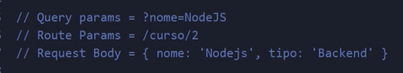
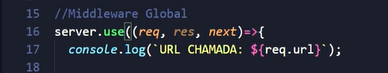


Os parametros



Outra forma de manipular um middleware global



Nesse caso essa msg vai aparecer no console dessa forma, pq foi editado no index 1



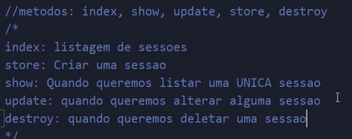
Modelo MVC, é a criação das pastas Model, View e Controller.

**Model –** Representa uma tabela no banco de dados, ou seja, o schema da aplicação (criação do file sempre o nome com letra maiúscula).

**View –** representa o front-end

**Controller –** Tem como objetivo tratar a requisição e devolver um resposta para a rota

Metodos controllers.



Na foto abaixo o que está sendo mandado e recebido é um email então pode colocar com vai estar no repositório desse projeto (olhar o repositório) caso for algo diferente tem que colocar 2 nome

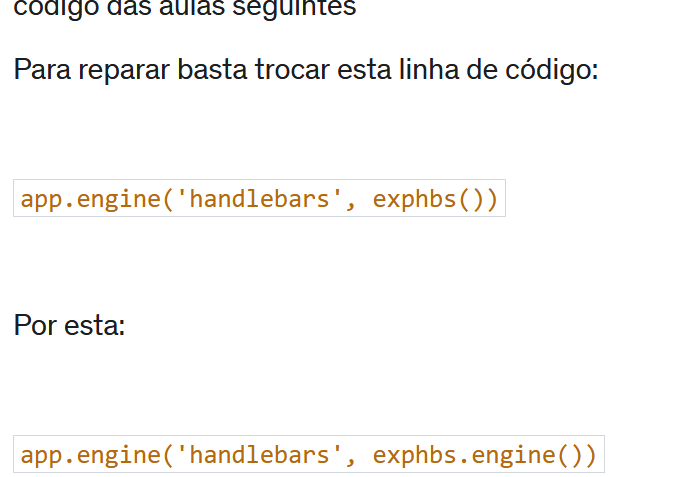


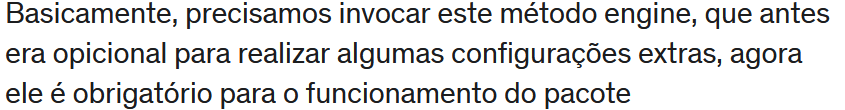
Para conseguir usar o chalk com o require é neessario instalar a versão 4.1.2 da seguinte maneira.

1. Desinstale localmente a versão atual do Chalk usando npm uninstall chalk ;

2. Instale localmente a versão anterior à 5 usando npm install chalk@4.1.2 .

Nem sempre é necessário desistalar para instalar





**CONFIGURAÇÃO BACKEND PARA PROJETO**



**Bcrypt –** Fazer algoritmo de senha, criptografar a senha

**Cookie parser –**

**Cors –** Mandar requisição para o nosso mesmo domínio. Isso é para quando com a API e frot-end no memso servidor

**Express –**

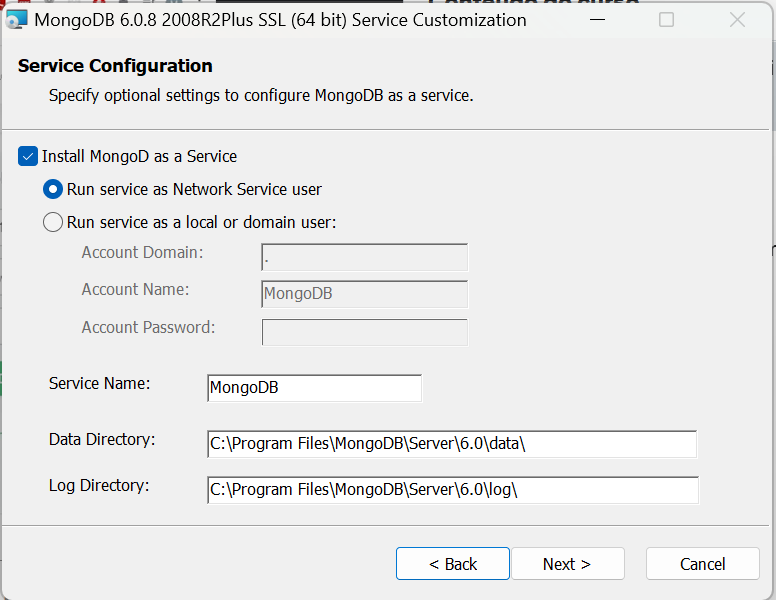
**Jsonqebtoken –** Método de autenticação

**Mongoose –** Banco de dados

**Multer –** Upload de fotos

**Nodemon –**

**PATH DE INSTALAÇÃO DO MONGODB**

****

mwkKyOE1sH8zxDrY

const supabase = require('../db/conn')

// Função para inserir um novo Usuário

async function createUser(*userData*) {

    const { data, error } = await supabase.from('users').insert([*userData*]);

    if (error) {

      console.error('Erro ao inserir usuário:', error);

      return null;

    }

    console.log('Usuário inserido com sucesso!');

    return data[0];

  }

  // Função para atualizar um Usuário existente

  async function updateUser(*userId*, *userData*) {

    const { data, error } = await supabase

      .from('users')

      .update(*userData*)

      .match({ id: *userId* });

    if (error) {

      console.error('Erro ao atualizar usuário:', error);

      return null;

    }

    console.log('Usuário atualizado com sucesso!');

    return data[0];

  }

  // Função para obter um Usuário pelo ID

  async function getUserById(*userId*) {

    const { data, error } = await supabase

      .from('users')

      .select()

      .match({ id: *userId* });

    if (error) {

      console.error('Erro ao obter usuário:', error);

      return null;

    }

    return data[0];

  }

*module*.*exports* = {

    createUser,

    updateUser,

    getUserById,

  };

Vagas 1 - <https://lnkd.in/dJCBNQQ>  
Vagas 2 - <https://lnkd.in/dqxNpdv>  
Vagas 3 - <https://lnkd.in/eehksxE>  
Vagas 4 - <https://lnkd.in/dPVfqMaZ>  
Vagas 5 - <https://lnkd.in/dX9saDWT>  
Vagas 6 - <https://lnkd.in/dyS43Vuq>  
Vagas 7 - <https://lnkd.in/dnRAqDyt>  
Vagas 8 - <https://lnkd.in/d9TM8A23>  
Vagas 9 - <https://lnkd.in/d3bECiCC>  
Vagas 10 - <https://lnkd.in/d5yDSJGb>  
Vagas 11 - <https://lnkd.in/drvJmae>  
Vagas 12 - <https://lnkd.in/dxa9hdbH>  
Vagas 13 - <https://lnkd.in/dRHHmSm>  
Vagas 14 - <https://lnkd.in/d-ErsDm>  
Vagas 15 - <https://lnkd.in/drD9a6s>  
Vagas 16 - <https://lnkd.in/dQa3AKW>  
Vagas 17 - <https://lnkd.in/dBajhqJ>  
Vagas 18 - <https://lnkd.in/et44F3WG>  
Vagas 19 - <https://lnkd.in/e2vCmMsn>  
Vagas 20 - <https://lnkd.in/eSQ3s2ad>  
Vagas 21 - <https://lnkd.in/eGxdVxAJ>  
Vagas 22 - <https://lnkd.in/ekCXhq6B>  
Vagas 23 - <https://lnkd.in/e7j7aaaG>  
Vagas 24 - <https://lnkd.in/demu6zH>  
Vagas 25 - <https://lnkd.in/dhfW7KEn>

function addAmount(*accountName*, *amount*) {

    const accountData = getAccount(*accountName*)

    if (!*amount*) {

        console.log(chalk.bgRed.black('Ocorreu um erro, tente novamente mais tarde!'))

        return deposit()

    }

    accountData.balance = parseFloat(*amount*) + parseFloat(accountData.balance)

*fs*.writeFileSync(`accounts/${*accountName*}.json`, JSON.stringify(accountData),

        function (*err*) {

            console.log(*err*)

        }

    )

    console.log(chalk.green(`Foi depositado o valor de R$${*amount*} na sua conta!`))

}