**17. Spring MVC Form Validation - Creating Custom Validation Rules**

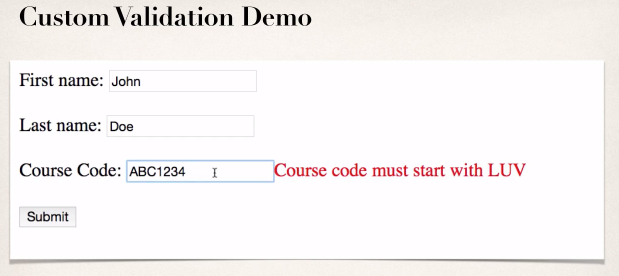
**Spring MVC Custom Validation**:

Generally, when we need to validate user input, Spring MVC offers standard predefined validators.

However, when we need to validate a more particular type input, we have the possibility of creating our own, custom validation logic.

**Custom Validation Demo**:

Here we create a form and, in this form, we have three fields “Fast Name”, “Last Name”, and “Course Code”. The course code must be start with LUV character. If we enter anything else it displays an error message saying “Course Code must start with LUV”



**Custom Validation**:

1. Perform custom validation based on your business rules
2. Our example: Course Code must start with “LUV”
3. Spring MVC calls our custom validation
4. Custom validation returns boolean value for pass/fail (true/false)

**Create a custom Java Annotation … from scratch (Advanced)**:

For custom validation … we will create a Custom Java Annotation **@CourseCode.** This is an advanced Spring MVC work.

@CourseCode(value="LUV", message="must start with LUV")

private String courseCode;

Here, **@CourseCode** is a validation rule that we actually apply to a given field.

**Development Process (Step-by-Step)**:

1. Create custom validation rule
   1. Create **@CourseCode** annotation
   2. Create **CourseCodeConstraintValidator**
2. Add validation rule to Customer class
3. Display error messages on HTML form
4. Update confirmation page

**1) Create custom validation rule**:

**a) Create @CourseCode annotation**:

**package** com.ruhul.odduu.validation;

**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 = CourseCodeConstraintValidator.**class**)

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

@Retention(RetentionPolicy.***RUNTIME***)

**public** **@interface** CourseCode {

// define default course code

**public** String value() **default** "LUV";

// define default error message

**public** String message() **default** "must start with LUV";

// define default group

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

// define default payload

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

}

**b) Create CourseCodeConstraintValidator:**

**File: CourseCodeConstraintValidator.java**:

**package** com.ruhul.odduu.validation;

**import** javax.validation.ConstraintValidator;

**import** javax.validation.ConstraintValidatorContext;

**public** **class** CourseCodeConstraintValidator **implements**

ConstraintValidator<CourseCode, String> {

**private** String coursePrefix;

@Override

**public** **void** initialize(CourseCode theCourseCode) {

coursePrefix = theCourseCode.value();

}

@Override

**public** **boolean** isValid(String theCode, ConstraintValidatorContext theConstraintValidatorContext) {

**boolean** result;

**if** (theCode != **null**) {

result = theCode.startsWith(coursePrefix);

} **else** {

**return** **true**;

}

**return** result;

}

}

**2) Add validation rule to Customer class**:

**package** com.ruhul.odduu.springmvc;

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

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

**import** javax.validation.constraints.NotNull;

**import** javax.validation.constraints.Pattern;

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

**import** com.ruhul.odduu.validation.CourseCode;

**public** **class** Customer {

**private** String firstName;

@NotNull(message = "is required")

@Size(min = 1, message = "is required")

**private** String lastName;

@Min(value = 0, message = "must be greater than or equal to zero")

@Max(value = 10, message = "must be less than or equal to zero")

**private** **int** freePasses;

@Pattern(regexp = "^[a-zA-Z0-9]{5}", message = "Only 5 chars/digits")

**private** String postalCode;

@CourseCode

**private** String courseCode;

**public** String getCourseCode() {

**return** courseCode;

}

**public** **void** setCourseCode(String courseCode) {

**this**.courseCode = courseCode;

}

**public** String getPostalCode() {

**return** postalCode;

}

**public** **void** setPostalCode(String postalCode) {

**this**.postalCode = postalCode;

}

**public** **int** getFreePasses() {

**return** freePasses;

}

**public** **void** setFreePasses(**int** freePasses) {

**this**.freePasses = freePasses;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

}

**3) Display error messages on HTML form**:

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

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Customer Registration Form</title>

<style type=*"text/css"*>

*.error* {

color: *red*;

}

</style>

</head>

<body>

<i><b>Fill out the form. Asterisk (\*) means required</b></i>

<br></br>

<form:form action=*"processForm"* modelAttribute=*"customer"*>

First Name: <form:input path=*"firstName"* />

<br></br>

Last Name (\*): <form:input path=*"lastName"* />

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

<br></br>

Free Passes: <form:input path=*"freePasses"* />

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

<br></br>

Course Code: <form:input path=*"courseCode"* />

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

<br></br>

Postal Code: <form:input path=*"postalCode"* />

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

<br></br>

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

</form:form>

</body>

</html>

**4) Update confirmation page**:

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

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Customer Confirmation</title>

</head>

<body>

The customer is confirmed: ${customer.firstName} ${customer.lastName}

<br></br>

Free Passes: ${customer.freePasses}

<br></br>

Postal Code: ${customer.postalCode}

<br></br>

Course Code: ${customer.courseCode}

</body>

</html>

**Display custom message**:

We can also add custom message for invalid input field. For custom message we have to add the value and message at the top of “***courseCode***” in **Customer.class**

**File: Customer.class:**

**package** com.ruhul.odduu.springmvc;

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

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

**import** javax.validation.constraints.NotNull;

**import** javax.validation.constraints.Pattern;

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

**import** com.ruhul.odduu.validation.CourseCode;

**public** **class** Customer {

**private** String firstName;

@NotNull(message = "is required")

@Size(min = 1, message = "is required")

**private** String lastName;

@Min(value = 0, message = "must be greater than or equal to zero")

@Max(value = 10, message = "must be less than or equal to zero")

**private** **int** freePasses;

@Pattern(regexp = "^[a-zA-Z0-9]{5}", message = "Only 5 chars/digits")

**private** String postalCode;

@CourseCode(value="JAV", message="must start with JAV")

**private** String courseCode;

**public** String getCourseCode() {

**return** courseCode;

}

**public** **void** setCourseCode(String courseCode) {

**this**.courseCode = courseCode;

}

**public** String getPostalCode() {

**return** postalCode;

}

**public** **void** setPostalCode(String postalCode) {

**this**.postalCode = postalCode;

}

**public** **int** getFreePasses() {

**return** freePasses;

}

**public** **void** setFreePasses(**int** freePasses) {

**this**.freePasses = freePasses;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

}

17. Spring MVC Form Validation - Creating Custom Validation Rules