

Teste para vaga de desenvolvedor BackEnd (DTS-Factory)

Equipe:

Equipe de back-end será composta de desenvolvedores de todos os níveis para atender as demandas dos diversos produtos da empresa.

Especificações do teste:

Este teste engloba o fluxo de trabalho básico da equipe atual, não foge das atividades cotidianas de um desenvolvedor back-end: criação da estrutura do projeto, controles, serviços, repositórios e modelos de dados.

O teste é composto por diversas atividades que definem a habilidade, nível profissional e experiência do desenvolvedor com relação às linguagens e ferramentas utilizadas pela equipe atual no desenvolvimento das aplicações.

O teste abaixo terá como objetivo avaliar o conhecimento técnico e pontos fortes do desenvolvedor nos seguintes critérios:

- Qualidade e legibilidade do código.
- Domínio sobre Java(1.8+) ,
- Conhecimentos sobre SpringBoot2, JPA/Hibernate, H2 e API Rest

Teste:

1. Para criar a API (micro serviço) Aluno em Spring Boot2, utilize **start.spring.io**, que fornecerá algum código de bootstrap (classe principal e arquivos pom.xml). Precisamos dos pacotes Web, JPA, Lombok e H2.
2. Monte a estrutura do projeto em pacotes model, repository, service, controller e handler.
3. A API Aluno deverá ter a seguinte configuração:
 - a. Rodar na porta: 8000
 - b. Contexto: /aluno
 - c. Utilizar jpa/hibernate
 - d. Utilizar H2 em memória, habilitando o console
4. Crie um handler para capturar os erros gerados (ex: 409 – Conflict para dados obrigatórios e não encontrados)
5. Crie o modelo (entidade JPA): Aluno com os seguintes atributos: id,nome,idade.
6. Crie um repositório JPA para controlar a entidade Aluno.
7. Crie um serviço de Aluno com os seguintes itens:
 - a. Recuperar todos os alunos
 - b. Recuperar um aluno por id
 - c. Criar um novo aluno validando os campos nome e idade com o BeanValidation.
 - d. Atualizar um aluno (verificando se o mesmo existe)
 - e. Remover um aluno
8. Crie um controlador para o serviço acima utilizando os conceitos rest.
9. Teste a API no SOAPUI.
Opcional: Swagger, Teste Unitário(JUnit).