

Desafio - Estágio

Descrição do desafio	1
Fluxo do sistema	1
Objetivo do desafio	2
Passos recomendados	2

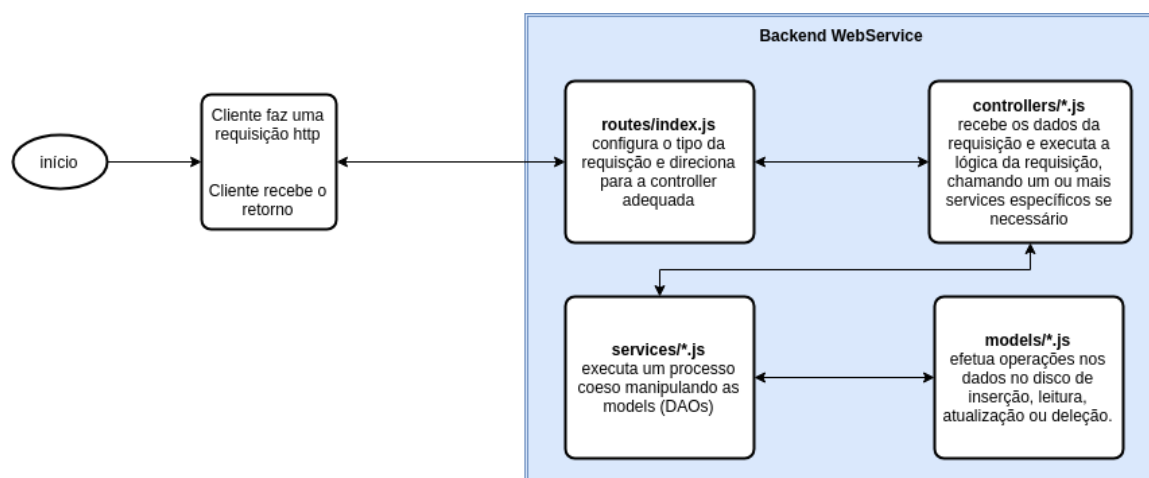
Descrição do desafio

O desafio consiste em trabalhar no backend *WebService* desenvolvido no framework **NodeJS**. O projeto anexado contém a estrutura básica do WebService em NodeJS com algumas funções e fluxos de exemplo. A estrutura do projeto está no seguinte formato:

- `config/` -> pasta com as configurações do servidor e um middleware http
- `app.js` -> arquivo inicial da aplicação com a configuração da porta que irá rodar
- `app/` -> diretório que compõe a lógica da aplicação.
- `app/controllers` -> controladores da aplicação que irá processar e gerenciar a lógica de cada rota.
- `app/models` -> classes que manipulam os dados em JSON (ou banco de dados): inserir, editar, excluir, ler.
- `app/routes` -> configuração das rotas do sistema. Mapeia a url, o tipo de requisição e para qual controller a requisição irá.
- `app/services` -> executa a lógica da aplicação de maneira bem desacoplada. Cada arquivo de *service* é responsável por manipular pequenas partes coesas do sistema.
- Além disso há dois arquivos jsons no projeto que serão utilizados no desafio:
 - `ar.json`
 - `luz.json`

Fluxo do sistema

Em síntese o fluxo de execução do WebService do desafio é como segue:



Objetivo do desafio

O objetivo é construir 3 requisições **HTTP GET** para retornar os dados dos arquivos .json:

1. Uma requisição para retornar o conteúdo do arquivo **ar.json**, com o endpoint:
 - a. /api/list/ar
2. Uma requisição para retornar o conteúdo do arquivo **luz.json**, com o endpoint:
 - a. /api/list/luz
3. Uma requisição para retornar o conteúdo de ambos os arquivos mesclados em uma única lista, com o endpoint:
 - a. /api/list/all

É necessário ter o NodeJS versão 8.x.x ou 10.x.x.

Para configurar o projeto basta executar o comando do npm dentro da pasta do projeto:

```
$ npm install
```

As dependências serão baixadas e instaladas. Também recomenda-se instalar o nodemon para facilitar o desenvolvimento, pois permite que o sistema se reinicie toda vez que um arquivo é salvo, de maneira automática:

```
$ npm install nodemon;
```

Para iniciar a aplicação basta executar o comando:

```
$ npm start (ou nodemon app.js; ou node app.js)
```

Há alguns endpoints pré-definidos com o fluxo completo que podem ser utilizados como base para o desenvolvimento dos demais.

Passos recomendados

- Instalar e configurar o NodeJS
- Configurar o ambiente com o npm
- Iniciar o projeto e testar requisitando localhost:8080
- Definir as rotas
- Criar a(s) controller(s)
- Criar o(s) service(s)
- Criar as models (DAOs)
- Implementar as lógicas necessárias

