FAQ: Spring MVC Custom Validation - Possible to validate with multiple strings?

Spring MVC Custom Validation - FAQ: Is it possible to integrate multiple validation string in one annotation?

Question:

Is it possible to integrate multiple validation string in one annotation? For example, validate against both LUV and TOPS.

Answer:

Yes, you can do this. In your validation, you will make use of an array of strings.

Here's an overview of the steps.

1. Update CourseCode.java to use an array of strings

2. Update CourseCodeConstraintValidator.java to validate against array of strings

3. Update Customer.java to validate using array of strings

Detailed Steps

1. Update CourseCode.java to use an array of strings

Change the value entry to an array of Strings:

```
1.      // define default course code
2.      public String[] value() default {"LUV"};
```

Note the use of square brackets for the array of Strings. Also, the initialized value uses curley-braces for array data.

2. Update CourseCodeConstraintValidator.java to validate against array of strings

Change the field for coursePrefixes to an array

```
private String[] coursePrefixes;
```

Update the isValid(...) method to loop through the course prefixes. In the loop, check to see if the code matches any of the course prefixes.

```
1.      @Override
2.      public boolean isValid(String theCode,
3.                             ConstraintValidatorContext theConstraintValidatorContext) {
4.          boolean result = false;
5.
6.          if (theCode != null) {
7.
8.              //
9.              // loop thru course prefixes
10.             //
11.             // check to see if code matches any of the course prefixes
12.             //
13.             for (String tempPrefix : coursePrefixes) {
14.                 result = theCode.startsWith(tempPrefix);
15.
16.                 // if we found a match then break out of the loop
17.                 if (result) {
18.                     break;
19.                 }
20.             }
21.         }
22.         else {
23.             result = true;
24.         }
25.
26.         return result;
27.     }
```

3. Update Customer.java to validate using array of strings

```
1.      @CourseCode(value={"TOPS", "LUV"}, message="must start with TOPS or LUV")
2.      private String courseCode;
```

Note the use of curley braces.

Complete Source Code:

https://gist.github.com/darbyluv2code/0275ddb6e70e085a10fd464e36a42739

That's it. This provides a solution to integrate multiple validation string in one annotation. In this example, the code validates against both LUV and TOPS.