06

Express, estimulando nosso bebê

No capítulo anterior, conseguimos compartilhar a pasta `alurapic/public`, permitindo que o nosso navegador acessasse `index.html`, nossa view principal da nossa aplicação Angular. Porém, nada era exibido.

ATENÇÃO: se você não fez o treinamento de AngularJS do Alura, pré-requisito deste treinamento, seus olhos vão arder e seu aproveitamento deixará a desejar. Please, não pule etapas, faça o treinamento. Durante este treinamento assumo que você já viu e praticou todo o código do treinamento de Angular.

O problema é que a aplicação Angular está conversando com o servidor, aplicando um verbo que ele, nosso servidor, coitadinho, sendo um bebê, ainda não entende. Vejamos um exemplo do código que realiza essa comunicação.

```
// alurapic/public/js/controllers/fotos-controller.js

// código da aplicação Angular, restante omitido

    recursoFoto.query(function(fotos) {
        $scope.fotos = fotos;
    }, function(erro) {
        console.log(erro);
    });
```

Temos `recursoFoto` que chama a função `query`. Esta função, por debaixo dos panos, realiza uma requisição do tipo GET, ou melhor, realiza uma requisição para o servidor empregando o verbo GET. Mas para qual endereço do servidor? Para isso, precisamos escrutinar o código do serviço `recursoFoto`:

```
// alurapic/public/js/services/meus-servicos.js

    return $resource('/v1/fotos/:fotoId', null, {
            'update' : {
                method: 'PUT'
            }
        });
// código posterior omitido
```

Recordando o que você aprendeu do curso de Angular, o $resource recebe como parâmetro o endereço do recurso que queremos acessar no servidor. Como só estamos querendo ler dados sem enviar qualquer parâmetro, a parte da URL que será utilizada será `/v1/fotos`. Sendo assim temos:

VERBO: GET

URL: /v1/fotos

Sendo assim, nosso servidor teria que ser capaz de retornar uma lista de fotos para o endereço `http://localhost:3000/v1/fotos`, mas infelizmente nosso servidor é um analfabeto, ou seja, ainda é um bebê aprendendo a conversar com o mundo.

Questão

Qual das opções abaixo configura o módulo `alurapic/config/express.js`, criando a URL `http://localhost:3000/v1/fotos` e retornando uma lista de fotos. Não se esqueça que o verbo GET deve ser empregado.

Selecione uma alternativa

A

```
var express = require('express');
var app = express();

app.use(express.static('./public'));

app.get('/v1/fotos', function(req, res) {

    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'
    ];
    res.json(fotos);
});

module.exports = app;
```

B

```
var express = require('express');
var app = express();

app.use(express.static('./public'));

app.get('/v1/fotos', function() {

    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'
    ];
    res.end(fotos);
});

module.exports = app;
```

C

```
var express = require('express');
var app = express();

app.use(express.static('./public'));

app.get('/v1/fotos', function(req, res) {

    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'
    ];
    json(fotos);
});

module.exports = app;
```