



AULA 04 – Vídeo 01

Introdução

Na última aula, documentamos a estrutura básica de nosso projeto em um arquivo README.md. Devemos nos lembrar de que essa documentação não é definitiva, e alguns aspectos do projeto ainda serão apresentados e deverão ser documentados.

Definidos nossos *endpoints*, começaremos, agora, a adicionar nossa camada de rotas. Recapitulando, já adicionamos o *express.js* em *package.json*, e já o importamos em nosso código. Além disso, também já instanciamos o *express* na constante 'app' e definimos uma porta. 'app' lida, portanto, com as requisições HTTP. Para registrarmos uma rota, podemos utilizar o método 'get()'. Utilizamos-lo para responder a requisições feitas à raiz do projeto. Por meio de uma função de callback, cujos argumentos são o objeto requisição e o objeto resposta, enviamos a resposta. Por fim, fora chamado o método 'listen()', que tem como parâmetros a porta em que a API será servida e uma função de callback a ser chamada na inicialização do servidor.

Com tudo isso em mãos, mudamos a mensagem enviada na resposta às requisições à raiz. Mudamos também a mensagem a ser colocada em console na função de callback passada para o método 'listen'. Nosso código ficou, portanto, assim:

```
const express = require("express");

const app = express();
const port = 3000;
```

```
app.get("/", (req, res) => {
    res.send("Bem-vindo(a) ao Perfil Profissional API");
});

app.listen(port, () => {
    console.log(`Perfil Profissional API rodando na
porta ${port}...`)
});
```

Como podemos fazer requisições? Há algumas opções. Em nossas aulas, utilizaremos a extensão para VSCode Thunder Client, cuja principal vantagem é operar dentro do VSCode. Você pode utilizar a opção que preferir.