

Logo da Orange Talents

# Implementando Validação customizada com Micronaut Validation

## Cenário:

Imagine que temos um sistema na empresa na qual possui uma API REST que permite clientes solicitarem um ou mais cartões de crédito. Um dos passos para solicitar um cartão de crédito é informar os dados do cliente solicitante, como nome, email, CPF, data de nascimento entre outras. Para isso, um desenvolvedor(a) do time modelou o DTO de entrada da aplicação como abaixo:

@Introspected  
data class ClienteRequest(  
 @field:NotBlank @field:Size(max=120)  
 val nome: String,   
 @field:Email  
 val email: String,   
 @field:NotNull @field:Past  
 val dataDeNascimento: LocalDate,   
 // outros atributos  
)

Mas ele(a) está com dificuldades de implementar a lógica de validação para permitir somente clientes maiores de idade. Ele(a) está tentando implementar uma validação customizada da Bean Validation com Micronaut mas sem muito sucesso.

Por esse motivo ele(a) te pediu ajuda.

Então, com base no DTO acima, como você implementaria uma validação customizada da Bean Validation com Micronaut para verificar se um cliente é maior de idade ou não?

## O que seria bom ver nessa resposta?

* **Peso 7**: Implementar uma anotação própria e um validador customizado respeitando o contrato da API do Micronaut Validation;
* **Peso 2**: Explicar a lógica implementada no método isValid() para validar a maior idade do cliente;
* **Peso 1**: Assumir no método isValid() que o atributo está válido quando seu valor for null, pois essa é uma responsabilidade da anotação @NotNull;

## Resposta do Especialista:

* Crio uma anotação @MaiorDeIdade e anoto o atributo dataDeNascimento do DTO. Nessa anotação eu adiciono as anotações básicas (@Target, @Retention etc), também adiciono a anotação @Constraint indicando o meu validador customizado, e adiciono o atributo message com a mensagem de erro "deve ser maior de idade";
* Implemento a classe MaiorDeIdadeValidator respeitando o contrato da interface do Micronaut Validation (o framework sugere usar uma interface dele em vez da Bean Validation). Então implemento o método isValid() com a lógica para calcular a idade do cliente com base na data de nascimento informada no DTO, se a idade for maior ou igual que 18 eu retorno true, caso contrário false;
* Ainda no validador customizado, no inicio do método isValid(), eu verifico se o atributo é null, pois se ele não for preenchido eu considero o atributo válido (retorno true) pois meu validador não é responsável pela obrigatoriedade do preenchimento do atributo, mas sim a anotação @NotNull;
* Por fim, via POSTman ou Insomnia eu testo a API REST para ter certeza que a validação está funcionando. Caso não esteja, eu consulto a documentação oficial do Micronaut Validation para me ajudar;