

Documentação da Aula 2 e Aula 2B NodeJS – Criando Rotas no Servidor

Nessa aula de hoje aprendemos como criar rotas no servidor, para que possamos retornar com uma resposta diferente a cada tipo de requisição. Foi usado puramente o **Nodejs** sem nenhum uso de **Framework**.

Uma rota é um caminho que será chamado por uma aplicação ou cliente e **responderá** alguma **informação**. Ou seja eu posso oferecer caminhos ou conteúdos diferente na URL para identificar uma navegação diferente, então por exemplo, posso identificar uma rota no conteúdo para uma **rota raiz** um conteúdo para rota **Home**. Assim quando a rota for requisitada ela irá abrir um conteúdo relacionado ao pedido ou um conteúdo diferente do navegação.

Analizando o Código A:

```
const http = require('http');
```

 aqui temos a solicitação da Função require para utilizar um módulo ou Biblioteca.

```
const porta = 3000;
```

```
as variáveis const que vão ser utilizadas para o uso do localhost e da porta de entrada
```

```
const host = '127.0.0.1';
```

```
foi criada uma função para usar as rotas percebe-se que dentro da função tem as variáveis req e res juntamente com a sintaxe writeHead vai ser usada para definir os cabeçalhos de uma resposta HTTP, logo também foi utilizado um cód HTML apenas para deixar o texto grande as condições if e else if carregam a var req e dentro dessa condição tem a res. write que serve justamente para passar a informação na tela do navegador
```

```
const servidor = http.createServer((req,res)=>{  
  res.writeHead(200,{ 'Content-Type': 'text/html' });  
  if(req.url== '/') {  
    res.write('<h1>Seja bem vindo</h1>');  
  } else if(req.url== '/canal') {  
    res.write('<h1>CFB Cursos');  
  } else if(req.url== '/curso') {  
    res.write('<h1>Curso de Node</h1>');  
  } else if(req.url== '/curso/node') {  
    res.write('<h1>Curso de Node</h1>')  
  }  
  res.end(); aqui temos a finalização da função  
});
```

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

 server.listen serve para iniciar o servidor que escuta conexões em uma porta especifica logo é a porta de entra para iniciar ou executar o código.

Analizando Código B:

```
const http = require('http');  
const url = require('url');  
const porta = 3000;  
const host = '127.0.0.1';
```

aqui temos a solicitação da Função require para utilizar um módulo ou Biblioteca só que dessa vez usando 2.
as variáveis const que vão ser utilizadas para o uso do localhost e da porta de entrada

foi criada uma função para usar as rotas percebe-se que dentro da função tem as variáveis req e res juntamente com a sintaxe writeHead vai ser usada para definir os cabeçalhos de uma resposta HTTP. A grande diferença é a variável url.parse que descreve o ato de analisar ou interpretar uma sequência de caracteres em formato estruturado. Logo temos a variável p que vem com o termo query para referir a uma solicitação que é feita para obter fonte de dados. Assim ele fica dentro dos parentes com a variável res.write seguindo a com código HTML.

```
const servidor = http.createServer((req,res)=>{  
  res.writeHead(200,{ 'Content-Type': 'text/html' });  
  res.write(req.url);  
  const p=url.parse(req.url,true).query;  
  res.write("</b>",p.nome);  
  res.write("</br>",p.curso);  
  res.end();  
});
```

aqui temos a finalização da função

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

server.listen serve para iniciar o servidor que escuta conexões em uma porta específica logo é a porta de entrada para iniciar ou executar o código.