

Instruções

Passo 1: Certifique-se de que há uma conexão sql com username = 'root' e password = 'password' operando em 'localhost'. Para este projeto, usei o MySQL Workbench e verifiquei essa configuração em Server -> Users and Privileges -> User Accounts.

Passo 2: É recomendado iniciar primeiro a api-rest porque se não houver a tabela do banco de dados criada, então nada vai funcionar corretamente. Para isso, em um terminal, entre na pasta api-rest-funcionarios e execute o comando "nodemon index.js". A api estará funcionando com os métodos POST, PUT, DELETE e GET.

Algumas aplicações como "Postman" permitem enviar requisições http sem necessidade de uma interface para tal. Neste caso, é importante salientar que é necessário que a primeira ação na api seja um GET em "localhost:4000/" para garantir a criação da tabela do banco de dados.

Após isso, procurei salientar por meio de respostas a requisições POST, PUT e DELETE em "localhost:4000/funcionario" quais os parâmetros corretos para conseguir êxito ao realizar essas operações.

Também é possível executar requisições GET em:

- "localhost:4000/funcionarios" para exibir os funcionários inseridos no banco de dados;
- "localhost:4000/funcionario/Id?Id={valor numérico, se existente no banco de dados}" para exibir um funcionário a partir de sua Id;
- "localhost:4000/funcionario/search"

Lembrando que tomei o cuidado de enviar respostas às requisições dando detalhes sobre as chaves importantes para preencher com valores na query.

Passo 3: Por fim, entre na pasta interface-funcionarios e, em um terminal, execute o comando "serve -s build". No endereço indicado pelo npm, geralmente localhost:5000, estará operando a interface com botões para busca (Search) e inclusão (Insert).

Sobre as funcionalidades Delete e Update, disponíveis na interface, foram idealizadas utilizando o conceito de Id, embora esse valor em nenhum momento fique disponível na interface.

A ideia é que a Id é um elemento precioso e que não seria exibido na interface, embora exibi-la na interface não seja um problema. Aqui ocorreu apenas uma questão conceitual em torno do elemento Id. Neste caso, seria necessário ter acesso ao banco de dados diretamente pela api para conseguir deletar ou atualizar um funcionário.