

Desafio Backend - Node.js

Proposta

O objetivo deste desafio é a criação de uma API REST básica para um aplicativo de tarefas (TO-DO).

Requisitos

1. Cadastro de um novo usuário

- A API deve receber o nome, e-mail e senha do usuário para realizar a criação do usuário.
- A senha deve ser armazenada na forma de hash.
- Deve ser gerado automaticamente um ID para o usuário.
- Não deve permitir criar dois usuários com o mesmo e-mail.

2. Inclusão de uma nova tarefa

- A API deve receber a descrição da tarefa e a prioridade (Alta, Média, Baixa).
- O usuário não deve ser passado no JSON da requisição, mas obtido através do token de acesso do usuário.
- Deve ser gerado automaticamente um ID para a tarefa criada.

3. Listar as tarefas pendentes

- A API deve retornar a lista de tarefas pendentes de um usuário.
- O usuário deve ser identificado pelo token de autenticação.

4. Autenticação do usuário por meio de e-mail e senha

- A API deve receber o e-mail do usuário e a senha. O sistema deve procurar o usuário pelo e-mail e validar a senha.
- Após a verificação das credenciais, deve ser gerado um token de acesso (preferencialmente JWT).

Segurança

- A API não deve permitir acesso a nenhuma funcionalidade sem autenticação, exceto login e cadastro.

Requisitos técnicos

- Plataforma: a API deve ser desenvolvida em Node.js.
- Banco de dados: pode ser utilizado um banco de dados em memória (ex: SQLite).
- Build e Execução: Adicionar informações no “readme” do código fonte para rodar a aplicação em ambiente local, build e deploy da aplicação.

- Código fonte: o código fonte deve ser disponibilizado no GitHub.
- Testes unitários: Deverá existir testes unitários cobrindo pelo menos 50% do código fonte disponibilizado.

Ferramentas e Tecnologias Sugeridas

- Framework: Express.js
 - Autenticação: JWT
 - Documentação: Swagger
 - Testes: Jest
-

Desafio Frontend - Angular

Proposta

O objetivo deste desafio é a criação de uma aplicação Angular para consumir a API de tarefas (TO-DO) desenvolvida no backend.

Requisitos

1. Tela de Cadastro de Usuário

- Um formulário para o usuário inserir nome, e-mail e senha.
- Validação dos campos do formulário.

2. Tela de Login

- Um formulário para o usuário inserir e-mail e senha.
- Validação dos campos do formulário.

3. Tela de Lista de Tarefas

- Exibir a lista de tarefas pendentes do usuário.
- Exibir um botão para marcar a tarefa como concluída.

Requisitos técnicos

- Plataforma: a aplicação deve ser desenvolvida em Angular.
- Estilização: utilizar CSS básico.
- Comunicação com a API: utilizar HttpClientModule para realizar as requisições HTTP.

Ferramentas e Tecnologias Sugeridas

- Framework: Angular 12+
- Estilização: CSS

Desafio Publicação:

E se você precisar disponibilizar essa aplicação na AWS? Descreva brevemente como o faria.