

Bean Validation

Validation FIRST

Fernando Teixeira Alves de Araujo

Introdução

Bean Validation é uma especificação do Java que define um modelo de validação baseado em anotações para validar dados de objetos de forma declarativa. É parte do JSR 380 (Bean Validation 2.0) e permite a integração com frameworks como JavaServer Faces (JSF) e JPA.

Principais Características

Anotações de Validação: Utiliza anotações como `@NotNull`, `@Size`, `@Min`, `@Max`, entre outras, para definir regras de validação diretamente nos atributos das classes.

Validador Personalizado: Permite a criação de validadores personalizados para atender a requisitos específicos.

Integração: Pode ser facilmente integrado com diversos frameworks e tecnologias Java, como Spring, Hibernate, JSF, JPA, etc.

Internacionalização: Suporta mensagens de validação internacionalizadas através de arquivos de mensagens.

Benefícios

Facilidade de Uso: As anotações tornam a validação de dados simples e intuitiva.

Manutenção Centralizada: As regras de validação são centralizadas nas classes de modelo, facilitando a manutenção.

Reutilização: As regras de validação podem ser reutilizadas em diferentes camadas da aplicação.

Confiabilidade: Garante que os dados atendem aos critérios de validação antes de serem processados ou armazenados.

Annotations

@NotNull: Garante que o valor do campo anotado não seja nulo.

@Null: Garante que o valor do campo anotado seja nulo.

@AssertTrue: Garante que o valor do campo anotado seja verdadeiro.

@AssertFalse: Garante que o valor do campo anotado seja falso.

@Min(valor): Garante que o valor numérico do campo anotado seja no mínimo o valor especificado.

@Max(valor): Garante que o valor numérico do campo anotado seja no máximo o valor especificado.

@DecimalMin(valor): Garante que o valor decimal do campo anotado seja no mínimo o valor especificado.

@DecimalMax(valor): Garante que o valor decimal do campo anotado seja no máximo o valor especificado.

Annotations

@Size(min, max): Garante que o tamanho do campo anotado (como uma string, coleção, mapa ou matriz) esteja dentro dos limites especificados.

@Digits(integer, fraction): Garante que o valor numérico do campo anotado tenha no máximo o número especificado de dígitos inteiros e fracionários.

@Past: Garante que a data anotada seja no passado.

@PastOrPresent: Garante que a data anotada seja no passado ou presente.

@Future: Garante que a data anotada seja no futuro.

@FutureOrPresent: Garante que a data anotada seja no futuro ou presente.

@Pattern(regex): Garante que o valor do campo anotado corresponda à expressão regular especificada.

Annotations

@Email: Garante que o valor do campo anotado seja um endereço de email válido.

@Positive: Garante que o valor numérico do campo anotado seja positivo.

@PositiveOrZero: Garante que o valor numérico do campo anotado seja positivo ou zero.

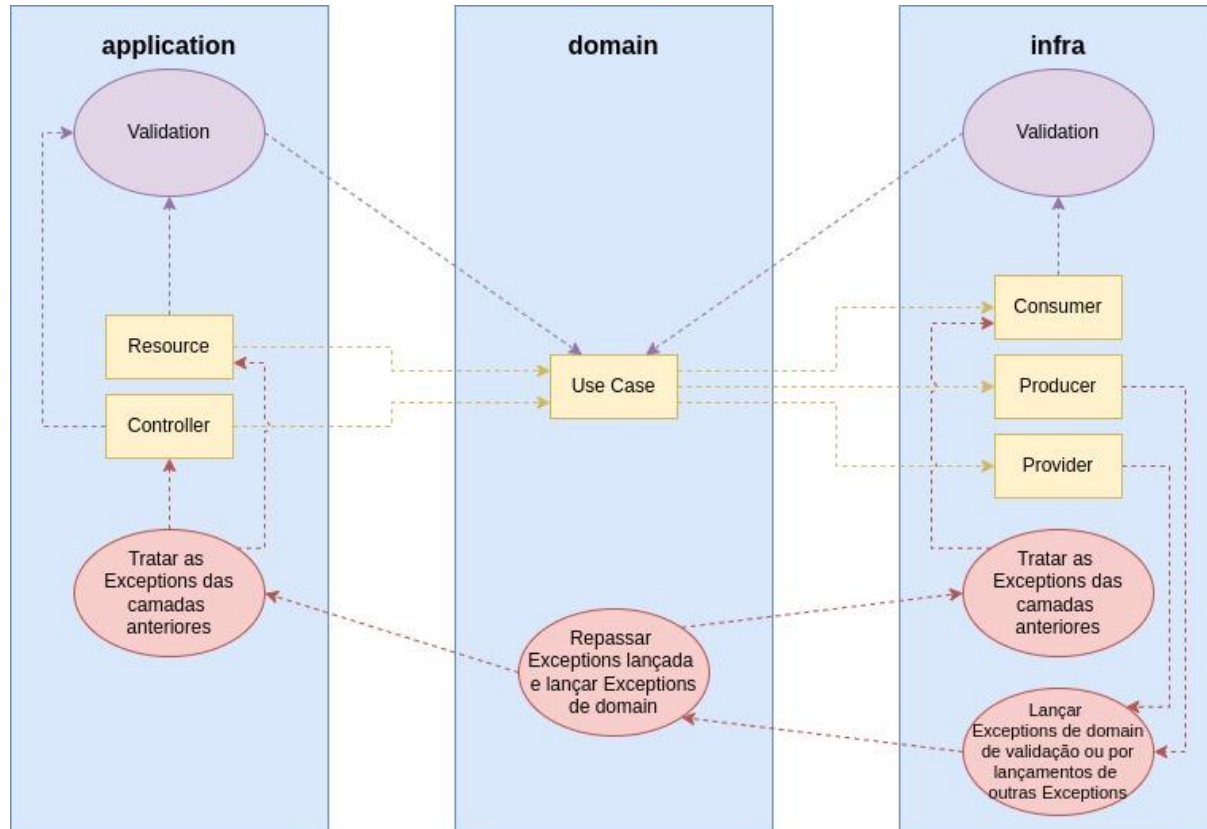
@Negative: Garante que o valor numérico do campo anotado seja negativo.

@NegativeOrZero: Garante que o valor numérico do campo anotado seja negativo ou zero.

@NotBlank: Garante que o valor do campo anotado (string) não seja nulo e não esteja em branco (contendo apenas espaços).

@NotEmpty: Garante que o valor do campo anotado (string, coleção, mapa ou matriz) não seja nulo e não esteja vazio.

Visão nas camadas



HandsOn

Validação em atributo de objeto;

Validação em parâmetro;

Validação em objeto complexo;

Validação em em método;