Seção 03 Fundamentos do Express.js e NPM

* Aula 01 O que é o Express.js e NPM

Vamos iniciar um projeto npm

Em sua pasta do projeto digite o código:

npm init -y

feito isso ele ira criar um arquivo package.json, onde ficará salvo as configurações do seu projeto

depois de iniciado o projeto npm vamos instalar o npm:

npm install express --save

será criado uma pasta node\_modules, onde ficam instaladas todas as suas bibliotecas

* Aula 02 Estrutura inicial

Vamos iniciar nossa aplicação express

Na pasta do nosso projeto, crie um arquivo index.js

index.js

const express = require('express')

// Importando o express

const app = express()

// Iniciando o express

app.listen(4000, erro => {

    if(erro) console.log('Ocorreu  um ERRO!')

    else console.log('Servidor iniciado com sucesso!!!')

})

Só isso já e o necessário para criar nosso servidor, a porta que escolhemos para executar será a 4000

Agora para rodar o servidor basta executar esse arquivo criado no terminal:

node index.js

agora podemos ir no navegador e colocar o endereço para verificar:

localhost:4000

porem no endereço sera apresentado apenas um erro de rotas, que ainda iremos ajustar, para criar as rotas o modelo mais simple e usando o app.get()

* Aula 03 Rotas

Index.js

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

    res.send('Sou uma página principal')

})

Mostra o que a rota irá apresentar ao usuário, aqui foi passado o /,

Logo quando o endereço principal for acessado será retornado essa informação

index.js

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

    res.send('Sou uma página principal')

})

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

    res.send('<h1>Bem vindo ao meu blog</h1>')

})

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

    res.send('<h1>Bem vindo ao meu canal no Youtube</h1>')

})

Caminhos de sua aplicação

* Aula 04 Carregamento automático

A cada vez que ateramos nosso arquivo precisamos parar o servidor e rodalo novamente para ver as alterações feita

Para não ter esse trabalho podemos utilizar um modulo no node chamado de nodemon

Esse nodemon para e inicia o servidor cada vez que ele verifica alguma alteração no seu arquivo

Para instalar o nodemon usamos o código:

npm install nodemon -g

o -g significa que será instalado de maneira global em todos os projetos em sua maquina

agora a cada alteração feita seu servidor sera fechado e aberto automaticamente, bastando apenas atualizar a pagina de seu projeto

importante que esse código seja feito no prompt de comando, no caminho do seu projeto

* Aula 05 Parametros

Parâmetros são forma de você passar valores dinâmicos direto na sua rota

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

    res.send('<h1>Oi</h1>')

})

Aqui foi criado um parâmetro nome, você pode criar quantos parâmetros quiser

Nesse tipo de rotas e obrigatório passar um paramentro para a rota,

Caso acesse /ola, ira apresentar um erro, sendo nessesario informar um parâmetro qualquer. ola/nomequaquer

Agora vamos pegar o valor que foi enviado como parâmetro

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

    // REQ => DADOS ENVIADOS PELO USUÁRIO

    // RES => RESPOSTA QUE VAI SER ENVIADA PARA O USUÁRIO

    let nome = req.params.nome

    res.send(`<h1>Oi ${nome}</h1>`)

})

O req.params.nome => recebe o valor passado no parametro nome

app.get('/ola/:nome/:empresa', (req, res) => {

    let nome = req.params.nome

    let empresa = req.params.empresa

    res.send(`<h1>Oi ${nome} do ${empresa}</h1>`)

})

Múltiplos parâmetros também e possível

Agora veremos parâmetros não obrigatórios

app.get('/blog/:artigo?', (req, res) => {

    res.send('<h1>Bem vindo ao meu blog</h1>')

})

Aqui o parâmetro artigo não e obrigatório para acessar a pagina blog

app.get('/blog/:artigo?', (req, res) => {

    let artigo = req.params.artigo

    if(artigo) {

        res.send(`Você está no blog do artigo de ${artigo}`)

    }

    else {

        res.send('Bem vindo ao meu blog')

    }

})

Aqui caso seja passado algum parâmetro ele irá para um caminho,caso não seja passado parâmetro ele irá para outro caminho

* Aula 06 Query params

Tipo de parâmetro que não e fixo na rota

localhost:4000/canal/youtube?canal=meuCanal

agora vamos pegar esse valor das query

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

    let canal = req.query['canal']

    if(canal) {

        res.send(`<h1>Você está no canal do ${canal}</h1>`)

    } else {

        res.send('<h1>Bem vindo ao Youtube</h1>')

    }

})

Esse tipo de abordagem está em dissuso