

Detalhando livros

Pedro sempre gosta de ir um pouco mais além e desenvolver funcionalidades criadas por ele para testar o que aprendeu! Então ele resolveu criar a funcionalidade de detalhamento de livros! Que consiste em criar uma página para detalhar os livros da base de dados. Para isso, ele fez as alterações a seguir!

```
// src/app/rotas/rotas.js

module.exports = (app) => {

  // demais rotas omitidas.

  app.get('/livros/:id', livroControlador.detalha());
};

// src/app/controladores/livro-controlador.js

const LivroDao = require('../infra/livro-dao');
const db = require('../../config/database');

class LivroControlador {

  // demais métodos omitidos.

  detalha() {
    return function(req, resp) {
      const id = req.params.id;
      const livroDao = new LivroDao(db);

      livroDao.buscaPorId(id)
        .then(livro =>
          resp.marko(
            require('caminho para o template de detalhes do livro'),
            { livro: livro }
          )
        )
        .catch(erro => console.log(erro));
    };
  }
}

module.exports = LivroControlador;
```

Supondo que o nosso colega tenha criado corretamente o template *marko* para o detalhamento de livros e, também corretamente, inserido o caminho para esse template no `require()` do método `detalha()`, podemos afirmar corretamente:

Selecione 2 alternativas

A

O código funcionará corretamente e irá apresentar a página de detalhamento de livros quando o usuário acessar a rota `/livros/:id`, onde `:id` é um *path param* representando o *id* do livro a ser detalhado.

B

O código não funcionará pois faltou criar uma chave no método `rotas()` da classe `LivroControlador`, mantendo o encapsulamento das URLs, como visto durante a aula.

C

O código não funcionará pois faltou passar o `id` do livro a ser detalhado para o método `detalha()`.

D

O ideal é que nosso colega não tivesse utilizado uma string para representar a nova rota criada.