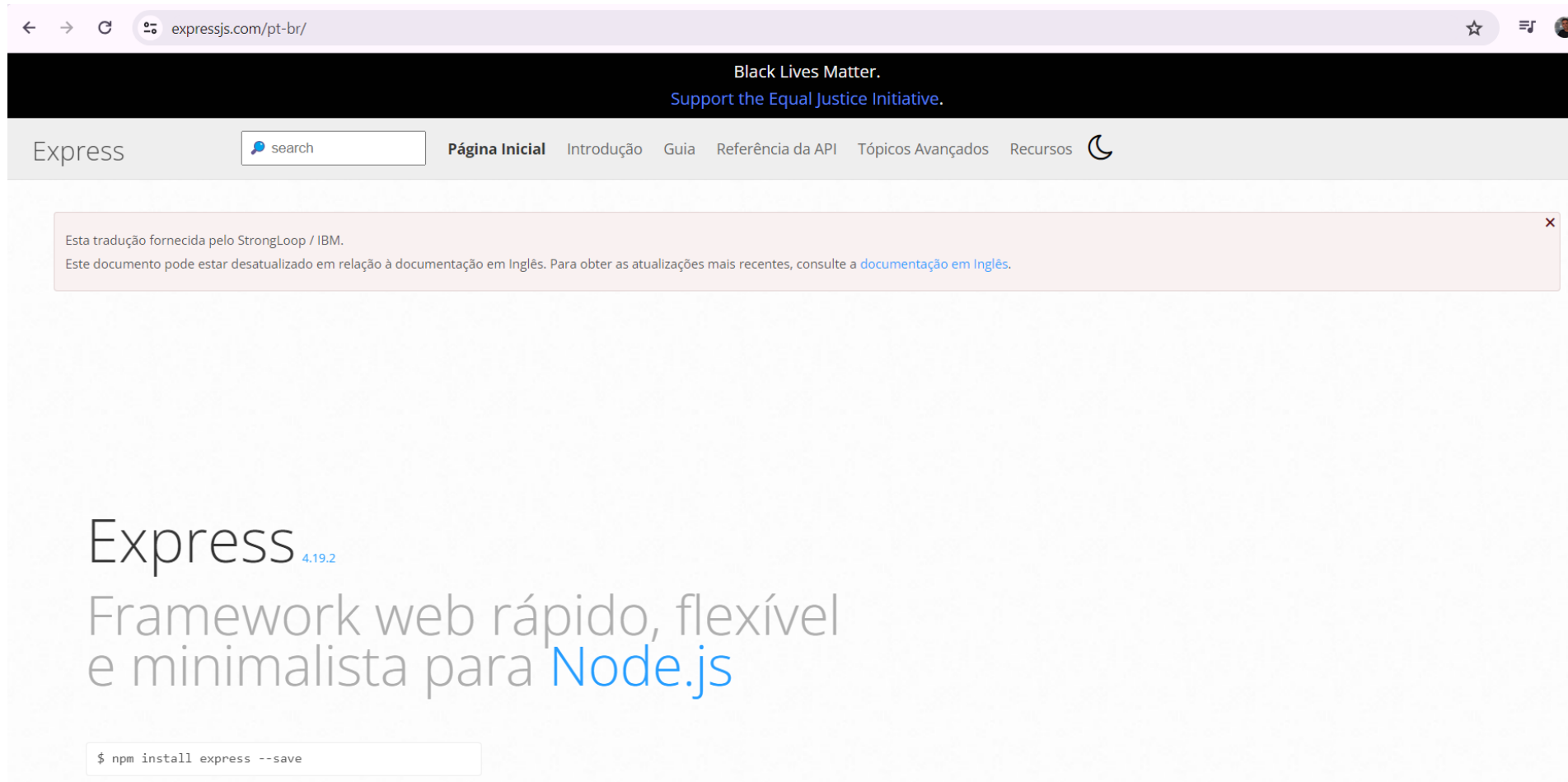




# NODE.JS

IW-I – Prof. Anderson Vanin

# INTRODUÇÃO AO EXPRESS

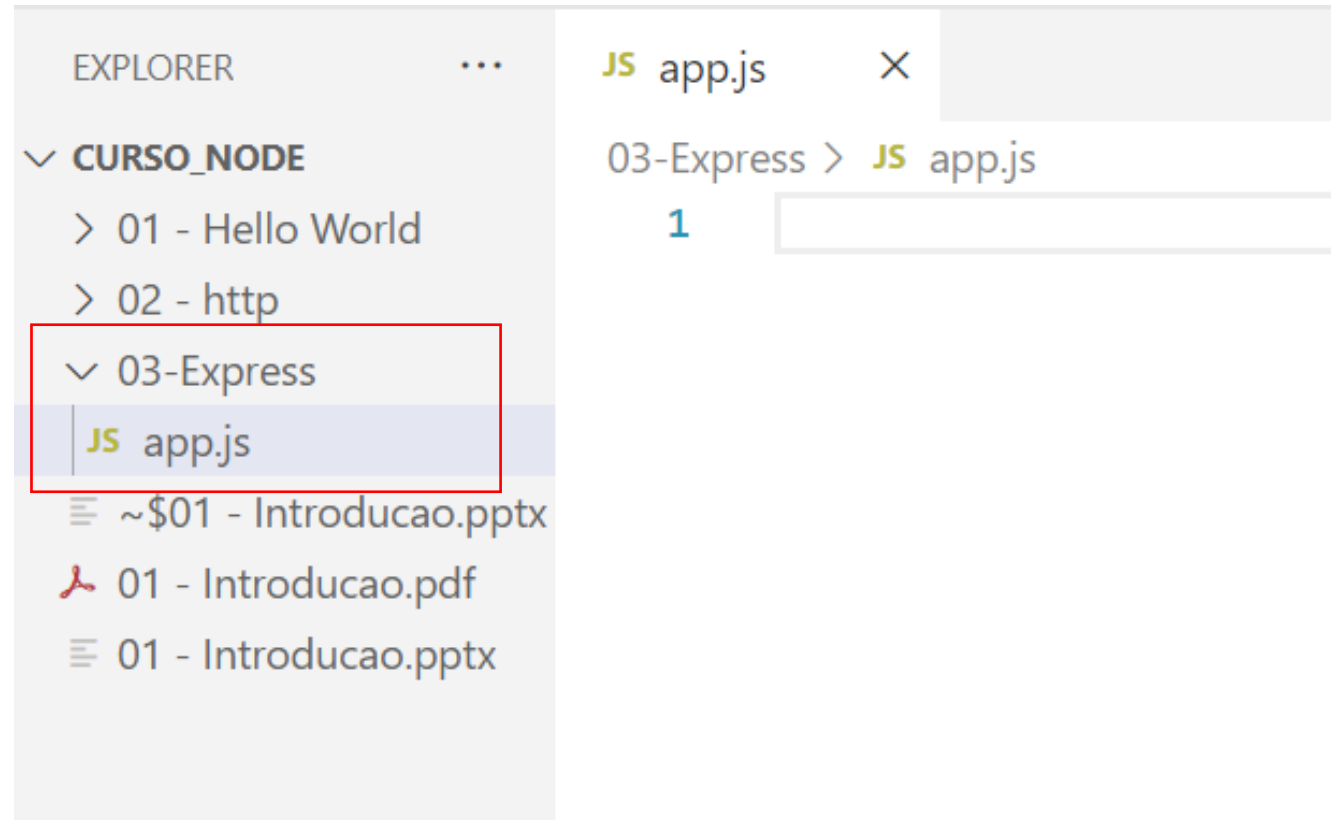
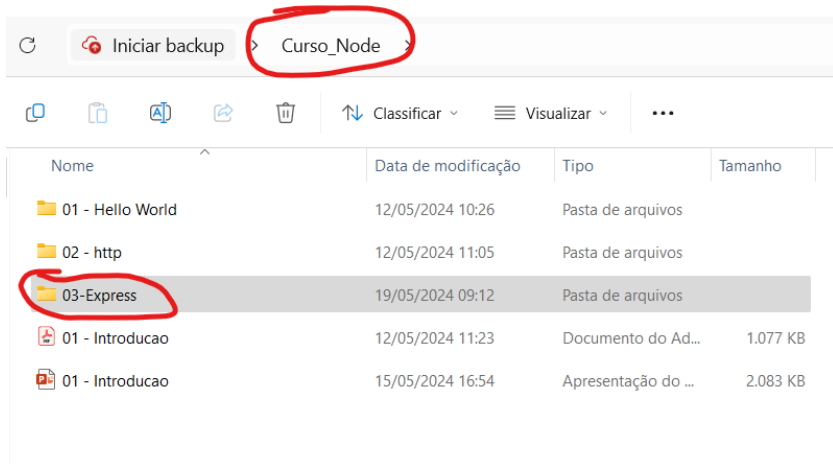


# INTRODUÇÃO AO EXPRESS

- O *Express* é um *framework* muito utilizado por diversas empresas para criar de forma rápida e eficiente várias aplicações web.
- É um *framework* minimalista.
- Para usar o *express* vamos precisar utilizar o **npm**.
- O **npm** é o maior repositório de pacotes que existe.

# INICIANDO

- Dentro da sua pasta Curso Node crie uma outra pasta chamada 03-express. Na pasta criada crie um arquivo chamado app.js e abra ela no vscode.





# INICIANDO

- Abra o terminal de comando na pasta criada e digite:

**npm install express --save**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:\Users\Anderson\Desktop\Curso\_Node\03-Express>npm install express --save

03-Express

> node\_modules

JS app.js

{ } package-lock.json

{ } package.json

Ao instalar o express, é criada uma pasta chamada node-modules onde são instalados todos os módulos e dependências da sua aplicação.

03-Express

node\_modules

> .bin

> accepts

> array-flatten

> body-parser

> bytes

> call-bind

> content-disposition

> content-type

> cookie

> cookie-signature

> debug

> define-data-property

> depd

> destroy

> ee-first

> encodeurl

> es-define-property

> es-errors

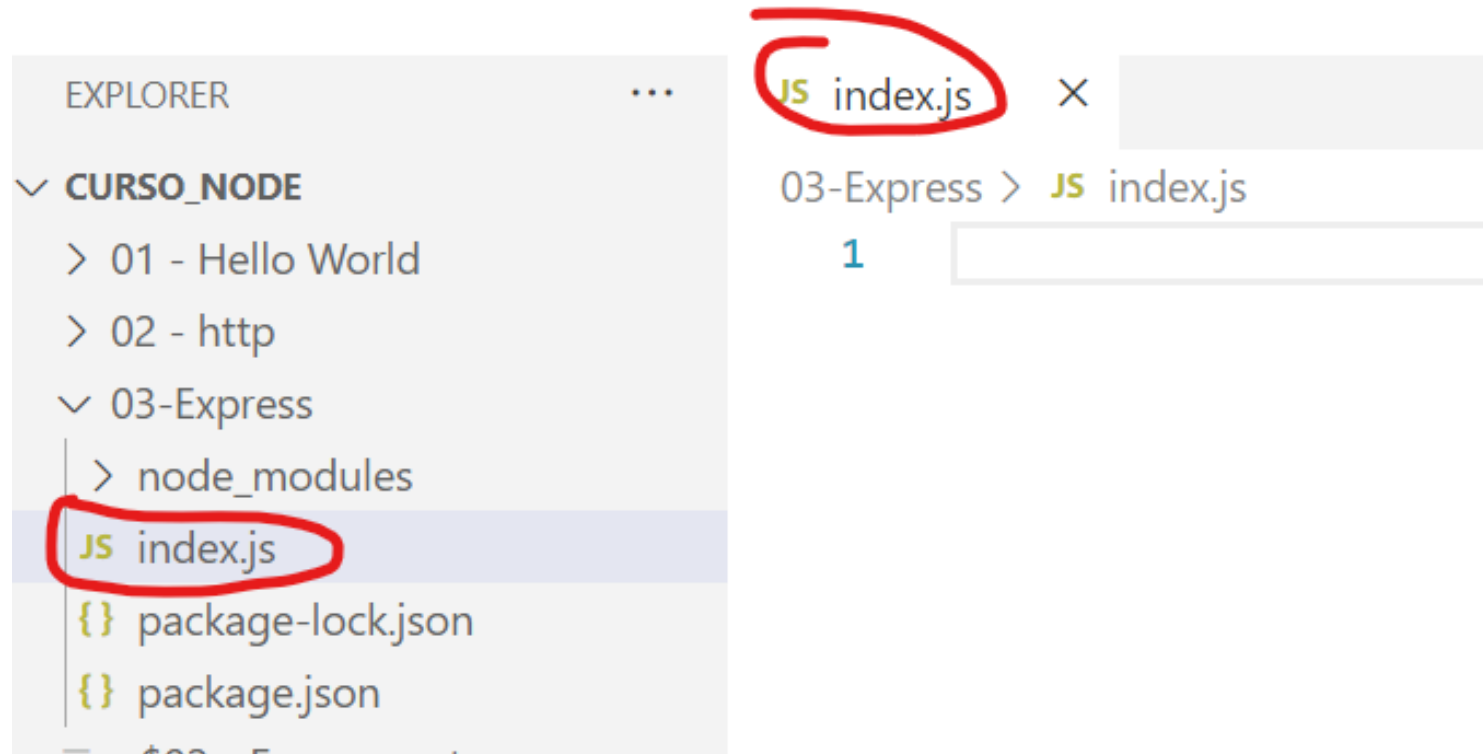
# PACKAGE-LOCK.JSON

- Este é um arquivo importante que lista todas as dependências de seu projeto:

```
194 },
195 "node_modules/express": ,
196   "version": "4.19.2",
197   "resolved": "https://registry.npmjs.org/express/-/express-4.19.2.tgz",
198   "integrity": "sha512-5T6nhjsT+EOMzuck8JjBHARTHfMht0POz1A60WV2pMD3gyXw2L
199   "dependencies": {
200     "accepts": "~1.3.8",
201     "array-flatten": "1.1.1",
202     "body-parser": "1.20.2",
203     "content-disposition": "0.5.4",
204     "content-type": "~1.0.4",
205     "cookie": "0.6.0",
206     "cookie-signature": "1.0.6",
207     "debug": "2.6.9",
208     "depd": "2.0.0",
209     "encodeurl": "~1.0.2",
210     "escape-html": "~1.0.3",
211     "etag": "~1.8.1",
212     "finalhandler": "1.2.0"
```

# ABRINDO UM SERVIDOR UTILIZANDO O EXPRESS

- Vamos renomear o arquivo *app.js* para *index.js* (na verdade esse deveria ter sido o nome no início desta aula ;-)).



# ABRINDO UM SERVIDOR UTILIZANDO O EXPRESS

- Vamos renomear o arquivo *app.js* para *index.js* (na verdade esse deveria ter sido o nome no início desta aula ;-)).

JS index.js X

03-Express > JS index.js > ...

```
1  const express = require('express')
2  const app = express()
3
4  // a variavel app será uma instância do módulo express
5
6  // transformando as variaveis que recebem os módulos em constantes
7  // você evita que eles sejam reescritas durante a execução do
8  // seu programa....
9
10
11
12
13
14
15  // Abrindo um servidor com express
16  // ESTA FUNÇÃO TEM QUE SER SEMPRE A ÚLTIMA LINHA DO SEU CÓDIGO
17  app.listen(8081)
```

Sua aplicação fica aqui

Coloque a chamada `app.listen(8081)` como sendo a última linha de seu código!

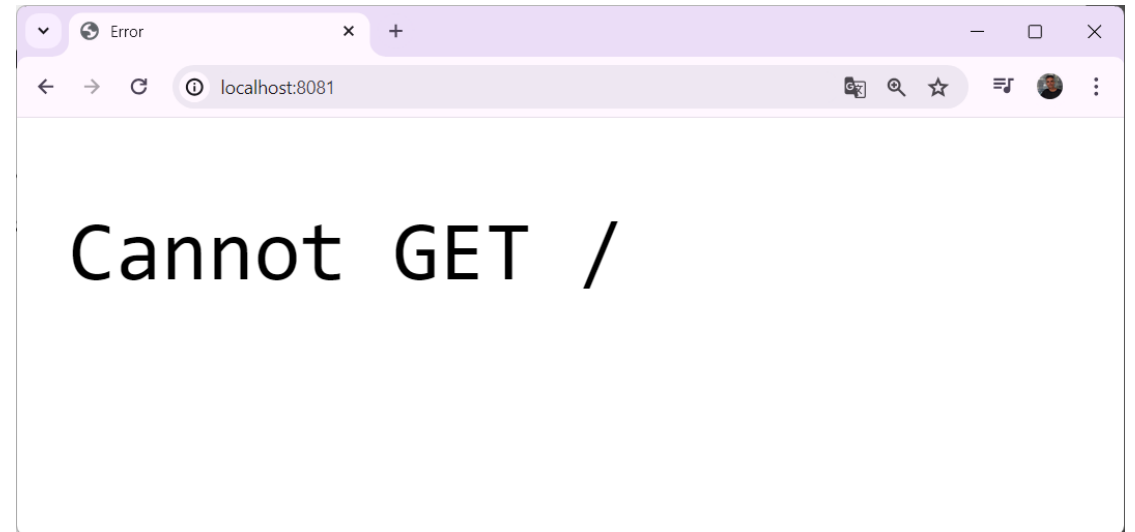


# ABRINDO UM SERVIDOR UTILIZANDO O EXPRESS

- Vamos rodar a aplicação *index.js* digite:

*node index.js* e depois abra o navegador em *localhost:8081*

```
C:\Users\Anderson\Desktop\Curso_Node\03-Express>node index.js  
█
```



# ABRINDO UM SERVIDOR UTILIZANDO O EXPRESS

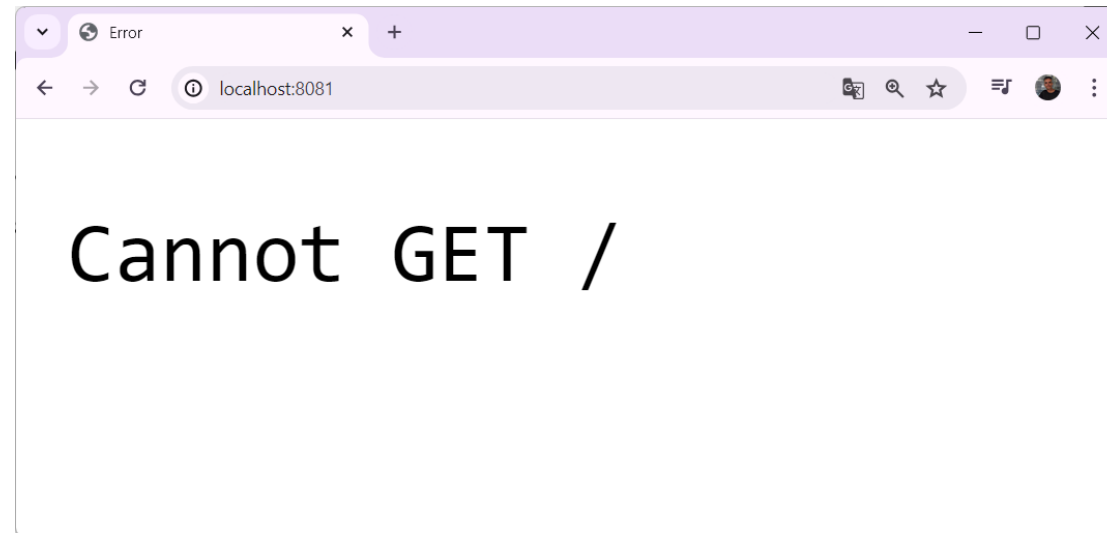
- Podemos chamar uma função de call-back para exibir mensagens:

```
// Abrindo um servidor com express
// ESTA FUNÇÃO TEM QUE SER SEMPRE A ÚLTIMA LINHA DO SEU CÓDIGO
app.listen(8081,function(){
  console.log("Servidor rodando na url http://localhost:8081")
})
```

```
C:\Users\Anderson\Desktop\Curso_Node\03-Express>node index.js
Servidor rodando na url http://localhost:8081
█
```

# ABRINDO UM SERVIDOR UTILIZANDO O EXPRESS

- Este erro acontece porque a nossa aplicação ainda não tem nenhuma rota definida!

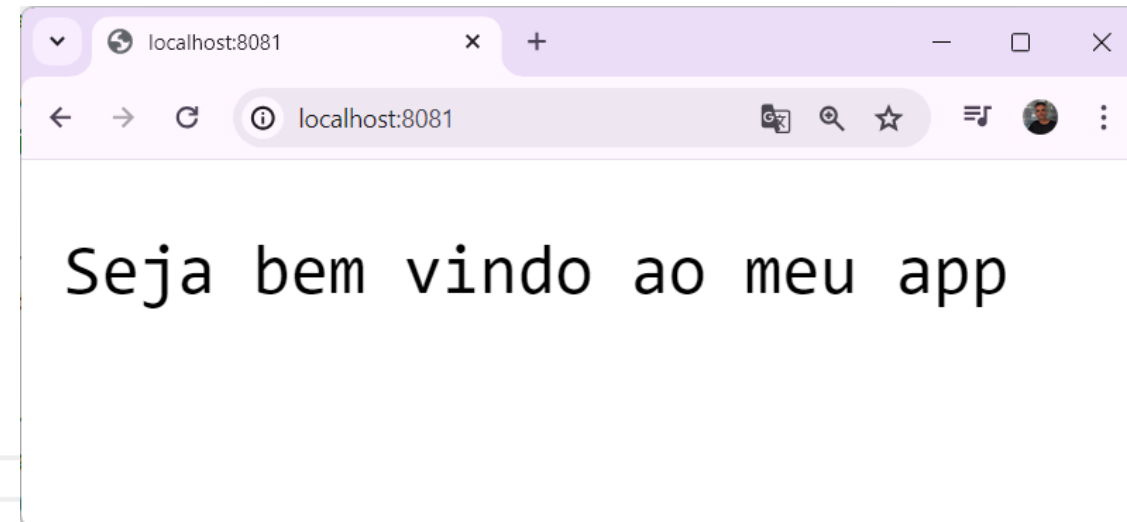


# CRIANDO UMA ROTA

- Vamos criar uma rota que será a principal de nossa aplicação.

03-Express > JS index.js > ...

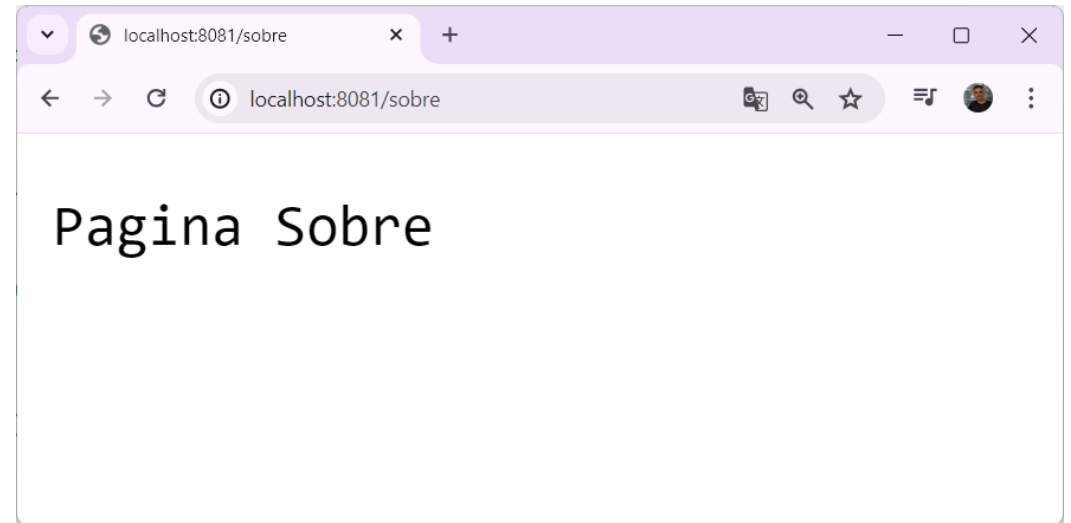
```
1  const express = require('express')
2  const app = express()
3  // a variavel app será uma instância do módulo express
4
5  // transformando as variaveis que recebem os módulos em constantes
6  // você evita que eles sejam reescritas durante a execução do
7  // seu programa....
8
9  app.get("/",function(req,res){
10 |   res.send("Seja bem vindo ao meu app")
11 | })
12
13 // Abrindo um servidor com express
14 // ESTA FUNÇÃO TEM QUE SER SEMPRE A ÚLTIMA LINHA DO SEU CÓDIGO
15 app.listen(8081,function(){
16 |   console.log("Servidor rodando na url http://localhost:8081")
17 | })
```



# CRIANDO OUTRAS ROTAS

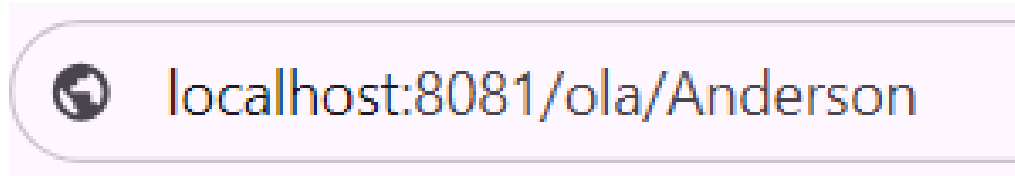
- Por exemplo vamos criar mais duas rotas
- <http://localhost:8081/sobre>
- <http://localhost:8081/blog>

```
8
9  app.get("/",function(req,res){
10  |   res.end("Seja bem vindo ao meu app")
11  | })
12
13  app.get("/sobre",function(req,res){
14  |   res.send("Pagina Sobre")
15  | })
16
17  app.get("/blog",function(req,res){
18  |   res.send("Pagina do Blog")
19  | })
20
21  // Abrindo um servidor com express
22  // ESTA FUNÇÃO TEM QUE SER SEMPRE A ÚLTIMA LINHA DO SEU CÓDIGO
```



# ROTAS DINÂMICAS – PARÂMETROS

- Vamos criar uma rota “*ola*” e a função dela será o de exibir o nome de uma pessoa.
- A ideia é que quando alguém digitar na url o seguinte:



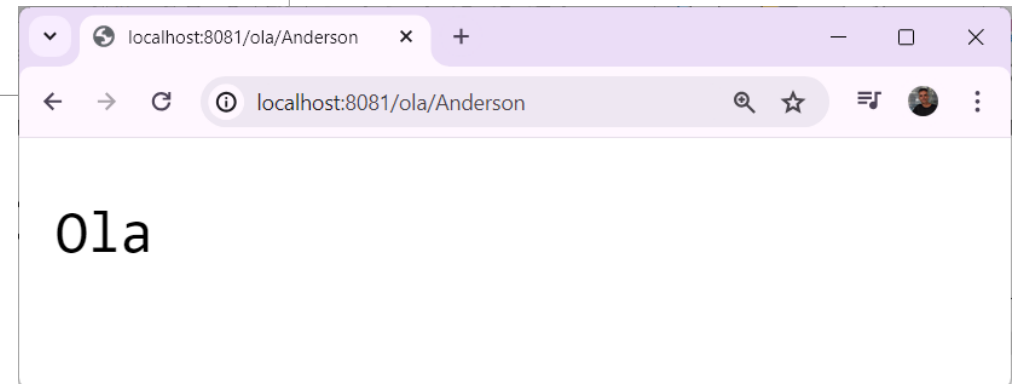
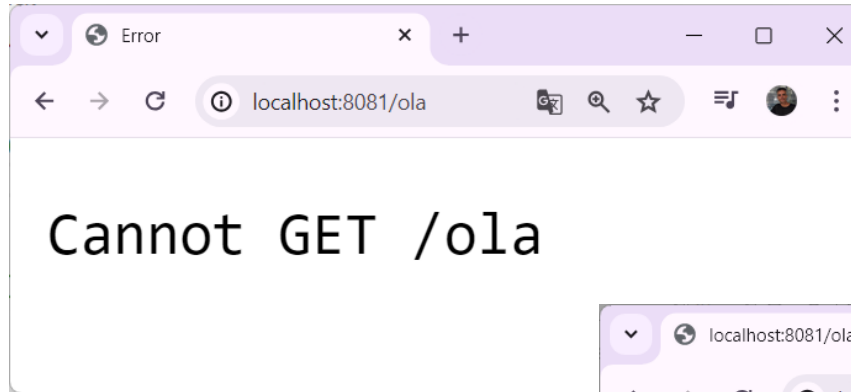
- O site exiba uma mensagem com o nome digitado.



# ROTAS DINÂMICAS – PARÂMETROS

```
21 app.get("/ola/:nome", function(req, res){  
22   |   res.send("Ola")  
23   | })
```

Como a rota foi criada com um parâmetro, é obrigatória a passagem de um valor na url da rota.



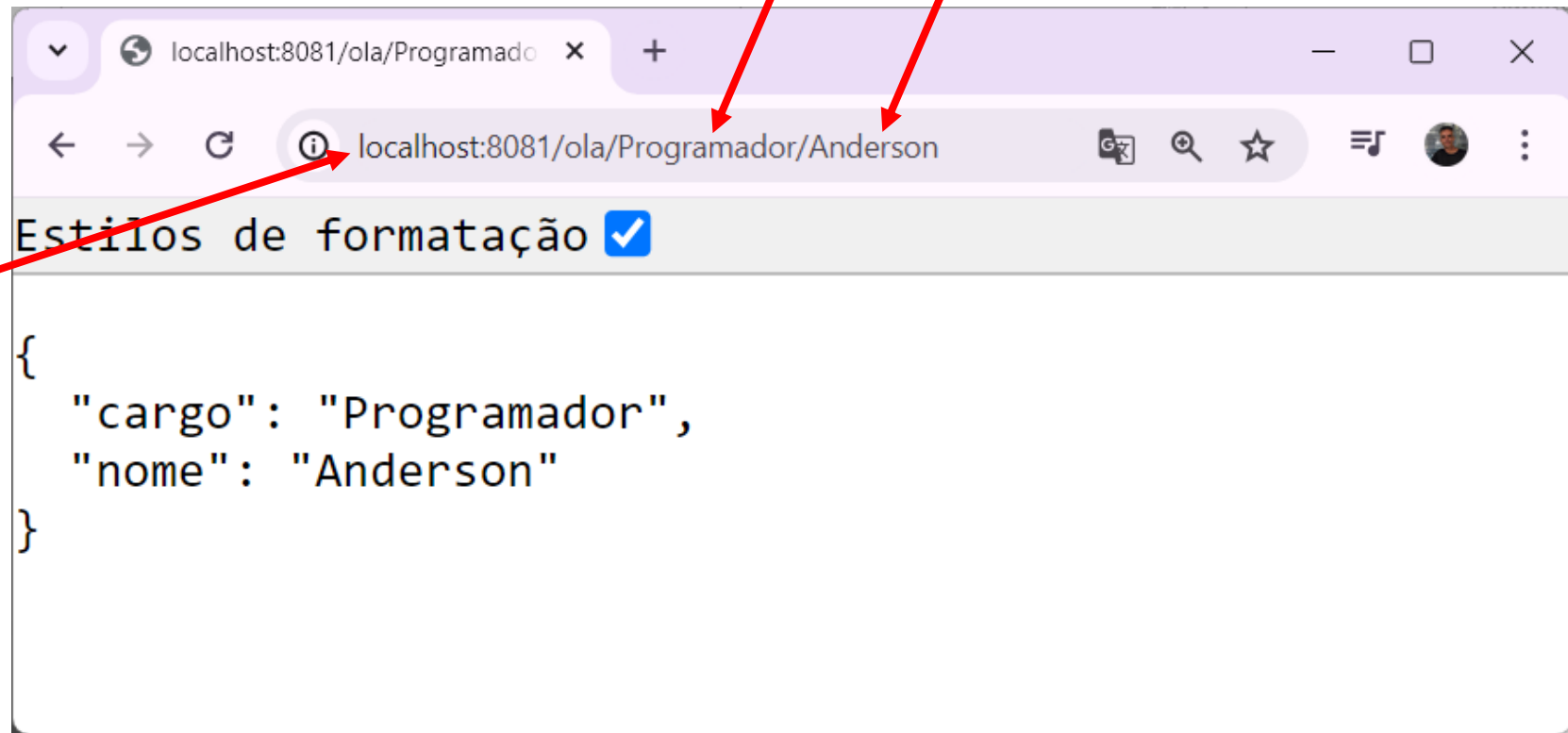
Você pode criar uma rota que contenha vários parâmetros separando-os com /

# ROTAS DINÂMICAS – PARÂMETROS

- Vamos criar a rota ola que irá receber dois parâmetros cargo e nome.

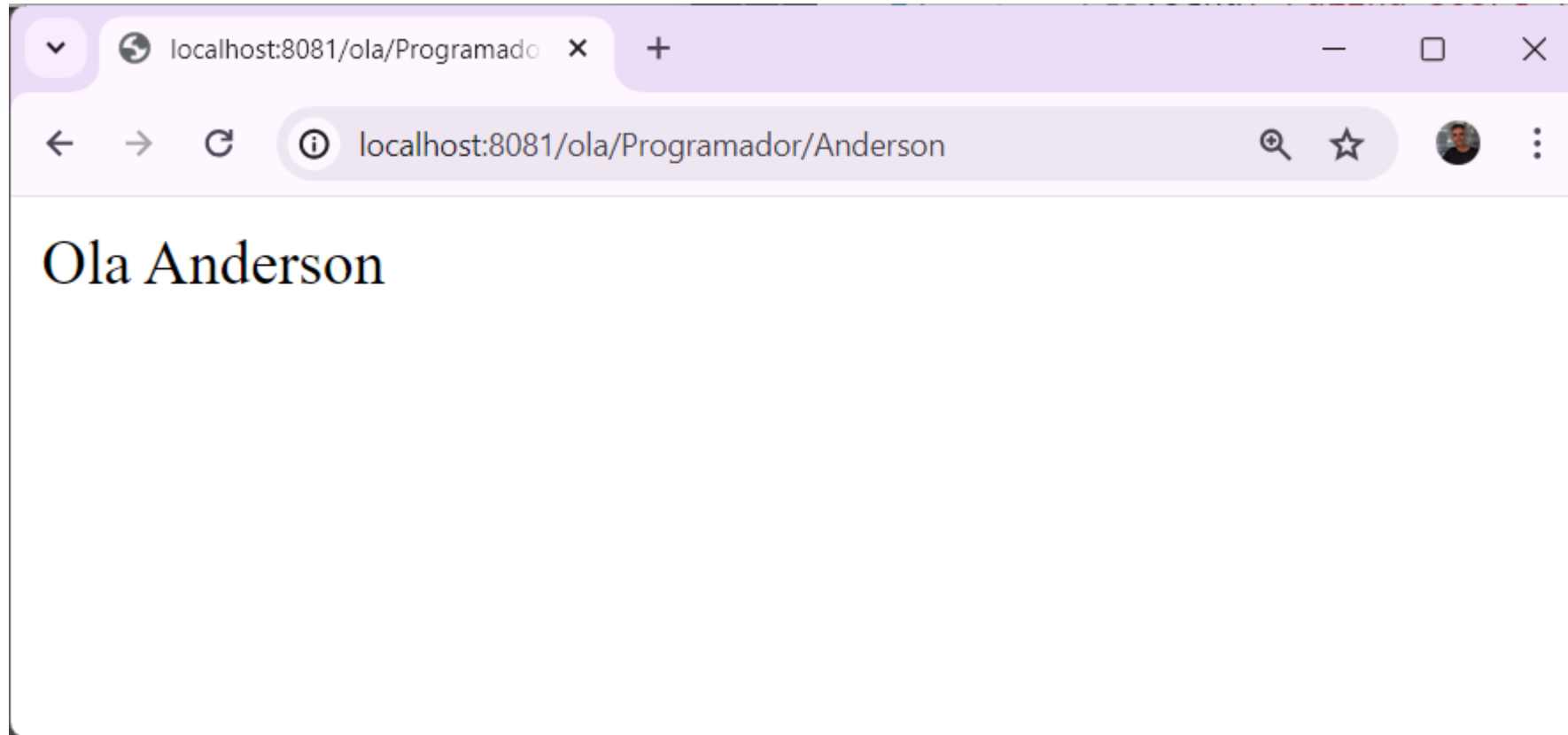
```
20
21 app.get("/ola/:cargo/:nome", function(req, res){
22   | res.send(req.params)
23 })
24
```

Aqui fizemos uma requisição ao servidor Node enviando dois parâmetros.




# ROTAS DINÂMICAS – PARÂMETROS

```
21 app.get("/ola/:cargo/:nome",function(req,res){  
22   |   res.send("Ola "+ req.params.nome)  
23   |   })  
24
```



# ROTAS DINÂMICAS – PARÂMETROS

- E para exibir os outros parâmetros?

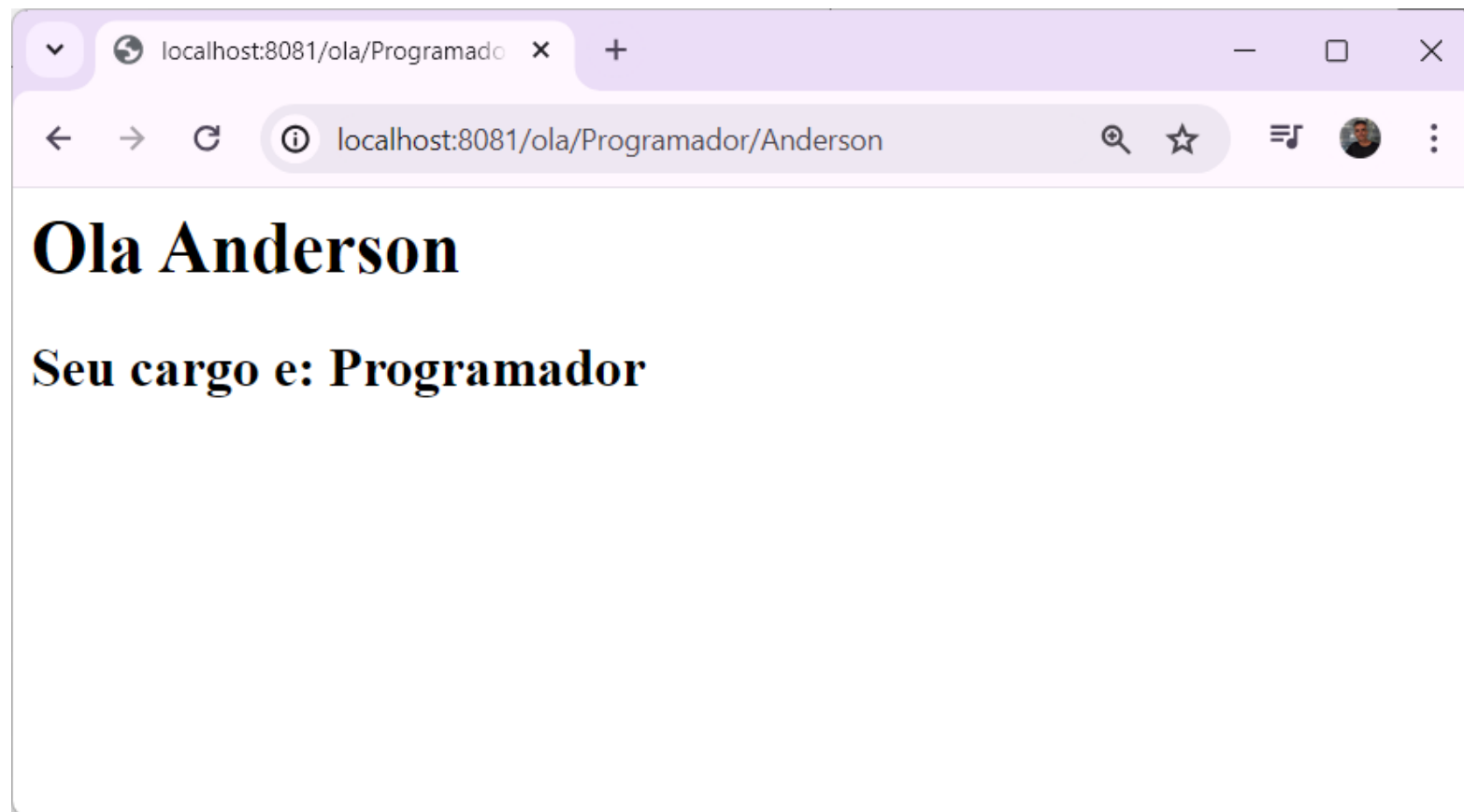


```
app.get("/ola/:c/:n",function(req,res){  
  res.send("<h1>ola "+req.params.nome+"</h1>")  
  res.send("<h2>ola "+req.params.cargo+" e: " + req.params.cargo+"</h2>")  
})
```

- ***res.send*** só pode ser chamado uma única vez!

# ROTAS DINÂMICAS – PARÂMETROS

```
app.get("/ola/:cargo/:nome",function(req,res){  
  res.send("<h1>Ola "+ req.params.nome+"</h1><h2> Seu cargo e: " + req.params.cargo+"</h2>")  
})
```



# NODEMON

- O módulo nodemon atualiza o servidor a cada alteração feita no código, sem a necessidade de parar o servidor e rodar novamente cada vez que você alterar seu código.
- Para instalar o nodemon digite:

***`npm install nodemon -g`***

- A flag **-g** significa que o nodemon vai ser instalado de forma global em seu sistema e não só em sua aplicação!

```
C:\Users\Anderson\Desktop\Curso_Node\03-Express>npm install nodemon -g  
  
added 29 packages in 3s  
  
4 packages are looking for funding  
run `npm fund` for details
```



# NODEMON

- Agora para rodar as suas aplicações utilize nodemon

***`nodemon index.js`***

```
C:\Users\Anderson\Desktop\Curso_Node\03-Express>nodemon index.js
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Servidor rodando na url http://localhost:8081
█
```

- Experimente fazer alguma alteração em alguma das rotas criadas anteriormente.

# NODEMON

```
8
9 app.get("/",function(req,res){
10   res.send("Seja bem vindo ao meu app - Agora fiz uma alteração")
11 })
12
13 app.get("/sobre",function(req,res){
14   res.send("Pagina Sobre")
15 })
16
17 app.get("/blog",function(req,res){
18   res.send("Pagina do Blog")
19 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C:\Users\Anderson\Desktop\Curso_Node\03-Express>nodemon index.js
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Servidor rodando na url http://localhost:8081
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Servidor rodando na url http://localhost:8081
```

