FIC 2 – Módulo II

Aula 10 – Criando rotas no Servidor Web utilizando o Express

Criação de páginas customizadas com Node.js

Leonardo Dalvi Pio

INSTITUTO FEDERAL
Espírito Santo

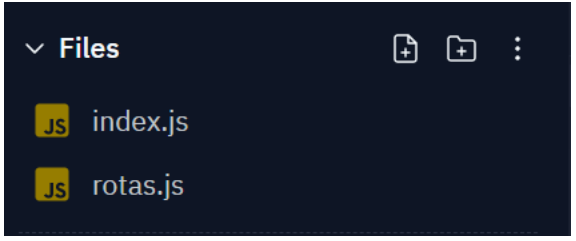
Criando rotas no Servidor Web utilizando o Express

- Criação de um servidor web.
- Criação de rotas no servidor web utilizando o *Framework Express*.
- Modularização (organização dos arquivos no servidor).
- No Replit, se o pacote/recurso (***pagkage.json***, ***express***) não estiver disponível, ele será baixado automaticamente.

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- No Replit, crie uma pasta nomeada “Aula 10”.
- Utilizaremos dois arquivos: ‘index.js’ e ‘rotas.js’.
- Dentro da pasta, crie um novo arquivo (New Repl). Selecione o Template ‘Node.js’ e atribua o título “aula10”. O arquivo ‘index.js’ será incluído em ‘Files’.
- Crie um novo arquivo nomeado ‘rotas.js’.



INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- No arquivo 'rotas.js', criaremos uma constante nomeada 'express', conforme código:

```
const express = require('express')
```

- Em seguida, criaremos uma constante chamada 'rotas' que receberá as rotas ('Router()') do módulo Express.

```
const rotas = express.Router()
```

- Criaremos uma variável (do tipo *JSON*) nomeada 'municipios' com as chaves: 'cidade' e 'info', incluindo o nome da cidade e a suas respectivas informações, conforme abaixo:

```
let municipios = [
  { 'cidade': 'vitoria', 'info': 'Vitória: Capital do ES' },
  { 'cidade': 'vilavelha', 'info': 'Vila Velha: Canela Verde' },
  { 'cidade': 'cachoeiro', 'info': 'Cachoeiro de Itapemirim: Princesa do Sul' },
  { 'cidade': 'colatina', 'info': 'Colatina: Princesa do Norte' }
]
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- Utilizando o comando 'get', em 'rotas', na raiz ('/'), criaremos uma *Arrow Functions* informando uma resposta do tipo 'json' (também poderia ser utilizado o comando 'write' - HTML) com: "Olá: Seja bem-vindo!"

```
rotas.get('/', (req, res) => {
  res.json({ Olá: 'Seja bem-vindo!' })
})
```

- Para adicionarmos parâmetros na nossa rota base, devemos incluir um novo caminho, que chamaremos de 'cidadeid'. Utilizaremos o comando 'get' novamente. Criaremos uma *Arrow Functions* para receber a requisição (req) e verificar a cidade informada.

Criando rotas com Express

- Criaremos uma constante 'cidade' que irá receber a requisição (req) do que foi digitado em 'cidadeid'.
- Criaremos também uma constante 'cidadeinfo' para receber a informação da cidade pesquisada. Utilizaremos o método 'find' para realizar a pesquisa em 'municipios' (apresentado no slide nº 4). Criaremos uma *Arrow Functions* (=>) nomeada 'i' para o parâmetro de entrada. Se a chave 'i.cidade' for igual à cidade informada no URL ('cidade'), ela retornará o conteúdo de 'municipios' (exibindo: cidade e a sua informação).
- O código é apresentado abaixo:

```
rotas.get('/:cidadeid', (req, res) => {
  const cidade = req.params.cidadeid
  const cidadeinfo = municipios.find(i => i.cidade == cidade)
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- Exemplo: Se a cidade “vitoria” for informada no URL, o código irá verificar, em ‘municipios’, se a cidade consta do nome pesquisado. Se a cidade informada for igual à chave (no caso: ‘cidade’), ela retorna a informação (da cidade pesquisada).

```
'cidade': 'vitoria'
```

```
'info': 'Vitória: Capital do ES'
```

- Em seguida, verificaremos, com um ‘if’, se o parâmetro informado se encontra na nossa lista de cidades (em ‘municipios’). Se uma cidade que não esteja presente em nossa lista for informada, uma mensagem de erro será exibida (conforme apresentado em HTTP Cat): 'Cidade não encontrada', juntamente com o nome da cidade pesquisada, conforme código

```
if (!cidadeinfo) {
  res.status(404).json(
    { erro: 'Cidade não encontrada!', cidadepesquisada: cidade }
  )
}
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- Se a pesquisa não apresentar erro, será exibido o status 200 (conforme apresentado em HTTP Cat) retornando a informação do nosso curso (que está em 'cidadeinfo').

```
else {
  res.status(200).json(cidadeinfo)
}
```

- Exportaremos o nosso módulo (para 'index.js') rotas através do comando abaixo:

```
module.exports = rotas
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- No arquivo 'index.js', criaremos uma constante nomeada 'express', conforme código:

```
const express = require('express')
```

- Em seguida, criaremos uma constante chamada 'rotas' informando o diretório do arquivo (criado anteriormente).

```
const rotas = require('./rotas')
```

- Utilizaremos a porta 443.

```
const porta = 443
```

- Criaremos uma constante chamada 'app' do 'express()'.

```
const app = express()
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

- Com o ‘app’ criado, utilizaremos o comando ‘use’ para aplicar o módulo de rotas a partir da ‘/’ (raiz).

```
app.use('/', rotas)
```

- Para finalizar, incluiremos o comando ‘.listen()’ e a porta de comunicação (porta 443) com o Servidor, e a mensagem: ‘Servidor rodando’ (com o comando: console.log) será apresentada.

```
app.listen(porta, () => { console.log('Servidor rodando') })
```

- Pronto! Já temos um Servidor Web utilizando o Framework Express!

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

Codificação final – **index.js**

```
1  const express = require('express')
2  const rotas = require('./rotas')
3  const porta = 443
4
5  const app = express()
6
7  app.use('/', rotas)
8
9  app.listen(porta, () => { console.log('Servidor rodando') })
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

Codificação final – **rotas.js**

```
1   const express = require('express')
2   const rotas = express.Router()
3
4   let municipios = [
5     { 'cidade': 'vitoria', 'info': 'Vitória: Capital do ES' },
6     { 'cidade': 'vilavelha', 'info': 'Vila Velha: Canela Verde' },
7     { 'cidade': 'cachoeiro', 'info': 'Cachoeiro de Itapemirim: Princesa do Sul' },
8     { 'cidade': 'colatina', 'info': 'Colatina: Princesa do Norte' }
9   ]
10
11  rotas.get('/', (req, res) => {
12    res.json({ Olá: 'Seja bem-vindo!' })
13  })
```

INSTITUTO FEDERAL
Espírito Santo

Criando rotas com Express

Codificação final (continuação) – **rotas.js**

```
14 rotas.get('/:cidadeid', (req, res) => {
15   const cidade = req.params.cidadeid
16   const cidadeinfo = municipios.find(i => i.cidade == cidade)
17
18   if (!cidadeinfo) {
19     res.status(404).json(
20       { erro: 'Cidade não encontrada!', cidadepesquisada: cidade }
21     )
22   }
23   else {
24     res.status(200).json(cidadeinfo)
25   }
26 })
27 module.exports = rotas
```

INSTITUTO FEDERAL
Espírito Santo

Testes!

- Para testar o código, inicie com o botão 'Run' do *Replit*.
- Na aba 'Console' é possível observar a mensagem: "Servidor rodando"!
- Na aba 'Webview' é possível observar a mensagem: "{"Olá":"Seja bem-vindo!"}"".
- Abra um nova aba do código no browser. (ex: https://APIdeRotas.usuario.repl.co/)
- Teste as rotas criadas informando o nome das cidades (já criadas em 'municipios' – rotas.js) no URL, conforme exemplo:
- https://APIdeRotas.usuario.repl.co/vitoria
- Teste também uma cidade que não foi criada.

INSTITUTO FEDERAL
Espírito Santo

INSTITUTO
FEDERAL
Espírito Santo