

Finalizando a aplicação

Assistindo a última aula, nossa colega Joana descobriu o motivo e a solução do nosso pequeno enigma sobre a misteriosa inclusão de um novo livro quando na verdade ele deveria ser alterado. E aí ela fez o seguinte código:

```
// custom-express.js

const express = require('express');
const app = express();
const bodyParser = require('body-parser');
const methodOverride = require('method-override');

app.use('/estatico', express.static('src/app/public'));

app.use(methodOverride((req, res) => {

  if (req.body && typeof req.body === 'object' && '_method' in req.body) {
    const method = req.body._method;
    delete req.body._method;

    return method;
  }
}));

app.use(bodyParser.urlencoded({
  extended: true
}));

<!-- form.marko -->

<html>
  <body>
    <h1>Cadastro de livros</h1>

    <form action="/livros" method="post">

      <div if(data.livro.id)>
        <input type="hidden" name="_method" value="PUT">
        <input type="hidden" name="id" value="${data.livro.id}">
      </div>

      <div>
        <label for="titulo">Titulo:</label>
        <input type="text" id="titulo" name="titulo" value="${data.livro.titul
      </div>
      <div>
        <label for="capa">Capa:</label>
        <input type="file" id="capa" name="capa" placeholder="imagem de capa"/
      </div>
      <div>
        <label for="preco">Preço:</label>
```

```
        <input type="text" id="preco" name="preco" placeholder="150.25" value=
    </div>
    <div>
        <label for="descricao">Descrição:</label>
        <textarea cols="20" rows="10" id="descricao" name="descricao" placehc
    </div>

    <input type="submit" value="Salvar"/>
    </form>
</body>
</html>
```

```
// rotas.js

module.exports = (app) => {

    // demais rotas.

    app.put('/livros', function(req, resp) {

        const livroDao = new LivroDao(db);

        livroDao.atualiza(req.body)
            .then(resp.redirect('/livros'))
            .catch(erro => console.log(erro));

    });
};
```

Desse modo, avaliando uma última vez e dando uma mãozinha pra nossa colega, podemos afirmar:

Selecione uma alternativa

A O código não irá funcionar como esperado, pois, como visto durante o curso, a ordem da definição dos *middlewares* é importante e altera o resultado. E nossa colega Joana incorretamente definiu o *middleware* do *method-override* antes do *middleware* do *body-parser*, fazendo exatamente o oposto ao que é dito na documentação do *method-override*.

B O código não irá funcionar, pois, no template `form.marko`, o correto seria fazer:

```
<div if(livro.id)>
    <input type="hidden" name="_method" value="PUT">
    <input type="hidden" name="id" value="{data.livro.id}">
</div>
```

C O código está correto e irá alterar o livro como esperado.