GPT 4o

Aqui está um guia completo para implementar a funcionalidade de listar os documentos no banco de dados em uma página HTML, utilizando **Node.js** com **Express**, **Mongoose**, e **Handlebars** (hbs). Também incluí os botões de "Alterar" e "Deletar".

**Passos**

**1. Crie a Rota no Node.js**

No arquivo do seu servidor Node.js (por exemplo, app.js ou index.js), você precisará de duas rotas principais:

* Uma rota para exibir a página com a lista de dados.
* Rota para **alterar** e **deletar** itens.

Aqui está o código:

javascriptCopiar

const express = require('express');

const mongoose = require('mongoose');

const router = express.Router();

// Modelo Mongoose

const SampleMconstrucao = mongoose.model('sample\_mconstrucao');

// Rota para listar os dados na página

router.get('/lojas', async (req, res) => {

try {

const lojas = await SampleMconstrucao.find().lean(); // .lean() torna o objeto legível pelo Handlebars

res.render('lojas', { lojas }); // Renderiza a lista

} catch (err) {

console.error(err);

res.status(500).send('Erro ao recuperar as lojas');

}

});

// Rota para deletar um item

router.post('/lojas/delete/:id', async (req, res) => {

try {

await SampleMconstrucao.findByIdAndDelete(req.params.id); // Deleta pelo ID

res.redirect('/lojas'); // Redireciona de volta para a lista

} catch (err) {

console.error(err);

res.status(500).send('Erro ao deletar o registro');

}

});

// Rota para exibir o formulário de alteração

router.get('/lojas/alterar/:id', async (req, res) => {

try {

const loja = await SampleMconstrucao.findById(req.params.id).lean();

res.render('alterarLoja', { loja });

} catch (err) {

console.error(err);

res.status(500).send('Erro ao recuperar o registro para edição');

}

});

// Rota para realizar a alteração

router.post('/lojas/alterar/:id', async (req, res) => {

try {

const { marcaloja, cidade, bairro, descricao } = req.body;

await SampleMconstrucao.findByIdAndUpdate(req.params.id, {

marcaloja,

cidade,

bairro,

descricao

});

res.redirect('/lojas'); // Redireciona para a página com a lista

} catch (err) {

console.error(err);

res.status(500).send('Erro ao alterar o registro');

}

});

module.exports = router;

**2. Crie o Template Handlebars para a Lista**

No diretório onde você armazena suas views (normalmente o diretório views), crie um arquivo chamado lojas.hbs para listar as lojas e adicionar os botões para alterar e deletar.

handlebarsCopiar

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Lista de Lojas</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body>

<div class="container mt-4">

<h1>Lista de Lojas</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Marca da Loja</th>

<th>Cidade</th>

<th>Bairro</th>

<th>Descrição</th>

<th>Ações</th>

</tr>

</thead>

<tbody>

{{#each lojas}}

<tr>

<td>{{marcaloja}}</td>

<td>{{cidade}}</td>

<td>{{bairro}}</td>

<td>{{descricao}}</td>

<td>

<!-- Botão Alterar -->

<a href="/lojas/alterar/{{\_id}}" class="btn btn-warning btn-sm">Alterar</a>

<!-- Botão Deletar -->

<form action="/lojas/delete/{{\_id}}" method="POST" style="display: inline;">

<button class="btn btn-danger btn-sm" onclick="return confirm('Tem certeza que deseja deletar?')">Deletar</button>

</form>

</td>

</tr>

{{/each}}

</tbody>

</table>

<a href="/" class="btn btn-primary">Voltar ao Início</a>

</div>

</body>

</html>

**3. Crie o Formulário para Alteração**

Crie um arquivo chamado alterarLoja.hbs no diretório de views. Esse template será usado para exibir o formulário de alteração dos dados.

handlebarsCopiar

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Alterar Loja</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body>

<div class="container mt-4">

<h1>Alterar Loja</h1>

<form action="/lojas/alterar/{{loja.\_id}}" method="POST">

<div class="mb-3">

<label for="marcaloja" class="form-label">Marca da Loja</label>

<input type="text" class="form-control" id="marcaloja" name="marcaloja" value="{{loja.marcaloja}}" required>

</div>

<div class="mb-3">

<label for="cidade" class="form-label">Cidade</label>

<input type="text" class="form-control" id="cidade" name="cidade" value="{{loja.cidade}}" required>

</div>

<div class="mb-3">

<label for="bairro" class="form-label">Bairro</label>

<input type="text" class="form-control" id="bairro" name="bairro" value="{{loja.bairro}}" required>

</div>

<div class="mb-3">

<label for="descricao" class="form-label">Descrição</label>

<textarea class="form-control" id="descricao" name="descricao" rows="3" required>{{loja.descricao}}</textarea>

</div>

<button type="submit" class="btn btn-success">Salvar Alterações</button>

<a href="/lojas" class="btn btn-secondary">Cancelar</a>

</form>

</div>

</body>

</html>

**4. Configure o Handlebars e os Middlewares**

Certifique-se de configurar corretamente o Handlebars no seu aplicativo Node.js. Por exemplo, no arquivo principal do servidor (app.js):

javascriptCopiar

const express = require('express');

const mongoose = require('mongoose');

const exphbs = require('express-handlebars');

const path = require('path');

const lojaRoutes = require('./routes/lojas'); // Arquivo com as rotas acima

const app = express();

// Conectar ao banco de dados MongoDB

mongoose.connect('mongodb://localhost/suabase', {

useNewUrlParser: true,

useUnifiedTopology: true

});

// Configurar o Handlebars como motor de visualização

app.engine('hbs', exphbs.engine({ extname: 'hbs' }));

app.set('view engine', 'hbs');

app.set('views', path.join(\_\_dirname, 'views'));

// Middlewares

app.use(express.urlencoded({ extended: true }));

app.use(express.json());

// Rotas

app.use('/', lojaRoutes);

// Servidor

const PORT = 3000;

app.listen(PORT, () => console.log(`Servidor rodando na porta ${PORT}`));

**5. Testar o Aplicativo**

* Acesse http://localhost:3000/lojas para visualizar a lista de lojas.
* Use os botões **Alterar** e **Deletar** para testar as funcionalidades.

**Resumo**

1. Criamos as rotas no Node.js para listar, alterar e deletar registros.
2. Configuramos o Handlebars como o motor de template.
3. Geramos páginas HTML para exibir a lista e o formulário de edição.
4. Adicionamos ações com botões para "Alterar" e "Deletar".

Se precisar de mais ajuda com estilos ou configurações adicionais, é só perguntar! 🚀

* Como estilizo a tabela da lista de lojas