# Node + Express+jwt+postgres

Texto

Descripción generada automáticamente

Dar npm init -y esto agrega un package json

Texto

Descripción generada automáticamente

Agregar type module

Captura de pantalla de un celular

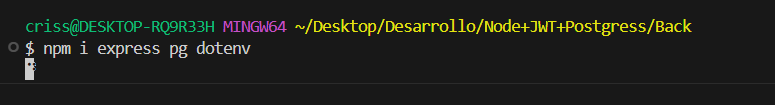
Descripción generada automáticamente

Agregar nodemon y start

Captura de pantalla de un celular

Descripción generada automáticamente

# Instalaciones express variables de entorno



Y nodemon

Texto

Descripción generada automáticamente

Agregar configuraciones en el index.js

// Importar los módulos necesarios

import 'dotenv/config'

import express from 'express';

const app = express();

// Definir una ruta básica

app.get('/', (req, res) => {

  res.send('Hola Mundo');

});

// Definir el puerto en el que el servidor va a escuchar

const port = process.env.PORT || 3000;

// Iniciar el servidor y escuchar en el puerto definido

app.listen(port, () => {

  console.log(`Servidor corriendo en el puerto ${port}`);

});

Npm run dev

Texto

Descripción generada automáticamente

# Crear carpetas para entorno de trabajo

Captura de pantalla de computadora

Descripción generada automáticamente

# Configurar la BD

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

Abrir el pgadmin

Crear db llamada jwt\_db

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ahora en query tool

Captura de pantalla de computadora

Descripción generada automáticamente

Agregar script

drop table if exists USERS;

-- Eliminar la tabla 'users' si existe

DROP TABLE IF EXISTS users;

-- Crear la tabla 'users'

CREATE TABLE users (

UID SERIAL PRIMARY KEY,

EMAIL VARCHAR(50) NOT NULL UNIQUE,

PASSWORD VARCHAR(60) NOT NULL,

USERNAME VARCHAR(50) NOT NULL

);

-- Insertar un registro en la tabla 'users'

INSERT INTO users (EMAIL, PASSWORD, USERNAME) VALUES

('test@test.com', '1234', 'test');

-- Seleccionar todos los registros de la tabla 'users'

SELECT \* FROM users;

# Conexión a postgress

En la pagina de postgres

Texto

Descripción generada automáticamente

Código connection.databse.js

import 'dotenv/config';

import pg from 'pg';

const { Pool } = pg;

const connectionString = process.env.DATABASE\_URL

export const db = new Pool({

    allowExitOnIdle: true,

    connectionString

})

    try {

        await db.query('SELECT NOW()');

        console.log('DATABASE CONNECTED');

    } catch (error) {

        console.error(error);

    }

Código env

DATABASE\_URL="postgresql://postgres:1234@localhost:5432/jwt\_db"

Prueba

node database/connection.database.js

Texto

Descripción generada automáticamente

# Creando el modelo

Se agrega user.models.js

Captura de pantalla de computadora

Descripción generada automáticamente

El codigo

import { db } from '../database/connection.database.js';

const create = async (email, password, username) => {

    const query = {

        text: `INSERT INTO users (email, password, username)

               VALUES ($1, $2, $3)

               RETURNING email, username, uid`,

        values: [email, password, username]

    };

    const { rows } = await db.query(query);

    return rows[0]; // Retorna la primera fila que contiene los datos insertados

};

const findOneByEmail = async (email) => {

    const query = {

        text: `SELECT \*

               FROM users

               WHERE email = $1`,

        values: [email]

    };

    const { rows } = await db.query(query);

    return rows[0]; // Retorna la primera fila que contiene el usuario encontrado

};

export const UserModel = {

    create,

    findOneByEmail

};

# Creando el contralor

Texto

Descripción generada automáticamente

import { UserModel } from "../models/user.model.js";

const register = async(req, res) => {

    try {

        console.log(req.body)

        const { username, email, password } = req.body

        return res.json({ ok: true, msg: "user ok" })

    } catch (error) {

        console.log(error)

        return res.status(500).json({

            ok: false,

            msg: 'Error Server'

        })

    }

}

const login = async(req, res) => {

    try {

    } catch (error) {

        console.log(error)

        return res.status(500).json({

            ok: false,

            msg: 'Error Server'

        })

    }

}

export const UserController =

 { register, login };

# Creando las RUTAS

Texto

Descripción generada automáticamente

import { Router} from "express";

import { UserController } from "../controllers/user.controller.js"

const router = Router()

router.post('/register', UserController.register);

router.post('/login', UserController.login);

export default router;

# Probando los cambios

Modificar el index.js

// Importar los módulos necesarios

import 'dotenv/config'

import express from 'express';

import userRouter from './routes/user.route.js'

const app = express();

app.use('/api/v1/users',userRouter)

// Definir una ruta básica

// app.get('/', (req, res) => {

//   res.send('Hola Mundo');

// });

// Definir el puerto en el que el servidor va a escuchar

const port = process.env.PORT || 3000;

// Iniciar el servidor y escuchar en el puerto definido

app.listen(port, () => {

  console.log(`Servidor corriendo en el puerto ${port}`);

});

Levantar

Imagen que contiene Logotipo

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Error

ndefined

TypeError: Cannot destructure property 'username' of 'req.body' as it is undefined.

at register (file:///C:/Users/criss/Desktop/Desarrollo/Node+JWT+Postgress/Back/controllers/user.controller.js:7:17)

at Layer.handle [as handle\_request] (C:\Users\criss\Desktop\Desarrollo\Node+JWT+Postgress\Back\node\_modules\express\lib\router\layer.js:95:5)

at next (C:\Users\criss\Desktop\Desarrollo\Node+JWT+Postgress\Back\node\_modules\express\lib\router\route.js:149:13)

at Route.dispatch (C:\Users\criss\Desktop\Desarrollo\Node+JWT+Postgress\Back\node\_modules\express\lib\router\route.js:119:3)

at Layer.handle [as handle\_request] (C:\Users\criss\Desktop\Desarrollo\Node+JWT+Postgress\Back\node\_modules\express\lib\router\layer.js:95:5)

agregar estas modificaciones

app.use(express.json());

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

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Instalar thunder client

Captura de pantalla de computadora

Descripción generada automáticamente

New request

Captura de pantalla de un celular

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Prueba agregando

{

"username": "test",

"email":"test@test.com",

"password":"1234"

}

Captura de pantalla de computadora

Descripción generada automáticamente

Ahora en el request