

Consolidando o cordão umbilical

Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://s3.amazonaws.com/caelum-online-public/mean-js/stages/11-alurapic.zip\)](https://s3.amazonaws.com/caelum-online-public/mean-js/stages/11-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`.

Busca

Fizemos uma integração parcial entre nosso backend Node e nosso banco MongoDB. Falta implementarmos as operações restantes. A próxima que atacarei será `buscaPorId`:

```
// alurapic/app/api/fotos.js
// código anterior omitido

api.buscaPorId = function(req, res) {

  model.findById(req.params.id)
  .then(function(foto) {
    if (!foto) throw new Error('Foto não encontrada');
    res.json(foto);
  }, function(error) {
    console.log(error);
    res.sendStatus(500);
  });
};

// código posterior omitido
```

Usei a função `model.findById` que recebe o ID da foto enviada na requisição. Quando o documento é encontrado a partir do seu ID, temos acesso a ele na função passada para `.then` da nossa promise. Mas atenção: se o documento não existir, a função receberá `undefined`. É por isso que precisamos testar na função de sucesso da promise: se for `undefined`, lançamos um `Error`. Isso fará com que o callback de fracasso da promise seja chamado. Um teste reiniciando o servidor e acessando uma foto indica que tudo funciona.

Exclusão

O passo seguinte é implementarmos nossa API de exclusão:

```
// alurapic/app/api/fotos.js
// código anterior omitido

api.removePorId = function(req, res) {

  model.remove({ '_id' : req.params.id})
  .then(function() {
    res.sendStatus(200);
  }, function(error) {
    console.log(error);
    res.sendStatus(500);
  });
};
```

```
};

// código posterior omitido
```

Para removermos um documento, usamos a função `model.remove` que recebe como primeiro parâmetro um JSON contendo o critério da remoção. No caso, estamos procurando um documento cujo `_id` seja o valor de `req.params.id`. Como resposta, escrevemos `res.sendStatus(200)` indicando apenas que tudo foi OK, pois não queremos enviar nada. Quando algum erro inesperado acontece, enviamos o código de erro `500`.

Agora vamos focar na inclusão de novos fotos:

Adição

```
// alurapic/app/api/fotos.js
// código anterior omitido

api.adiciona = function(req, res) {

  model.create(req.body)
    .then(function(foto) {
      res.json(foto);
    }, function(error) {
      console.log(error);
      res.sendStatus(500);
    });
};

// código posterior omitido
```

Não há grande novidades aqui, a não ser a função `model.create` que recebe como parâmetro a foto que recebemos em `req.body`. Quando a função é executada com sucesso, recebemos como parâmetro o documento que foi criado, inclusive com seu ID já preenchido. Dependendo da sua aplicação, pode ser interessante ter acesso ao ID do novo documento. Este documento é enviado como resposta. Se algo der errado, enviamos o já conhecido código 500. Teste, teste, teste! Funciona, maravilha!

Atualização

Por fim, a atualização:

```
// alurapic/app/api/fotos.js
// código anterior omitido

api.atualiza = function(req, res) {

  model.findByIdAndUpdate(req.params.id, req.body)
    .then(function(foto) {
      res.json(foto);
    }, function(error) {
      console.log(error);
      res.sendStatus(500);
    })
};
```

```
};  
// código posterior omitido
```

Usamos a função `model.findByIdAndUpdate` que recebe como primeiro parâmetro o ID do documento que desejamos atualizar e como segundo os novos dados daquele documento.

Terminou?

Muito bem, recriei do zero o servidor do **alurapic**, aquele que foi disponibilizado para o curso de Angular. Não fiz uma mera cópia do código, tanto porque o servidor que lhe foi fornecido tinha a estrutura mínima para funcionar sem qualquer outro tipo de preocupação e muito menos se integrava com o MongoDB. Procurei passar para você não apenas cada tecnologia, mas como elas se integram. Está com saudades já?

Para prestigiar você, que chegou até o final do treinamento eu elaborei um bônus especial. Onde está? No próximo capítulo!