

Listagem estática de livros

Transcrição

Nesta etapa, começaremos a criar uma listagem de livros. Para isso, usaremos o `marko.js`, bastante conhecido no mundo Node, que serve para gerar templates HTML. Inclusive, ele funciona muito bem com o `express`, como podemos conferir na [documentação do Marko \(https://markojs.com/docs/express/\)](https://markojs.com/docs/express/).

No terminal, executaremos o seguinte comando:

```
npm install marko@4.13.4-1 --save-exact
```

Dessa forma, instalaremos uma versão específica do Marko, que será listado, no `package.json`, como uma das dependências da nossa aplicação. Para habilitarmos essa biblioteca na nossa aplicação, precisaremos adicionar duas linhas de código no arquivo `custom-express.js`, onde configuramos o `express` no nosso projeto:

```
require('marko/node-require').install();  
require('marko/express');
```

Na pasta "app", criaremos uma nova pasta chamada "views", na qual armazenaremos todos os arquivos de visualização do nosso sistema. Nela, criaremos outra pasta, chamada "livros", onde ficarão todas as páginas referentes ao domínio de livros (listagem, adição, remoção e assim por diante).

Dentro dela, teremos outra subpasta, chamada "lista", na qual criaremos um arquivo `lista.marko`. Todos os arquivos com a extensão `.marko` (que funcionam com o Marko) serão compostos por puro HTML. No `rotas.js`, removeremos o código HTML que criamos para `/livros`, e colaremos nesse novo arquivo, adicionando uma simples tabela com duas colunas e um único elemento:

```
<html>  
  <head>  
    <meta charset="utf-8">  
  </head>  
  <body>  
    <h1> Listagem de livros </h1>  
  
    <table>  
      <tr>  
        <td>ID</td>  
        <td>Título</td>  
      </tr>  
      <tr>  
        <td>1</td>  
        <td>Fundamentos do Node</td>  
      </tr>  
    </table>  
  </body>  
</html>
```

No `rotas.js`, estávamos criando um HTML fixo dentro do método `resp.send()`. Agora, queremos acessar o conteúdo armazenado em `lista.marko`. Para isso, executaremos o método `resp.marko()`, que foi habilitado com a inclusão do Marko no nosso projeto. Com ele, poderemos exibir arquivos `.marko` para o cliente na resposta, sendo necessário apenas importar o arquivo com essa extensão, o que também é feito com o método `require()`.

Nesse caso, teremos:

```
app.get('/livros', function(req, resp) {  
  resp.marko(  
    require('../views/livros/lista/lista.marko')  
  );  
});
```

Salvando todas as alterações no nosso código, poderemos acessar <http://localhost:3000/livros> (<http://localhost:3000/livros>) no navegador, recebendo como resposta:

Listagem de livros

ID Título

1 Fundamentos do Node

Porém, não é só isso! Queremos passar dados para o *template* Marko de modo que eles simplesmente sejam listados. Por exemplo, passar uma lista de livros para a página e fazer com que o Marko gere a listagem de livros automaticamente. Faremos isso a seguir!