Residência em Software

Módulo Programação JAVA (Avançado)

MÊS 01

INSTITUIÇÃO EXECUTORA

cepedi UESC

COORDENADORA

MCTI FUTURO Softex

APOIO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Residência em Software

Validando Aplicações com **Bean Validation**

Java Specification Requests (JSR 380)

Visão geral:
Validar a **entrada do usuário** é um requisito super comum na maioria dos aplicativos, e a estrutura **Java Bean Validation** se tornou o **padrão** para lidar com esse tipo de lógica.

JSR 380, também conhecido como **Bean Validation**, é um **padrão Java** usado para realizar **validação em aplicativos** Java.

Ele garante que as propriedades de um bean Java **atendam a critérios específicos** aplicando **restrições usando anotações** como @NotNull, @Min e @Max.

Residência em Software

Validando Aplicações com **Bean Validation**

Alguns pontos-chave sobre JSR 380:

- A **Validação de Bean** é comumente usada para validar a entrada do usuário em aplicativos.
- Ele se baseia nos recursos da **API Bean Validation** introduzida no Java EE 7.
- A versão mais recente, Jakarta Bean Validation 3.0, faz parte do **Jakarta EE** e **JavaSE**.
- Requer Java 17 ou superior e usa Spring Boot 3.x, que inclui Hibernate Validator 8.0.0.

Residência em Software

Validando Aplicações com **Bean Validation**

Algumas anotações usadas no padrão JSR:

@NotNull valida que o valor da propriedade anotada não é nulo.
@AssertTrue valida se o valor da propriedade anotada é true.
@Size valida se o valor da propriedade tem um tamanho entre os atributos min e max. Podemos aplicá-lo às propriedades String, Collection, Map e array.
@Min valida se a propriedade anotada tem um valor não menor que o atributo value.
@Max valida se a propriedade anotada tem um valor não maior que o atributo value.
@Email valida que a propriedade anotada é um endereço de email válido.

Residência em Software

Validando Aplicações com **Bean Validation**

Algumas anotações usadas no padrão JSR:

@Pattern: para dizer que um campo de cadeia de caracteres só é válido quando corresponde a uma determinada expressão regular.

Algumas anotações aceitam **atributos adicionais**, mas o atributo **message** é comum a todos eles. Essa é a mensagem que geralmente será renderizada quando o valor da respectiva propriedade **falhar na validação**.

Residência em Software

Validando Aplicações com **Bean Validation**

Anotações adicionais que podemos encontrar no JSR:

@NotEmpty valida que a propriedade não é nula ou vazia. Podemos aplicá-lo aos valores String, Collection, Map ou Array.
@NotBlank pode ser aplicado somente a valores de texto e valida que a propriedade não é nula ou espaço em branco.
@Positive e **@PositiveOrZero** se aplicam a valores numéricos e validam que eles são estritamente positivos ou positivos, incluindo 0.
@Negative e **@NegativeOrZero** se aplicam a valores numéricos e validam que eles são estritamente negativos, ou negativos, incluindo 0.
@Past e **@PastOrPresent** valida que um valor de data está no passado ou **no passado, incluindo o presente**. Podemos aplicá-lo a tipos de **data**.
@Future e **@FutureOrPresent** valida que um valor de data está no futuro ou no futuro, incluindo o presente.

Validando Aplicações com **Bean Validation**

Declarando restrições:

As restrições na validação do Jakarta Bean são expressas por meio de anotações Java. Existem quatro tipos principais de restrições:

- restrições de campo;
- restrições de propriedade;
- restrições de elemento de contêiner;
- restrições de classe.

Nem todas as restrições podem ser colocadas em todos esses níveis.
Nenhuma das restrições padrão definido pode ser colocada no nível de classe.
A anotação determina em quais elementos uma restrição pode ser colocada (na própria anotação de restrição).
Por fim, também temos a alternativa de criar **restrições personalizadas**.

Residência em Software

Validando Aplicações com **Bean Validation**

Instalação Maven: Arquivo: pom.xml

```
<dependency>
    <groupId>org.hibernate.validator</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>8.0.1.Final</version>
</dependency>
```

Validando Aplicações com **Bean Validation**

Arquivo: build.gradle

```
implementation group: 'org.hibernate', name: 'hibernate-validator', version: '8.0.1.Final'
```

Residência em Software

Validando Aplicações com **Bean Validation**

Usando anotações de validação:

```
import jakarta.validation.constraints.AssertTrue;
import jakarta.validation.constraints.Max;
import jakarta.validation.constraints.Min;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import jakarta.validation.constraints.Email;
```

Residência em Software

Validando Aplicações com **Bean Validation**

Usando anotações de validação:

```java
@NotNull(message = "Name cannot be null")
private String name;

@AssertTrue
private boolean isRegistered;

@Size(min = 10, max = 200, message = "About Me must be between 10 and 200 characters")
private String aboutMe;

@Min(value = 18, message = "Score should not be less than 18")
@Max(value = 150, message = "Score should not be greater than 150")
private int score;

@Email(message = "Email should be valid")
private String email;

```

Residência em Software

Validando Aplicações com **Bean Validation**

Usando anotações de validação:

https://gist.github.com/rog3r/49c2d594f2e160b15b9ca895bde133d5

https://gist.github.com/rog3r/d1d608abd43c5ca76ee75b477cae2e98

Residência em Software

PP007

Orientações

- Implemente a validação de Beans nos seus models;
- valide os atributos com as anotações pertinentes.

Residência em Software

Contato

rogerio.jesus@cepedi.org.br

https://moodle.residenciatic18.cepedi.org.br/