14

Consolidando seu conhecimento

Preparado para dar continuidade no projeto do treinamento? Para fazê-lo, é importante que você tenha feito o último exercício do capítulo anterior ou tenha baixado o stage com o código completo do capítulo anterior no texto explicativo deste capítulo. Seguem as linhas gerais do que deve ser feito e, como sempre, sem fazer spoilers ou dar de mão beijada a questão:

1 - Uma conexão do mongoose não é suficiente, precisamos não apenas conectar ao banco, mas realizar operações de consulta, por exemplo. É por isso que você precisa criar um modelo. O padrão em nosso projeto é que todos ficarão em `alurapic/app/models`. Crie a pasta caso ainda não tenha criado. O modelo de foto será `alurapic/app/models/fotos.js`.

2 - É claro, você precisa do Mongoose para criar seu modelo, mas antes do modelo pensar em existir, é necessário criar um esquema (Schema) de uma foto. Recorde que no MongoDB esquemas são criados no lado da aplicação e servem de base para a criação de modelos. Crie um esquema que tenha `titulo`, `url` e `grupo`. Sendo que os dois primeiros são do tipo String e o último Number. Inclusive todos serão obrigatórios.

3 - Compile um modelo chamado `Foto` com base no esquema que foi criado.

4 - Na programação não há milagres, para que nosso sistema automaticamente carregue o módulo do nosso modelo, registrando-o no Mongoose, é necessário fazer o carregamento automático da pasta `alurapic/app/models`. Ainda lembra o papel do consign nesta história?

5 - O ponto alvo da nossa alteração é o arquivo `alurapic/app/api/fotos.js`. Lembre-se que são nossas API's que acessam o banco de dados e por isso dependem de um modelo do Mongoose. Confie em mim, apague todo o conteúdo do seu arquivo `alurapic/app/api.fotos.js` e cole o esqueleto apenas da API:

```
// alurapic/app/api/fotos.js

module.exports = function(app) {

    var api = {};

    api.lista = function(req, res) {

    };

    api.buscaPorId = function(req, res) {

    };

    api.removePorId = function(req, res) {

    };

    api.adiciona = function(req, res) {

    };

    api.atualiza = function(req, res) {

    };
```

```
    return api;
};
```

6 - Foque apenas `api.lista` . Use nosso modelo para buscar uma lista de fotos e disponibilizá-la como resposta no formato JSON.

7 - É claro, como só temos `api.lista` implementado, todo restante que havíamos feito deixou de existir e nossa aplicação Angular ficou comprometida. Não se preocupe, no próximo capítulo implementaremos o restante da nossa API.

Responda

INSERIR CÓDIGO		FORMATAÇÃO