

Entendendo o que são Rotas e seu funcionamento

Rotas

Uma rota é um caminho que será chamado por uma aplicação ou cliente e responderá com alguma informação caso ela exista.

Cada rota pode ter uma ou mais funções, deve ser única na aplicação e ao receber uma chamada faz todo o processamento necessário para retornar os dados que foi solicitado.

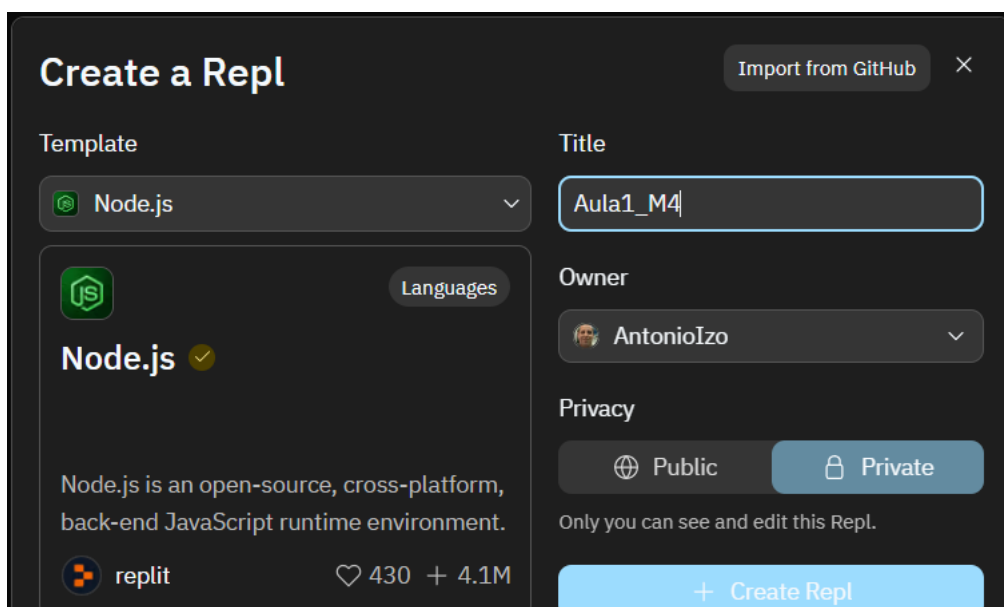
Por exemplo, uma rota na api chamada **/clientes**, terá funções para retornar uma lista de clientes. O **/clientes** é o caminho da identificação dessa rota.

O sistema pode até ter duas rotas chamadas **/clientes**, porém o tipo de solicitação para as rotas precisam ser diferentes, por exemplo uma pode ter o tipo GET e a outra o tipo POST.

As rotas são constituídas por 4 partes, são elas a instância do express, o método de requisição (GET, POST, PUT, DELETE) e a função que será executada:

```
app.METHOD(PATH, HANDLER)
```

Vamos codificar o primeiro exemplo de rotas.
Abra o repl.it e crie um novo projeto nodejs.



Uma rota como visto, é composta de 4 partes e uma rota principal.


A rota principal é a inicial do sistema e somente possui o “/”.

Vamos ver um exemplo utilizando o Olá Mundo.

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

app.get("/", function (req, res) {
  res.send("Olá Mundo!");
});

app.listen(porta, () => {
  console.log("Exemplo de uso de rotas.");
});
```



The screenshot shows a code editor on the left with the same JavaScript code as the previous block. On the right, a web browser window displays "Olá Mundo!". Below the browser, a terminal window shows the command prompt with the text "Exemplo de uso de rotas." and a hint to hit control+c to enter REPL.

Veja que a rota “/” foi identificada e retornou uma mensagem na tela.

O importante é que a rota localizou seu caminho.

Falando de caminho de rota, ele pode ser igual desde que tenha diferentes métodos de requisição para ela, e para declarar os diferentes tipos no express, se fazemos da seguinte forma.

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

app.get("/clientes", function (req, res) {
  res.send("Lista de Clientes");
});

app.post("/clientes", function (req, res) {
  res.send('Inserir um cliente');
});
```

Veja que há duas rotas /clientes, porém uma delas é para listar todos os usuários e a outra é para inserir um usuário.

Conforme as rotas de um sistema crescem, é uma boa prática separar essas rotas em arquivos, imagine 100 rotas no mesmo arquivo, dificultaria muito a manutenção.

Para fazer isso existe o **express.Router**, com ele é possível escrever todas as rotas em um arquivo e depois exportar para que seja usado no arquivo principal, com isso as rotas de um módulo ficam separadas.

Vamos praticar.

Crie uma pasta chamada routes e dentro dela crie um arquivo chamado clienteRoute.js. O clienteRoute.js será responsável por realizar a importação das rotas, para isso, adicione o seguinte código no clienteRoute.js:

```
1 var express = require('express');
2 var router = express.Router();
3
4 router.get('/clientes', function(req, res) {
5   res.send("Lista todos os clientes");
6 });
7
8 module.exports = router;
```

clienteRoute.js

```
var express = require('express');
var router = express.Router();

router.get('/clientes', function(req, res) {
  res.send("Lista todos os clientes");
});

module.exports = router;
```

No arquivo index.js vamos retirar a rota a rota "/".

```
app.get("/", function (req, res) {
  res.send("Olá Mundo!");
});
```

Testando.

Cannot GET /

Repare que o sistema informa que não localizou uma rota para o "/". Está correto!

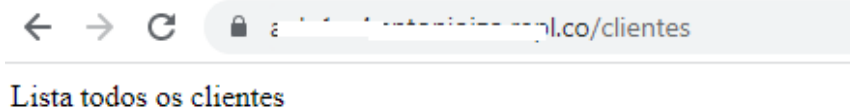
Ainda não realizamos uma rota para ele.

Dentro do arquivo index.js

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

Ao entrar na rota principal no index.js, será chamado a variável clienteroute do arquivo clienteRoute.js.

Vamos agora testar a rota que criamos /clientes.



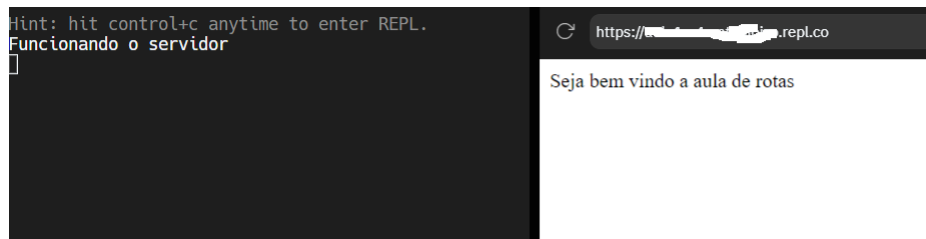
Veja que agora aparece o resultado da nossa rota.

Vamos criar uma rota para a o "/" e assim retirar o erro de não localizado da rota.

Dentro do arquivo clienteRoute.js crie a rota para o "/" que chamaremos de rota principal.

```
// ROTA PRINCIPAL  
router.get('/', function(req, res) {  
  res.send("Seja bem vindo a aula de rotas");  
});
```

E a nossa rota principal retornará um **res.send** com a mensagem de seja bem vindo para nosso usuário.



Nossas rotas estão funcionando em um arquivo diferente.

Na próxima aula você vai conhecer sobre os parâmetros dinâmicos.

Até lá...