

Spring MVC Validation

Custom Validation



Custom Validation Demo

First name:

Last name:

Course Code:

Submit

Custom Validation

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

Create a custom Java Annotation ... from scratch

Advanced

- ❖ So far, we've used predefined validation rules: @Min, @Max, ...
- ❖ For custom validation ... we will create a **Custom Java Annotation**
 - ❖ **@CourseCode**

```
@CourseCode(value="LUV", message="must start with LUV")
private String courseCode;
```

Development Process

Step-By-Step

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

Development Process - Drill Down

1. Create custom validation rule
 - a. Create **@CourseCode** annotation
 - b. Create **CourseCodeConstraintValidator**

Step-By-Step

Step 1a: Create @CourseCode annotation

Advanced

Usage Example

```
@CourseCode(value="LUV", message="must start with LUV")
private String courseCode;
```

Step 1a: Create @CourseCode annotation

Advanced

```
@Constraint(validatedBy = CourseCodeConstraintValidator.class)
@Target( { ElementType.METHOD, ElementType.FIELD } )
@Retention(RetentionPolicy.RUNTIME)
public @interface CourseCode {

    ...
}
```

```
@CourseCode(value="LUV", message="must start with LUV")
private String courseCode;
```

Step 1a: Create @CourseCode annotation

Advanced

```
@Constraint(validatedBy = CourseCodeConstraintValidator.class)
@Target( { ElementType.METHOD, ElementType.FIELD } )
@Retention(RetentionPolicy.RUNTIME)
public @interface CourseCode {

    // define default course code
    public String value() default "LUV";
}

...
```

```
@CourseCode(value="LUV", message="must start with LUV")
private String courseCode;
```

Step 1a: Create @CourseCode annotation

Advanced

```
@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";

    ...
}
```

```
@CourseCode(value="LUV", message="must start with LUV")
private String courseCode;
```

Advanced

Step 1b: Create CourseCodeConstraintValidator

```
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 {
            result = true;
        }

        return result;
    }
}
```