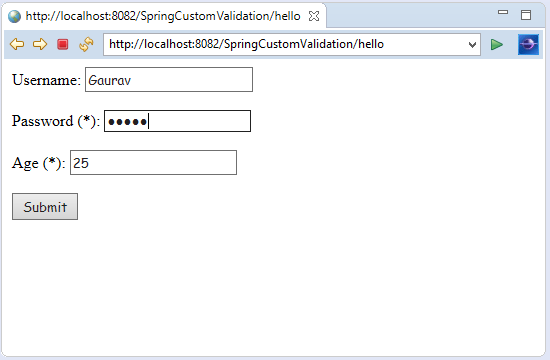
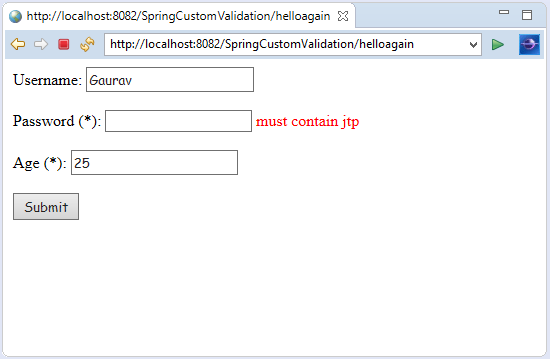
**</body>**

**</html>**

**Output:**

Here, we entered the password without having "jtp" sequence.

Now, we entered the password having "jtp" sequence.

