



## AULA 04 – Vídeo 04

### Desenvolvimento

Já configuramos as duas rotas que utilizam o método GET para os perfis. Uma requisição para '/perfil' retornará os cinco últimos perfis. Uma requisição para '/perfil/:id' retornará o perfil vinculado ao id passado como parâmetro. Agora, devemos configurar a rota de utiliza o método POST, que servirá para cadastrar um novo perfil. Nosso código deverá ficar assim:

```
app.post("/perfil", (req, res) => {
  let novoPerfil = req.body;
  console.log(req.body);
  if (novoPerfil) {
    novoPerfil.id = geradorID;
    perfis.push(novoPerfil);
    geradorID++;
    res.json(novoPerfil);
  } else {
    res.status(400).json({
      message: "Erro ao buscar perfil : Dados
incompletos"
    })
  }
});
```

Antes de começarmos, devemos ter em mente que os dados trafegados entre cliente e servidor estão em formato .json, e não como objeto node, com o qual

podemos trabalhar. Podemos configurar o express para fazer a conversão automática entre os formatos sempre que houver uma requisição. Para isso, usamos a seguinte instrução, escrita após a definição de nossa constante 'app'.

```
app.use(express.json());
```

Com isso, podemos e precisamos acessar os dados do novo perfil, que devem existir no *body* da requisição, pelo operador ponto. Colocamos as informações do novo perfil em uma variável denominada 'novoPerfil'.

Para que possamos cadastrar um novo perfil, este deve existir na requisição, o que deve ser verificado. Podemos fazê-lo com uma decisão condicional.

Vejamos, primeiramente, a cláusula *e/se*. Caso a requisição seja feita sem os dados do novo perfil, devolveremos o código de erro 400, junto com uma mensagem. O erro 500 indica um erro interno ao servidor. O erro 400 indica "*bad request*", um problema com a requisição. O código 200 indica que a requisição foi feita de maneira correta, mas pode aparecer quando algum dado passado corretamente é, por algum motivo, inválido.

Caso as informações do novo perfil tenham sido passadas corretamente, teremos um novo perfil, ainda sem o ID. Este deve ser colocado por nós. Para isso, podemos utilizar nosso 'geradorID', atribuindo seu valor ao novo perfil e atualizando-o. Por fim, podemos devolver como resposta o novo perfil acrescido do ID.

A seguir, devemos testar o código que acabamos de fazer. Para isso, devemos nos lembrar de mudar o método de requisição para POST. Para testar, fazemos uma requisição pelo Thuder Client, passando um objeto json com todas as informações do novo perfil para a rota '/perfil'. Notemos que a resposta, à direita na tela a seguir, já contém o id do novo perfil:

index.js

TC New Request X

POST http://localhost:3000/perfil

Send

Query

Headers 2

Auth

Body 1

Tests

Json

Xml

Text

Form

Form-encode

GraphQL

Binary

Json Content

```
1 {
2   "usuario": {
3     "email": "marcela@email.com",
4     "senha": "123334"
5   },
6   "nome": "Marcela Lopes",
7   "dataNascimento": "2022-02-14T00:00:00.000Z",
8   "disponibilidadeMudanca": true,
9   "disponibilidadeHorario": "Integral",
10  "educacao": [
11    {
12      "instituicao": "Escola 1",
13      "ingresso": "2012-02-28T00:00:00.000Z",
14      "conclusao": "2015-02-28T00:00:00.000Z",
15      "nivelEscolaridade": "Ensino Superior"
16    }
17  ],
18  "certificacoes": [
19    {
20      "instituicao": "High Tech Cursos",
21      "titulo": "Fábrica de Programador",
22      "cargaHoraria": 80
23    }
24  ],
25  "conexoes": []
26 }
```

Status: 200 OK Size: 473 Bytes Time: 32 ms

Response Headers 6 Cookies Results Docs New

```
1 {
2   "usuario": {
3     "email": "marcela@email.com",
4     "senha": "123334"
5   },
6   "nome": "Marcela Lopes",
7   "dataNascimento": "2022-02-14T00:00:00.000Z",
8   "disponibilidadeMudanca": true,
9   "disponibilidadeHorario": "Integral",
10  "educacao": [
11    {
12      "instituicao": "Escola 1",
13      "ingresso": "2012-02-28T00:00:00.000Z",
14      "conclusao": "2015-02-28T00:00:00.000Z",
15      "nivelEscolaridade": "Ensino Superior"
16    }
17  ],
18  "certificacoes": [
19    {
20      "instituicao": "High Tech Cursos",
21      "titulo": "Fábrica de Programador",
22      "cargaHoraria": 80
23    }
24  ],
25  "conexoes": [],
26  "id": 8
27 }
```