//carregando o módulo express

const express = require('express');

// criar uma cópia inteira do framework dentro da variável

// criamos constante para não ter sobrescrição

const app = express();

// criando um servidor http com express

// colocamos no listen a porta que usaremos, a função deve ser a última do código

//função de callback é uma função que é executada sempre que algum evento acontece, sendo assim, na função listen devemos colocar uma função de callback, para quando ela estiver sendo executada, execute a função de callback

app.listen(8081, function(){

    console.log('Servidor rodando na url http://localhost:8081');

});

//localhost:8081



ROTAS

//carregando o módulo express

const express = require('express');

// criar uma cópia inteira do framework dentro da variável

// criamos constante para não ter sobrescrição

const app = express();

//devemos criar uma rota principal da aplicação

// Rota são os caminhos para a minha apicação

//req é a requisição que recebemos

//res o objeto que mandamos ao cliente

app.get("/", function(*req*, *res*){

*res*.send("Seja bem vindo ao meu app!")

})

//podemos criar um caminho na aplicação criando outra rota

app.get("/sobre", function(*req*, *res*){

*res*.send("Minha pagina sobre");

})

//podemos criar um caminho na aplicação criando outra rota

app.get("/blog", function(*req*, *res*){

*res*.send("Bem vindo ao meu blog")

})

// criando um servidor http com express

// colocamos no listen a porta que usaremos, a função deve ser a última do código

//função de callback é uma função que é executada sempre que algum evento acontece, sendo assim, na função listen devemos colocar uma função de callback, para quando ela estiver sendo executada, execute a função de callback

app.listen(8081, function(){

    console.log('Servidor rodando na url http://localhost:8081');

});

//localhost:8081

**Intalar o nodemon**

Nodemon é uma biblioteca que permite você não desligar e ligar o servidor, ele reinicia automaticamente toda vez que salvamos no Visual Code.

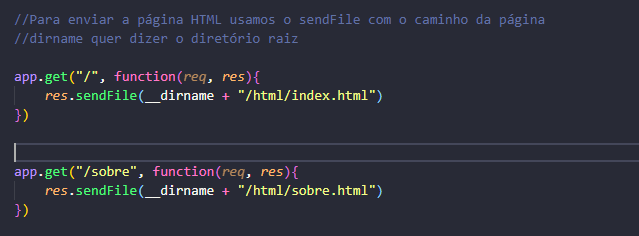
Comando:

Npm install nodemon -g (o g é para instalar globalmente no sistema, não só na pasta)



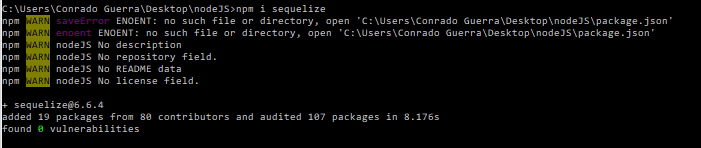
**Carregando página em html**

Usamos a função:



Dentro dela utilizamos o **\_\_dirname** (O que quer dizer o nosso diretório raiz), concatenando com a página que queremos.

Para utilizarmos o banco de dados no node, devemos instalar a biblioteca **SEQUELIZE:**



Devemos instalar o handlebars aula06

Aula 06 Montando Formulário HTML

Como montar o formulário através de handlebars e como chama-lo à página.

Como enviar formulários, via get/post

Instalamos o bodyParser para receber dados de qualquer formulário, com o body parser podemos escolher quais dados recebemos.

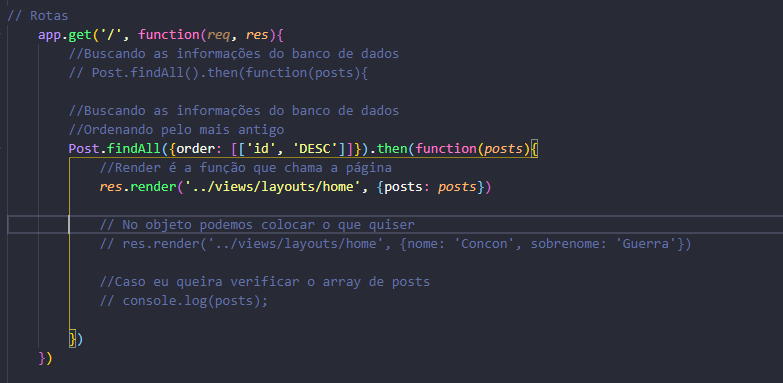
Aula 07:

Criamos uma pasta chama “models” na raiz do aplicativo, dentro dessa pasta criamos um JS chamado db, que será a conexão do banco de dados.

Criamos os models para cara utilização que tiver no banco de dados, boas práticas: Primeira letra maiúscula, Nome no singular.

Função de criar posts e redirecionar à pagina principal.

Buscando informações do banco de dados:

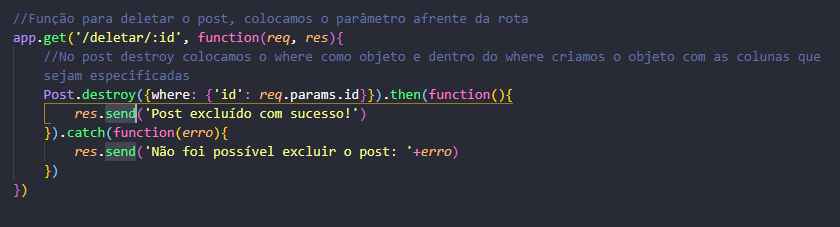


Para buscar todas as informações na tabela, usamos a função findall, com a resposta dessa função, utilizamos o then, podendo acessar os arrays do select

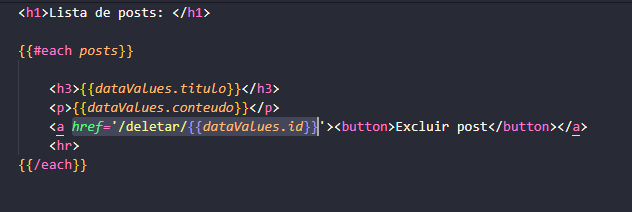
Na página home chamamos os dados obtidos pelo findAll, utilizando o each, separamos o array:



Excluindo itens do banco de dados:



No front:

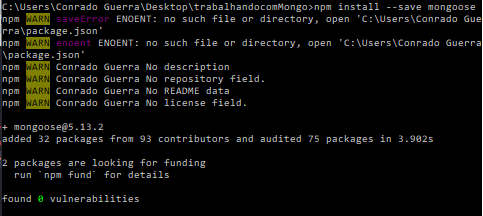


Mongo DB:

Temos que criar uma pasta c:/data/db

Depois no terminal damos o comando “mongod”.

Temos que instalar o mongoose para trabalhar com mongo no Node:



Para conectar chamamos o mongoose:

