

Desafio Back-end Viasoft

Antes de mais nada, agradecemos pelo seu interesse em fazer parte da Viasoft! A seguir, você encontrará todas as informações necessárias para começar o seu teste.

No dia da entrevista técnica

Na data agendada para a entrevista técnica, por favor, tenha sua aplicação rodando localmente em sua máquina para podermos executar os testes e discutir os pontos desenvolvidos, bem como possíveis dúvidas. Realizaremos um code review juntos. Você terá a oportunidade de explicar seu raciocínio, como estruturou a aplicação e como o projeto pode ser aprimorado.

Ambiente de desenvolvimento

Java 17, Maven e a lib **Spring Web**. Pode ser utilizado outras libs em conjunto.

Objetivo

Imagine **(sem realizar a integração)** que você precisa enviar e-mail mediante plataformas como AWS e OCI.

O teste consiste em criar uma aplicação REST com endpoint que recebe dados para envio de email, **com apenas uma requisição, sem alterar o objeto de entrada**, dependendo da configuração setada em `application.properties` o objeto deve ser adaptado para novas classes, também deve ser serializado e impresso no console.

Requisitos

1 - Crie uma aplicação REST com um endpoint, que deverá receber um objeto e processar as informações.

A requisição deve ser feita utilizando o Postman ou uma ferramenta similar.

2 - Crie uma classe DTO e Modele os atributos contendo as seguintes informações:

- E-mail do destinatário
- Nome do destinatário
- Email do remetente
- Assunto
- Conteúdo

2.1 - O objeto recebido no Controller será uma instância da classe criada acima.

3 - Crie em `application.properties` um atributo chamado `mail.integracao`, que o valor pode ser configurado com **OCI** ou **AWS**.

4 - Crie a classe chamada **EmailAwsDTO**, com os seguintes atributos:

recipient	E-mail destinatário. Max: 45 caracteres
recipientName	Nome destinatário. Max: 60 caracteres
sender	E-mail remetente. Max: 45 caracteres
subject	Assunto do e-mail. Max: 120 caracteres
content	Conteúdo do e-mail. Max: 256 caracteres

5 - Crie a classe chamada **EmailOciDTO**, com os seguintes atributos:

recipientEmail	E-mail destinatário. Max: 40 caracteres
recipientName	Nome destinatário. Max: 50 caracteres
senderEmail	E-mail remetente. Max: 40 caracteres
subject	Assunto do e-mail. Max: 100 caracteres
body	Conteúdo do e-mail. Max: 250 caracteres

6 - Ao receber a requisição, adaptar a informação do objeto para a classe `EmailAwsDTO` ou `EmailOciDTO`, dependendo da configuração em `application.properties`.

7 - Serializar o objeto em JSON e imprimir no console.

8 - Se a requisição ocorrer com sucesso retornar status 204.

9 - Se a requisição falhar, tratar os erros com status 400 ou 500.

Avaliação

Habilidades de criação de projeto e arquitetura back-end:

- Capacidade analítica
- Implementação de arquitetura em camadas
- Desacoplamento dos componentes
- Conhecimento sobre REST
- Apresentação de código limpo e organizado
- Aplicação e conhecimento de SOLID
- Identificação e aplicação de Design Patterns
- Validação dos dados
- Tratamento de erros

O que será diferencial

- Documentação
- Teste unitário