

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
<div>Spring Boot<ul style="list-style-type: none"><li>Introduction to Spring Boot</li><li>Multi-Tier Architecture</li><li>Beans</li><li>Bean Scopes</li><li>Bean Validation</li><li>Dependency Injection</li><li>Components</li><li>Configuration</li><li>Connecting to a Database</li><li>Entities</li><li>Postman</li><li>Controllers</li><li>Services</li><li>Repositories</li><li>Custom Queries</li><li>Data Transfer Objects</li><li>Lombok</li><li>Custom Exceptions</li><li>Swagger</li><li>Profiles</li><li>Pre-Populating Databases for Testing</li><li>Unit testing with Mockito</li></ul></div>

# Bean Validation

## Contents

- Overview
  - @Min
  - @Max
  - @Size
  - @NotNull
  - @Pattern
- Tutorial
- Exercises

## Overview

Spring allows for the *validation* of bean fields simply by using annotations.

These annotations might be used to avoid implementing validation logic within each layer of an application, which is time-consuming and error-prone.

Annotations must be declared in the data modelling class (as these classes are used throughout the application), so that data can be carried from one layer to the next.

**Field-level** annotations are applied to *attributes*, while **property-level** annotations are applied to *methods*. It is recommended to stick to one or the other within one class.

*note: it is **NOT** recommended to annotate a field and the accompanying getter methods, as this would cause the field to be validated twice.*

### @Min

Sets the minimum value of a number field.

```
@Min(0)
private int age;
```

### @Max

Sets the maximum value of a number field.

```
@Min(0)
@Max(110)
private int age;
```

### @Size

Restricts the size of a text field to a set range.

```
@Size(min = 2, max = 12)
private String username;
```

### @NotNull

Prevents a field from being set to **null**.

```
@Size(min = 2, max = 12)
@NotNull
private String username;
```

### @Pattern

<div><div></div><div>Testing</div></div>
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

Checks the field against a given RegEx pattern.

```
@Pattern("^[a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})$")
private String email;
```

## Tutorial

There is no tutorial for this module.

## Exercises

1. Create a **profile** class.
2. Create the following attributes: **forename**, **surname**, **dateOfBirth**, **gender**, and **ethnicity**.
3. Set a min and max constraint on the **forename** and **surname** fields.
4. Use the appropriate validation to ensure the **ethnicity** field is not blank.
5. Use a RegEx pattern to check the format of the **dateOfBirth** field.