

Não gostou da foto? Retoque-a!

Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://s3.amazonaws.com/caelum-online-public/mean-js/stages/08-alurapic.zip\)](https://s3.amazonaws.com/caelum-online-public/mean-js/stages/08-alurapic.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Só não esqueça de baixar as dependências do projeto no terminal com o comando `npm install`.

Novo verbo, nova ação

Precisamos completar a funcionalidade do nosso servidor suportando alterações de foto. Até agora, usamos o verbo GET para obtermos um recurso do servidor, DELETE para apagarmos e POST para incluírmos. Quando queremos realizar uma alteração, é comum o uso do verbo PUT. É claro, você já sabe que o Express possui uma função para cada um desses verbos.

Vamos criar nossa rota de atualização:

```
module.exports = function(app) {  
  
  var api = app.api.foto;  
  
  // mudando a declaração da rota e adicionando suporte ao verbo POST  
  app.route('/v1/fotos')  
    .get(api.lista)  
    .post(api.adiciona);  
  
  app.route('/v1/fotos/:id')  
    .get(api.buscaPorId)  
    .delete(api.removePorId)  
    .put(api.atualiza);  
};
```

Recebendo duas encomendas

Veja que chamamos a função `put` para a rota `/v1/fotos/:id`. A razão disso é que nossa aplicação enviará o ID do recurso que precisa ser atualizado e um JSON com os dados da atualização.

Agora, vamos elaborar uma ação para atualização:

```
var CONTADOR_ID = 2;  
  
var fotos = [  
  { _id: 1, titulo: 'Leão', url: 'http://www.fundosanimais.com/Minis/leoes.jpg' },  
  { _id: 2, titulo: 'Leão 2', url: 'http://www.fundosanimais.com/Minis/leoes.jpg' }  
];  
  
var api = {};  
  
api.lista = function(req, res) {  
  
  res.json(fotos);  
};
```

```
api.buscaPorId = function(req, res) {

    var foto = fotos.find(function(foto) {
        return foto._id == req.params.id;
    });

    res.json(foto);
};

api.removePorId = function(req, res) {

    fotos = fotos.filter(function(foto) {
        return foto._id != req.params.id;
    });

    res.sendStatus(204);
};

api.adiciona = function(req, res) {

    var foto = req.body;
    foto._id = ++CONTADOR_ID;
    fotos.push(foto);
    res.json(foto);
};

api.atualiza = function(req, res) {

    var fotoId = req.params.id;
    var foto = req.body;

    var indice = fotos.findIndex(function(foto) {
        return foto._id == fotoId;
    });

    fotos[indice] = foto;
    res.sendStatus(200);

};

module.exports = api;
```

Pesquisando e atualizando

Procuramos a foto que desejamos atualizar por seu ID na lista de fotos através da função `findIndex`. Usamos seu índice para acessar a foto no array substituindo-a pelos dados que também recebemos na requisição. No final, dados como resposta para o servidor o código de status 200, indicando que a requisição foi executada sem qualquer problema. Vamos testar? Perfeito! Conseguimos atualizar nossas fotos.

Chegou a hora de deixarmos de trabalhar com dados voláteis para trabalhar com dados persistentes, interagindo com um banco de dados no próximo capítulo!

