

Arquivos estáticos

Transcrição

Ainda temos um problema pendente na nossa aplicação: não conseguimos remover o livro da nossa listagem da maneira que esperávamos.

Failed to load resource: the server responded with a status of 404 (Not Found)

Refused to execute script from '<http://localhost:3000/remove-livro.js>' (<http://localhost:3000/remove-livro.js>) because its MIME type ('text/html') is not executable, and strict MIME type checking is enabled

O próprio navegador nos informa que, na requisição, o recurso `remove-livro.js` não foi encontrado. Lembre-se de que, para toda página que queremos retornar para o navegador, estamos criando uma rota na aplicação. Porém, em nenhum momento criamos uma rota para esse arquivo estático de JavaScript.

Desse modo, precisamos que nossa aplicação verifique se as requisições que ela recebe são referentes a um arquivo estático. Em caso positivo, ela precisa tratá-las de maneira a conseguir servir esses arquivos para o navegador (ou qualquer outro cliente que esteja acessando-a).

De volta ao nosso código, criaremos, dentro da pasta "app", uma pasta "public" na qual ficarão os arquivos estáticos da nossa aplicação. Dentro dela, criaremos outra pasta, chamada "js", em que serão alocados especificamente os arquivos de JavaScript. Também poderíamos ter uma pasta "img" para as imagens, "CSS" para estilos, e assim por diante.

Moveremos o arquivo `remove-livros.js` para a recém-criada pasta "js". Precisaremos, então, filtrar as requisições que chegam à nossa aplicação. Já conhecemos um ótimo recurso para isso: os *middlewares* da plataforma Node.

No arquivo `custom-express.js`, após a declaração do `bodyParser`, começaremos declarando o método `app.use()`. Para ele, passaremos, como primeiro parâmetro, uma URL que será usada para ativar o nosso *middleware*. Queremos que o `express` faça essa ativação sempre que encontrar a url `/estatico`, um identificador que indicará para nossa aplicação que está sendo requerido um recurso estático.

Como segundo parâmetro, passaremos a configuração do *middleware* a ser executado. A boa notícia é que o próprio objeto `express` possui uma configuração própria para arquivos estáticos, e somente é necessário chamá-la com `express.static()`.

Em seguida, indicaremos o diretório dos arquivos estáticos da nossa aplicação, passando uma string `src/app/public`:

No arquivo `lista.marko`, alteraremos a tag `<script>` para que reflita a localização do `remove-livro.js`:

```
<script src="/estatico/js/remove-livro.js">
</script>
```

Salvando essas alterações, reiniciaremos a aplicação e acessaremos a URL <http://localhost:3000/livros> (<http://localhost:3000/livros>). Nela, finalmente poderemos remover os livros da lista com sucesso! A seguir, implementaremos a próxima funcionalidade da nossa aplicação, que é a edição dos nossos livros.

