

# Spring Validation

# The Need for Validation

- Check the user input form for
  - Required Fields
  - Valid numbers in range
  - Valid format (Postal Code)
  - Custom business rule

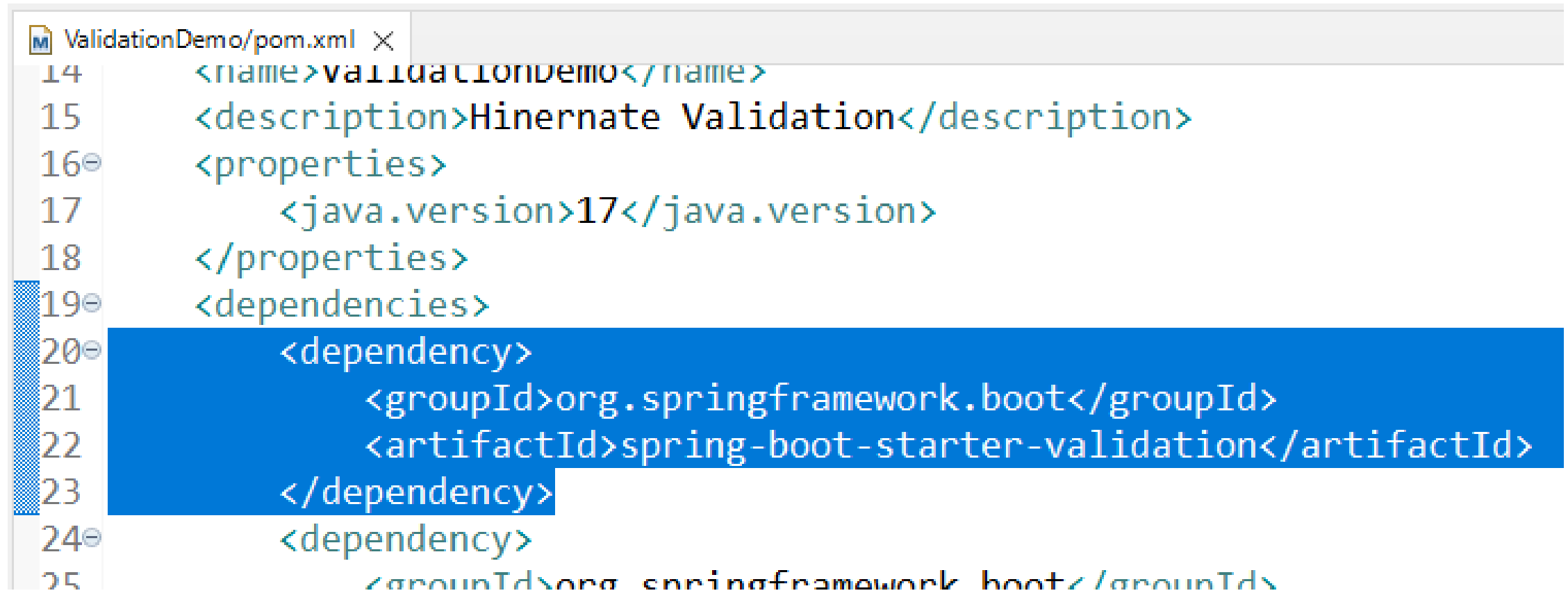
# Spring and Validation

- Java has a standard Bean Validation API
- Spring version 4 and higher supports Bean Validation API
- Preferred method of validation for Spring Apps
- Simply add validation jars to our project

# Validation Annotations

Annotation	Description
<b>@NotNull</b>	Checks that the annotated value is not null
<b>@Min</b>	Must be a number $\geq$ value
<b>@Max</b>	Must be a number $\leq$ value
<b>@Size</b>	Size must match the given size
<b>@Pattern</b>	Must match a regular expression pattern
<b>@Future / @Past</b>	Date must be in future or past of given date
<b><i>others ...</i></b>	

# Add Validation Dependency



The screenshot shows a code editor window titled "ValidationDemo/pom.xml". The XML content is as follows:

```
14     <name>validationDemo</name>
15     <description>Hinernate Validation</description>
16     <properties>
17         <java.version>17</java.version>
18     </properties>
19     <dependencies>
20         <dependency>
21             <groupId>org.springframework.boot</groupId>
22             <artifactId>spring-boot-starter-validation</artifactId>
23         </dependency>
24         <dependency>
25             <groupId>org.springframework.boot</groupId>
```

Lines 20 through 23 are highlighted in blue, indicating the newly added dependency for Spring Boot Starter Validation.

# Model Class with Annotation

```
Student.java x
1 package com.rit.demo;
2
3 import javax.validation.constraints.Max;
4
5
6
7 public class Student {
8
9     @Min(100)
10    @Max(999)
11    private int studId;
12
13    @Size(min=3, max=10)
14    private String name;
15
16    //constructors
17    //getters & setters
18    //toString
19 }
```

# Controller with @Valid (for Validation)

StudentController.java ×

```
1 package com.rit.demo;
2
3 import javax.validation.Valid;
4
5
6
7
8
9
10 @RestController
11 @RequestMapping("stud")
12 public class StudentController {
13
14     @PostMapping
15     public Student saveStudent(@Valid @RequestBody Student student) {
16         return student;
17     }
18 }
```

# Valid Inputs

POST localhost:9001/stud

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {  
2   .... "studId": 103,  
3   .... "name": "Anand"  
4 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "studId": 103,  
3   "name": "Anand"  
4 }
```



# Invalid studId

```
{  
  .... "studId": 13,  
  .... "name": "Anand"  
}
```



400 Bad Request

44 ms

5.92 KB

[Save Response](#)

Raw

Preview

Visualize

JSON



```
"error": "Bad Request",  
"trace": "org.springframework.web.bind.MethodArgumentNotValidException:  
  Validation failed for argument [0] in public com.rit.demo.Student com.rit.  
  demo.StudentController.saveStudent(com.rit.demo.Student): [Field error in  
  object 'student' on field 'studId': rejected value [13]; codes [Min.student.  
  studId,Min.studId,Min.int,Min]; arguments [org.springframework.context.  
  support.DefaultMessageSourceResolvable: codes [student.studId,studId];
```

# Invalid Name

```
{  
  .... "studId": 103,  
  .... "name": "Ra"  
}
```



400 Bad Request

13 ms

5.92 KB

[Save Response](#)

Raw

Preview

Visualize

JSON



```
"error": "Bad Request",  
"trace": "org.springframework.web.bind.MethodArgumentNotValidException:  
  Validation failed for argument [0] in public com.rit.demo.Student com.rit.  
  demo.StudentController.saveStudent(com.rit.demo.Student): [Field error in  
  object 'student' on field 'name': rejected value [Ra]; codes [Size.student.  
  name,Size.name,Size.java.lang.String,Size]; arguments [org.springframework.  
  context.support.DefaultMessageSourceResolvable: codes [student.name,name];
```

# Adding Custom Message with Validation

```
Student.java x
6
7 public class Student {
8
9     @Min(value=100, message="StudId must be greater than 100" )
10    @Max(value=999, message="StudId must be less than 1000")
11    private int studId;
12
13    @Size(min=3, max=10, message="Name must be between 3 and 10 characters")
14    private String name;
15
16    //constructors
17    //getters & setters
18    //toString
19
```

# To display the custom message we need a controller advice class

```
@ControllerAdvice
@Component
public class CustomExceptionHandler extends ResponseEntityExceptionHandler {
    @Override
    protected ResponseEntity<Object> handleMethodArgumentNotValid(MethodArgumentNotValidException ex, HttpHeaders headers,
        HttpStatus status, WebRequest request) {

        Map<String, List<String>> body = new HashMap<>();

        List<String> errors = ex.getBindingResult().getFieldErrors().stream()
            .map(DefaultMessageSourceResolvable::getDefaultMessage) .collect(Collectors.toList());

        body.put("errors", errors);

        return new ResponseEntity<>(body, HttpStatus.BAD_REQUEST);
    }
}
```

```
3+ import java.util.HashMap;
17
18 @ControllerAdvice
19 @Component
20 public class CustomExceptionHandler extends ResponseEntityExceptionHandler {
21     @Override
22     protected ResponseEntity<Object> handleMethodArgumentNotValid(
23         MethodArgumentNotValidException ex, HttpHeaders headers,
24         HttpStatus status, WebRequest request) {
25         Map<String, List<String>> body = new HashMap<>();
26
27         List<String> errors = ex.getBindingResult()
28             .getFieldErrors()
29             .stream()
30             .map(DefaultMessageSourceResolvable::getDefaultMessage)
31             .collect(Collectors.toList());
32
33         body.put("errors", errors);
34
35         return new ResponseEntity<>(body, HttpStatus.BAD_REQUEST);
36     }
37 }
```

# Custom Message Returned

```
{  
  .... "studId": 13,  
  .... "name": "Ra"  
}
```



400 Bad Request

Raw

Preview

Visualize

JSON



```
{  
  "errors": [  
    "StudId must be greater than 100",  
    "Name must be between 3 and 10 characters"  
  ]  
}
```

@NotNull,  
@NotBlank  
&  
@NotEmpty

Student.java X

```
2
3+ import javax.validation.constraints.NotBlank;
6
7 public class Student {
8
9@   @NotNull(message="FirstName cannot be null")
10   private String firstName;
11
12@   @NotBlank(message="MiddleName must have at least 1 character")
13   private String middleName;
14
15@   @NotEmpty(message="LastName cannot be empty")
16   private String lastName;
17
```



```
{  
  .... "firstName": null,  
  .... "middleName": "",  
  .... "lastName": ""  
}
```



400 Bad F



Raw

Preview

Visualize

JSON



```
{
```

```
  "errors": [  
    "FirstName cannot be null",  
    "MiddleName must have atleast 1 character",  
    "LastName cannot be empty"  
  ]  
}
```

```
{
  .... "firstName": "",
  .... "middleName": "",
  .... "lastName": ""
}
```



400 Bad Request



Raw

Preview

Visualize

JSON



```
{
  "errors": [
    "MiddleName must have atleast 1 character",
    "LastName cannot be empty"
  ]
}
```

```
{
  .... "firstName": ". ",
  .... "middleName": ". ",
  .... "lastName": ". "
}
```



400 Bad Request

y

Raw

Preview

Visualize

JSON



```
{
```

```
  "errors": [
```

```
    "MiddleName must have atleast 1 character"
```

```
  ]
```

```
}
```

```
{
  .... "firstName": "a",
  .... "middleName": "b",
  .... "lastName": "c"
}
```



y

Raw

Preview

Visualize

JSON

```
{
  "firstName": "a",
  "middleName": "b",
  "lastName": "c"
}
```

# More Annotations



- @Email
- @Positive
- @PositiveOrZero
- @Negative
- @NegativeOrZero
- @Past
- @PastOrPresent
- @Future
- @FutureOrPresent

Student.java ×

```
9 import com.fasterxml.jackson.annotation.JsonFormat;
10
11 public class Student {
12
13     @Past(message="DateOfBirth must be a Past Date")
14     @JsonFormat(pattern = "yyyy/MM/dd")
15     private Date dateOfBirth;
16
17     @PastOrPresent(message="BookBorrowedDate must be a Past or Present Date")
18     @JsonFormat(pattern = "yyyy/MM/dd")
19     private Date bookBorrowedDate;
20
21     @Future(message = "BookReturnDate must be a future Date")
22     @JsonFormat(pattern = "yyyy/MM/dd")
23     private Date bookReturnDate;
```

```
{
  .... "dateOfBirth": "2000/05/25",
  .... "bookBorrowedDate": "2000/05/25",
  .... "bookReturnDate": "2024/05/25"
}
```



/ Raw Preview Visualize JSON  

```
{
  "dateOfBirth": "2000/05/25",
  "bookBorrowedDate": "2000/05/25",
  "bookReturnDate": "2024/05/25"
}
```

```
{
  .... "dateOfBirth": "2100/05/25",
  .... "bookBorrowedDate": "2000/05/25",
  .... "bookReturnDate": "1000/05/25"
}
```



400 Bad Requ

Raw

Preview

Visualize

JSON



```
{
  "errors": [
    "BookReturnDate must be a future Date",
    "DateOfBirth must be a Past Date"
  ]
}
```



# `${validatedValue}` interpolation

```
Student.java X
1
2
3 import javax.validation.constraints.Size;
4
5 public class Student {
6
7     @Size(min=5,
8           max=10,
9           message="name : '${validatedValue}' must be "
10                + "between {min} and {max} characters")
11     private String name;
```

```
{  
  .... "name": "Ram"  
}
```



400 Bad Request 21

ty

Raw

Preview

Visualize

JSON



```
{
```

```
  "errors": [  
    "name : 'Ram' must be between 5 and 10 characters"  
  ]
```

```
}
```

# Regular Expression

[abc] Find one character from the options between the brackets

[^abc] Find one character NOT between the brackets

[0-9] Find one character from the range 0 to 9

| Find a match for any one separated by | as in: cat|dog|fish

. Find just one instance of any character

^ Finds a match as the beginning of a string as in: ^Hello

\$ Finds a match at the end of the string as in: World\$

\d Find a digit

\s Find a whitespace character

# Regular Expression

$n^+$  Matches any string that contains at least one  $n$

$n^*$  Matches any string that contains zero or more occurrences of  $n$

$n^?$  Matches any string that contains zero or one occurrences of  $n$

@Email

@Email(regex = ".\*@gmail.com\$")

@Pattern(regex = "^\\d{10}\$")

```
@NotBlank @Size(min = 3, max = 8)
private String name;
@NotNull @Min(0) @Max(100)
private int age;
@NotNull @PositiveOrZero
private int credits;
@Digits(integer = 5, fraction = 2)
private double balance;
@Past
private LocalDate dob;
@email(regex = ".*@gmail.com$")
private String email;
@Pattern(regex = "^\\d{10}$")
private String phone;
@AssertTrue
private boolean active;
```

```
{
  ... "name": "Ra",
  ... "age": 300,
  ... "credits": -120,
  ... "balance": 2300.1234,
  ... "dob": "2032-11-21",
  ... "email": "abc@yahoo.com",
  ... "phone": "95001004567",
  ... "active": false
}
```

[org.springframework.web.bind.MethodArgumentNotValidException: Validation failed for argument [0] in public com.valid.demo.request.Customer com.valid.demo.controller.CustomerController.addCustomer (com.valid.demo.request.Customer) with 8 errors: [Field error in object 'customer' on field 'email': rejected value [abc@yahoo.com]; codes

```
{
  .... "name": "Ra",
  .... "age": 300,
  .... "credits": -120,
  .... "balance": 2300.1234,
  .... "dob": "2032-11-21",
  .... "email": "abc@yahoo.com",
  .... "phone": "95001004567",
  .... "active": false
}
```

l.email,Email.java.lang.String,Email]; arguments  
kt.support.DefaultMessageSourceResolvable: codes  
guments []; default message [email],  
aints.Pattern\$Flag;@defbb0a,.\*@gmail.com\$];  
well-formed email address]] [Field error in  
'credits': rejected value [-120]; codes  
redits,PositiveOrZero.credits,PositiveOrZero.int,  
kt.support.DefaultMessageSourceResolvable: codes  
; arguments []; default message [credits]];  
reater than or equal to 0]] [Field error in

```
com.valid.demo.controller.CustomerController.addCustomer  
(com.valid.demo.request.Customer) with 3 errors: [Field  
error in object 'customer' on field 'isActive': rejected
```

```
{  
  .... "name": "Raju",  
  .... "age": 30,  
  .... "credits": 120,  
  .... "balance": 2300.12,  
  .... "dob": "2002-11-21",  
  .... "email": "abc@yahoo.com",  
  .... "phone": "95001004567",  
  .... "active": false  
}
```

```
[Pattern.customer.phone, Pattern.phone, Pattern.java.lang.  
String, Pattern]; arguments
```

POST



http://localhost:8080/customer

Body



raw



JSON



Beautify

```
1 {
2   ... "name": "Raju",
3   ... "age": 30,
4   ... "credits": 120,
5   ... "balance": 2300.12,
6   ... "dob": "2002-11-21",
7   ... "email": "abc@gmail.com",
8   ... "phone": "9500100456",
9   ... "active": true
10 }
```

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

```
1 {
2   "name": "Raju",
3   "age": 30,
4   "credits": 120,
5   "balance": 2300.12,
6   "dob": "2002-11-21",
7   "email": "abc@gmail.com",
8   "phone": "9500100456",
9   "active": true
10 }
```



```
1 package com.valid.demo.exception;
2 import java.util.HashMap;
3
4 @ControllerAdvice
5 public class GlobalExceptionHandler {
6
7     @ExceptionHandler(MethodArgumentNotValidException.class)
8     public ResponseEntity<Object> handleValidationExceptions(
9         MethodArgumentNotValidException ex) {
10         Map<String, String> errors = new HashMap<>();
11         ex.getBindingResult().getAllErrors().forEach(error -> {
12             String fieldName = ((FieldError) error).getField();
13             String errorMessage = error.getDefaultMessage();
14             errors.put(fieldName, errorMessage);
15         });
16         return new ResponseEntity<>(errors, HttpStatus.BAD_REQUEST);
17     }
18 }
```

POST

http://localhost:8080/customer

Send

Body

...

raw

JSON

Beautify

```
1 {  
2   "name": "Ra",  
3   "age": 300,  
4   "credits": -120,  
5   "balance": 2300.1234,  
6   "dob": "2032-11-21",  
7   "email": "abc@yahoo.com",  
8   "phone": "95001004567",  
9   "active": false  
10 }
```

Body



400 Bad Request

561 ms

478 B

Save Response

Pretty

JSON



```
1 {  
2   "balance": "numeric value out of bounds (<5 digits>.<  
3   "credits": "must be greater than or equal to 0",  
4   "phone": "must match \"^\\d{10}$\"",  
5   "dob": "must be a past date",  
6   "name": "size must be between 3 and 8",  
7   "active": "must be true",  
8   "email": "must be a well-formed email address",  
9   "age": "must be less than or equal to 100"  
10 }
```

Thank you