

Sistemas Distribuídos e Mobile

Prof. Flávio Calado
Prof. Marcelo Amorim

Desafio Servidor CRUD Simples Valor (5 pontos)

Introdução

Uma tarefa comum para desenvolvedores é criar um CRUD (acrônimo em inglês Create, Read, Update Delete), que é em suma as quatro operações básicas aplicadas a um banco de dados.

De posse desta definição simples é necessário entender os conceitos que envolvem a criação do CRUD, por mais simples que se seja é um importante saber que é uma demanda que mais é solicitada em particular no início da carreira.

Nosso desafio é configurar inicialmente um servidor que realize a operação CRUD configurando uma API de forma rápida.

Cronograma de entrega

1. O trabalho é individual
2. O resultado do projeto deve ser mantido no repositório GitHub <https://github.com/> de cada estudante
3. O link para o repositório do projeto deve ser enviado através da pasta criada no Classroom até o dia 02 de abril.

Artefatos de entrega

- 1) Criar o servidor node utilizando o pacote Express
- 2) Criar um Array JSON no servidor para armazenar os dados de 5 clientes fictícios
Exemplos:
<https://www.delftstack.com/pt/howto/javascript/javascript-json-array-of-objects/>
- 3) Você deverá criar as seguintes APIs
 - a. GET /clientes – retornar todos os clientes
 - b. GET /cliente/{id} – retornar os dados de um único cliente
 - c. POST /cliente/{id}/{nome}/{endereço}/{email} – cadastrar um novo cliente
 - d. PATCH /cliente/{id}/{nome}/{endereço}/{email} – alterar os dados de um cliente
 - e. DELETE /cliente/{id} – deletar os dados de um cliente com base em seu id
- 4) Cada uma das APIs deve ser testada no vscode (ou postman) com o objetivo de entender os comandos do protocolo HTTP e a relação direta com um CRUD.

O modelo JSON

JSON (JavaScript Object Notation) é um modelo de armazenamento e transmissão de informação em formato texto. É um modelo simples, porém bastante utilizado em aplicações Web em função da sua capacidade de estruturar informações de maneira compacta do que a obtida pelo modelo XML, o que torna este modelo rápido quando das aplicações de parser. O que justifica a adoção do modelo por organizações como Google, Yahoo, dentre outras.

A sintaxe no JSON é simples, para cada valor representado, atribui-se um nome (rótulo) que descreve o seu significado, como na figura abaixo onde "id": "1" id é o rótulo e "1" é o valor do rótulo.

Desta forma temos a estrutura atômica do modelo, tem a sintaxe nome/valor que é representado pelo nome (significado) entre aspas duplas, seguido do valor, sendo que os valores podem possuir três tipos básicos: numérico (inteiro ou real), booleano e strings todos representados na figura abaixo.

```
{
  "cliente":
  [
    {
      "id": 1,
      "nome": "jose carmino",
      "endereco": "rua sem nome sem numero",
      "email": "jose.gomes@saojudas.br",
      "altura": 1.93,
      "casado": true
    }
  ]
}
```

A partir dos tipos primitivos, é possível construir tipos mais elaborados que são:

- **Arrays** os quais são delimitados por colchetes, e seus elementos são separados por vírgulas, como ilustrado na figura abaixo.

```
["RJ", "SP", "MG", "BA", "ES"]
```

- **Arrays de objetos:** os arrays podem armazenar objetos complexos

```
var jsonArray = {  
  "players": [  
    { "name": "Messi", "goals": 8 },  
    { "name": "Ronaldo", "goals": 22 },  
    { "name": "Costa", "goals": 20 },  
    { "name": "Neymar", "goals": 13 },  
    { "name": "Arabi", "goals": 6 },  
    { "name": "Bale", "goals": 3 },  
    { "name": "Toquero", "goals": 0 } ]  
};
```

- **A matriz é semelhante ao Array**, a diferença é que a matriz é composta por várias linhas delimitados por colchetes.

```
[  
  [1,5],  
  [3,33],  
  [120,99]  
]
```

Por fim os objetos podem ser compostos por múltiplos pares nome/valor, arrays e outros objetos, o que possibilita representar qualquer tipo de informação.

Atividade

A estrutura do JSON consiste em um objeto de cliente, formado por dois conjuntos de dados compostos pelos rótulos "id", "nome", "endereço" e "email", como apresentado abaixo.

```
{  
  "cliente": [  
    {  
      "id": 1,  
      "nome": "jose carmino",  
      "endereço": "rua sem nome sem numero",  
      "email": "jose.gomes@saojudas.br"  
    },  
    {  
      "id": 2,  
      "nome": "joao da silva",  
      "endereço": "rua sem nome sem numero",  
      "email": "joao.silva@saojudas.br"  
    }  
  ]  
}
```



LIVE

ecossistema
ânima

```
"endereco": "rua da silva sem numero",  
"email": "joao.silva@saojudas.br"  
}  
]  
}
```

Testando o servidor

Feito isso é possível abrir a URL <http://localhost:3000/cliente> em um navegador, este trará como resultado o conteúdo do arquivo JSON.