03

Encapsulando o Express

Transcrição

Nessa etapa tentaremos melhorar o *design* do código da nossa aplicação. Vamos relembrar o que fizemos até agora:

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

const app = express()

app.listen(3000, function() {
    console.log('Servidor rodando na porta 3000');

});

app.get('/', function(req, resp) {
    resp.send(
        `
            <html>
                <head>
                    <meta charset="utf-8">
                </head>
                <body>
                    <h1> Casa do Código </h1>
                </body>
            </html>
        `
    );
});

app.get('/livros', function(req, resp) {
    resp.send(
        `
            <html>
                <head>
                    <meta charset="utf-8">
                </head>
                <body>
                    <h1> Listagem de livros </h1>
                </body>
            </html>
        `
    );
});
```

Nós trouxemos o módulo `express` para o nosso código com o `require('express')`, e em seguida chamamos a função `express()`. Com o método `listen()`, criamos o servidor, definindo duas rotas com o método GET para nossa aplicação.

Repare que estamos fazendo três coisas diferentes no mesmo arquivo:

- a importação do `express`

- a criação do servidor
- a definição das rotas

Segundo um princípio da programação, nossos códigos precisam ter responsabilidades únicas. Portanto, nosso objetivo é separar essas responsabilidades.

Primeiramente, criaremos um módulo customizado que importe o `express` e execute a função `express()`, atribuindo seu retorno à uma constante `app`. Para isso, dentro da pasta "config" (que está dentro da pasta "src"), criaremos o arquivo `custom-express.js`.

Passaremos para esse novo arquivo os trechos do código referentes às operações citadas:

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

const app = express();
```

Em `server.js`, precisaremos passar o caminho do módulo que foi criado. Isso é feito com `./`, indicando a raiz do projeto, seguido do resto do caminho:

```
const app = require('./src/config/custom-express');
```

Se executarmos nosso código dessa maneira, receberemos um erro: o Node não conseguirá identificar o método `listen()` no objeto `app`. Isso acontece pois em nenhum momento exportamos esse objeto do `express` no nosso módulo customizado! Para corrigirmos esse problema, basta escrevermos as exportações desse módulo no arquivo:

```
module.exports = app;
```

Dessa forma, nosso `server.js` voltará a ser executado corretamente e poderá ser acessado pelo navegador. Com isso, conseguimos encapsular toda a configuração do `express` em um único arquivo - a primeira parte das melhorias de *design* que faremos no nosso código.

A seguir, vamos retirar as rotas do arquivo `server.js`, o que requer uma solução um pouco mais rebuscada.