

Requisição POST

A requisição post é uma requisição que não é passada via navegador, ou seja, ela fica escondida proporcionando mais segurança às informações que são passadas.

Vamos exemplificar utilizando o projeto da aula 2.

Vamos alterar o arquivo **clienteRoute.js** para **Route.js**, logo após vá no arquivo index.js e altere o caminho da rota.

A linha abaixo deverá ser alterada.

```
var clienteRoute = require('./routes/clienteRoute');
```

Ficando assim:

```
var route = require('./routes/route');
```

Altere também o código.

```
app.use('/', clienteRoute);
```

para

```
app.use('/', route);
```

logo após no route.js crie uma lista conforme o código abaixo.

```
var cidades = ['Mimoso', 'Cachoeiro', 'Serra'];
```

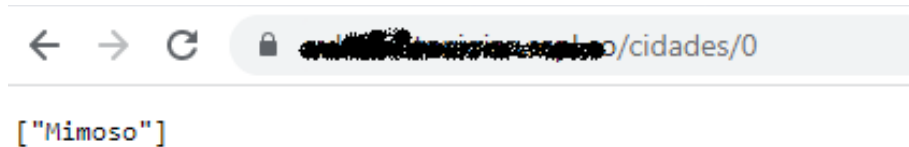
O comando acima está criando uma lista de cidades.

Em seguida vamos criar uma rota GET para consultar os dados deste array, utilizando o índice do vetor para acessar o valor de cada item da lista, da seguinte forma

```
router.get('/cidades/:id', (req, res) => {  
  let id = req.params.id;  
  return res.json([cidades[id]])  
});
```

Declaramos uma let chamada id para receber o req.params.id. O let é uma declaração que garante a utilização somente dentro do bloco que ela pertence.

Vamos executar nosso arquivo.



Repare que ao digitar /cidades/0, apareceu o valor que está na posição 0.

Agora vamos cadastrar novas cidades em nosso array utilizando o método POST, no Express podemos facilmente utilizar o **método app.post()**. Vamos criar uma rota e enviar o nome da nova cidade pela requisição post da seguinte forma:

Vamos adicionar a seguinte linha acima do app.use no arquivo index.js.

```
app.use(express.urlencoded({ extended: true }));
```

A linha acima deixa os dados da requisição de forma organizada para utilização pelo json.

Crie o método app.post() no arquivo Route.js

```
router.post('/cidades/cadastrar', (req, res) => {  
  let nome = req.body.nome;  
  cidades[cidades.length] = nome;  
  return res.json([cidades[cidades.length - 1]]);  
});
```

Agora iremos colocar no index.js abaixo da declaração **const porta** as informações sobre a rota do formulário e o handlebars.

```
const hand = require("express-handlebars");  
app.set('view engine', 'html');  
app.engine("handlebars", hand.engine());  
app.set("view engine", "handlebars");
```

Nosso arquivo index.js ficará assim:

```
const express = require("express");  
const app = express();  
const porta = 3000;
```

```
//CONFIGURAÇÃO DO HANDLEBARS
const hand = require("express-handlebars");
app.set('view engine', 'html');
app.engine("handlebars", hand.engine());
app.set("view engine", "handlebars");

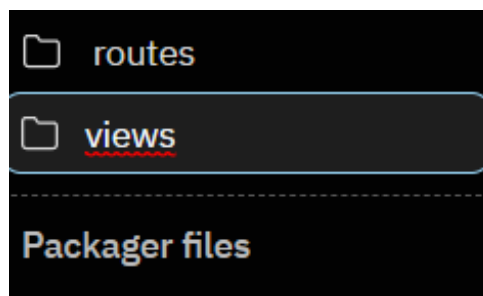
//ROTAS
var route = require('./routes/route');

//CONFIGURANDO A UTILIZAÇÃO DOS DADOS JSON
app.use(express.urlencoded({ extended: true }));

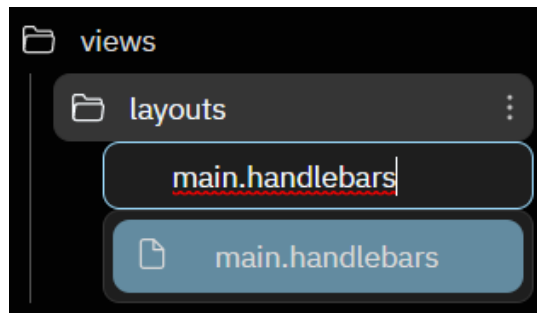
// CAMINHO DAS ROTAS
app.use('/', route);

//SERVIDOR
app.listen(3000, ()=>{
  console.log('Funcionando o servidor');
})
```

Temos que criar uma pasta views para utilização do handlebars.



Dentro da pasta views, crie a pasta layouts e o arquivo main.handlebars.



Precisamos ter a pasta views e layouts para utilização do handlebars.

O arquivo main é utilizado para a configuração inicial das views conforme vimos no módulo III.

Crie um formulário simples para teste de envio do post dentro da pasta views com o nome de **form.handlebars** e cole o arquivo abaixo.

```
<h1 class="text-center my-3">Cadastro de cidades</h1>
<form action="" method="">
  <div class="row w-50 d-block m-auto g-3">
    <div class="col-12">
      <label class="form-label" for="cidade">Nome da cidade:</label>
      <input
        class="form-control"
        type="text"
        name="nome"
        id="nome"
        placeholder="Digite o nome da cidade"
      />
    </div>

    <div>
      <input class="btn btn-primary my-3" type="submit" value="Cadastrar
cidade" />
    </div>
  </div>
</form>
```

Vamos incluir no arquivo de Route.js nossa rota para o formulário.

Vamos alterar a rota /cidades para enviar o render para o form.handlebars.

```
// ROTA principal cidade
router.get("/cidades", function(req, res) {
  res.render("form");
  //res.send("Seja bem vindo a rota de cidades");
});
```

Temos que configurar nosso main para receber nosso layout do form e incluir o bootstrap.

Main.handlebars

```
<html lang="pt-br">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  {{! BOOTSTRAP 5 }}
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1Hlcje39Wm4jDKdf19U8gl4ddQ3GYNS7NTKfAdVQSZe" crossorigin="anonymous"></script>
  {{!-- ARQUIVOS LOCAL --}}
<div class="container">
  {{{body}}}
</div>
</body>
</html>
```

Agora no form.handlebars, na linha 2 inclua o action="post" e no action o caminho para a rota de cadastro. A linha ficará assim.

```
<form action="/cidades/cadastrar" method="post">
```

Vamos verificar se a rota e o formulário estão funcionando executando nosso código.

at /cidades

Cadastro de cidades

Nome da cidade:

Cadastrar cidade

`["Venda Nova do Imigrante"]`

O método POST recebe a requisição do browser, realiza o tratamento e devolve para o navegador a informação utilizando o JSON.

Na próxima aula você verá como utilizar os parâmetros de URL.

Até lá...