Links Uteis

O projeto passo a passo  
<https://github.com/paulosalvatore/ocean-jornada-fullstack-novembro-22>

Programa para compartilhar tela escrita (Tipo uma Lousa)  
<https://miro.com/app/board/uXjVPGdjF2Y=/>

Pacote Express, para rodar o projeto em HTTP  
<https://www.npmjs.com/package/express>

Para configurar o Prittier  
<https://www.youtube.com/watch?v=vFByu7-f858>

Outros

<https://www.youtube.com/watch?list=PLzFYWfRVHp5Jx-iNLxhUydXwnYPqmV2MU&v=F81k9w8gDJs&feature=youtu.be&ab_channel=PauloSalvatore>

Nodemon

<https://www.npmjs.com/package/nodemon>

Banco de dados, Hanking

<https://db-engines.com/en/ranking>

Comando MongoDB

<https://www.npmjs.com/package/mongodb>

Para verificar algumas coisas da função await

<https://www.youtube.com/watch?v=QMjZyQZMOSY&feature=youtu.be&ab_channel=PauloSalvatore>

Bloco de Notas que envia texto para outras pessoas pela URL

<https://wordpad.cc/>

Banco de dados Mongo, para colocar na nuvem

<https://www.mongodb.com/atlas/database>

Colocar endereço IP fixo

<https://www.noip.com/pt-BR>

O PACKGE.JSON é o arquivo mais importante do projeto. É o coração. Para chamar ele, vai em Terminal > New Terminal e digita **npm init -y**

Criar o **index.js**, clicando em novo e editando o nome para **index.js**

Começa a escrever os códigos e para verificar o que foi escrito, usa o TERMINAL para ver, escrevendo o comando **node index.js**

Para fazer interface gráfica, criação de servidor, etc, é necessário instalar o EXPRESS, cada projeto deve ser feita a instalação

Depois de fazer o npm init -y é possível instalar o express usando o comando **npm install express**

\*Caso você tenha pego algum projeto sem a pasta **node\_modules** é só chamar a **npm install** que a pasta será criada no projeto já existem

Quando o node index.js estiver rodando no terminal, para parar é só clicar no terminal e apertar CRTL + C

**Instalar o Thunder Client**

Para testar a aplicação BackEnd usamos o Thunder Client, para tanto, ir em EXTENSOES e digitar Thunder Client e instalar. Depois clicar no ícone RAIO do lado esquerdo e gerar **New Request** , nesse caso mudamos o endereço para http://localhost:3000

**Usar os operadores**

**Conceito:** REST é uma convenção para melhorar a arrumação

Diagrama

Descrição gerada automaticamente

Fazer a aplicação usando NOME e IMAGEM

Lista ou Arrays é a mesma coisa.

**Const** – Criar informações que não mudam, não pode ser alterado

**Let** – A informação pode ser mudada

Var – Permite ser alterada

O CONST permite colocar item dentro do cochetes, exemplo fazer um PUSH no CONST lista [1,2,3], colchete é o símbolo que insere uma nova lista.

Iremos instalar o NODEMON que reinicializa automaticamente a aplicação quando salva.

Podemos instalar de forma completa, na maquina, ou somente no projeto.

Iremos usar a instalação somente no projeto: **npm install --save-dev nodemon**

Criaremos o SCRIPT, então modificamos

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

Por

  "scripts": {

    "start": "node index.js",

    "dev": "nodemon index.js"

  },

Para rodar o script devemos digitar no terminal: **npm run dev** , esse Script deve ser iniciado toda vez que abrir o projeto para trabalhar.

Para encerrar a aplicação **npm run dev** é necessário clicar no terminal e clicar Ctrl+C duas vezes

**Tip**: Atl + shift + seta para baixo duplica a linha

Para usar o MongoDB precisa instalar no terminal do VSCODE

npm i mongodb

E colocar outras constantes no VSCODE

const { MongoClient, ObjectId } = require("mongodb")

fica entre:

const express = require('express')

const { MongoClient, ObjectId } = require("mongodb")

const app = express()

Depois é necessário colocar o endereço do banco de dados para fazer a conexão, então precisa configurar a rota

const url ="mongodb://localhost:27017";

Depois temos que declarar o nome do banco de dados exatamente igual ao criado

const BancoDadosNome = "Ocean\_Jornada\_Fullstack\_Novembro\_2022";

Agora é hora de conectar, e é necessário usar o comando connect dentro de uma CONST com qualquer nome, nesse caso iremos usar o nome cliente

const client = MongoClient.connect(url);

como leva tempo uma conexão com o banco de dados, pois é uma PROMISE, é necessário fazer um Async e Await, necessita criar uma função, que daremos o nome de main, pois será o principal e faremos um main() para executar. Obs: Para usar o Await é necessário fazer uma função Async

async function main() {

    const client = await MongoClient.connect(url);

}

main();

Para que tudo funcione dentro do Async, é necessário colocar todo o código dentro do main(), trazendo o fechamento para o final

}

main();

Para colocar na NUVEM

Entrar no site <https://www.mongodb.com/atlas/database>

Criar usuário e senha

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Adm: Admin

Senha: GqB1qB34Drs9wrdt

Transfor de mongodb+srv://admin:<password>@cluster0.pdjfbuo.mongodb.net/?retryWrites=true&w=majority

para

URL do BD: mongodb+srv://admin:GqB1qB34Drs9wrdt@cluster0.pdjfbuo.mongodb.net/

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Conecta sua aplicação

