

Linguagens de Marcação e Scripts

Prof. Aníbal Cavalcante de Oliveira

UFC - QXD0164 - 2019.2

Agenda - Aula 21

Consumindo nossa API de cadastro de aniversários;

- Realizando chamada com os métodos:
 - POST (cadastrar um novo objeto)
 - PUT (atualizar um objeto já existente)
 - DELETE (excluir um objeto existente)

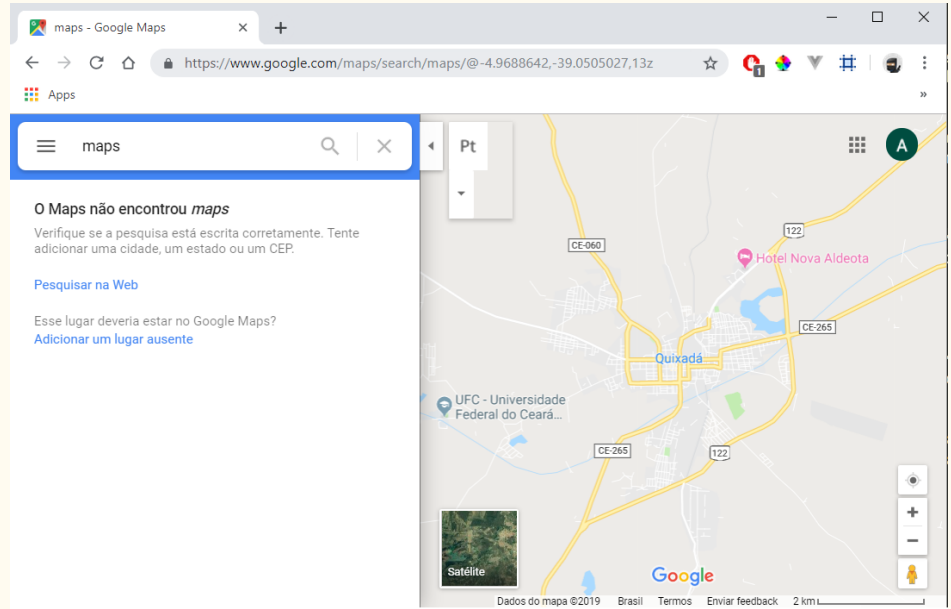
Conceito de API ("Application Programming Interface")

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web.

Através das APIs, os aplicativos podem se comunicar uns com os outros de forma transparente para os usuários.

O Google Maps é exemplo de uma API.

Por meio de seu código original, muitos outros sites e aplicações utilizam os dados do Google Maps adaptando-o da melhor forma a fim de utilizar esse serviço. Os desenvolvedores do site do hotel utilizam do código do Google Maps para inseri-lo em um determinado local de sua página



Mas o que significa API na prática?

As APIs proporcionam a integração entre sistemas que possuem linguagem totalmente distintas de maneira ágil e segura.

Exemplos de integração usando API's:

- 1 - Facebook / UBER;
- 2 - Google Maps / Sites de hotéis;
- 3 - Sites de E-Commerce / Operadoras de crédito (PayPal, PagSeguro);
- 4 - Logar /Cadastrar em sites com Redes Sociais (Google Account, Facebook);

A nossa API de Aniversários

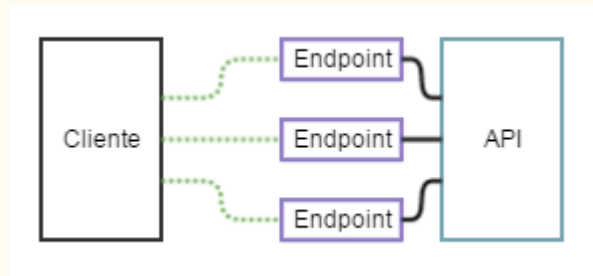
A API de aniversário de usuários está hospedada em: <https://lmsqxd.herokuapp.com/>

Os fontes estão em: <https://github.com/hanibalce/projetoSimplesNodeExpressMongo.git>

Ela foi construída com Node.js, Express e MongoDB.

Possui as seguintes **URLs** com os seguintes **endpoints**:

- 1 - GET /usuários
- 2 - GET /usuários/:id
- 3 - POST /usuários
- 4 - DELETE /usuários/:id



Um **endpoint** de uma API é a URL onde seu serviço pode ser acessado por uma aplicação **cliente**.

Requisição AJAX - GET - /usuários

Parte 1 - Lista todos os usuários cadastrados, portanto retorna um Array que pode ser percorrido. Abrir arquivo listar.js e inserir o código abaixo.

```
function carregarJSON() {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){  
            var resposta = JSON.parse(this.responseText);  
            var tabelaBody = document.getElementById("angel").lastElementChild;  
            tabelaBody.innerHTML = "";  
            for (var i = 0; i < resposta.length; i++) {  
                ...  
            }  
        }  
    }  
    xhttp.open("GET", "https://lmsqxd.herokuapp.com/usuarios");  
    xhttp.send(null);  
}
```

Requisição AJAX - GET - /usuários

Parte 2 - Dentro do laço for montar cada linha da tabela.

```
for (var i = 0; i < resposta.length; i++) {  
  
    var td1 = document.createElement("td");  
    var td2 = document.createElement("td");  
    var td3 = document.createElement("td");  
    var td4 = document.createElement("td");  
  
    td1.appendChild(document.createTextNode(resposta[i]._id));  
    td2.appendChild(document.createTextNode(resposta[i].nome));  
    td3.appendChild(document.createTextNode(resposta[i].email));  
    td4.appendChild(document.createTextNode(resposta[i].nascimento));  
  
    var tr = document.createElement("tr");  
    tr.appendChild(td1);  
    tr.appendChild(td2);  
    tr.appendChild(td3);  
    tr.appendChild(td4);  
    tabelaBody.appendChild(tr);  
}
```

Requisição AJAX - GET - /usuários/:id

Buscar um aniversariante pelo seu id. O id é chave que cada usuário ganha ao ser cadastrado. É um atributo único dado pelo MongoDB. Abrir o arquivo **buscar.js**;

```
function buscarPorId() {
    var id = document.getElementById("id").value;
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){
            var resposta = JSON.parse(this.responseText);
            var tabelaBody = document.getElementById("angel").lastElementChild;
            ...
        }
    }
    xhttp.open("GET", "https://lmsqxd.herokuapp.com/usuarios/" + id);
    xhttp.send(null);
}
```


Requisição AJAX - GET - /usuários/:id

Criar a tabela apenas com o elemento encontrado.

```
var td1 = document.createElement("td");
var td2 = document.createElement("td");
var td3 = document.createElement("td");
var td4 = document.createElement("td");

td1.appendChild(document.createTextNode(resposta._id));
td2.appendChild(document.createTextNode(resposta.nome));
td3.appendChild(document.createTextNode(resposta.email));
td4.appendChild(document.createTextNode(resposta.nascimento));

var tr = document.createElement("tr");
tr.appendChild(td1);
tr.appendChild(td2);
tr.appendChild(td3);
tr.appendChild(td4);
tabelaBody.appendChild(tr);
```

Requisição AJAX - DELETE - /usuários/:id

Para excluir o usuário usamos a mesma URL anterior porém com o método **DELETE** do protocolo HTTP. Abrir arquivo `excluir.js`

```
function excluir() {
    var id = document.getElementById("id").value;
    var elemento = document.getElementById("angel");
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4){
            if (this.status >= 200 && this.status < 300){
                elemento.innerHTML = "Objeto excluído com sucesso!";
            }else{
                elemento.innerHTML = "Erro ao excluir contato!";
            }
        }
    }
    xhttp.open("DELETE", "https://lmsqxd.herokuapp.com/usuarios/" + id);
    xhttp.send(null);
}
```

Requisição AJAX - POST - /usuários

Para salvar um novo usuário em nossa API, devemos realizar os seguintes passos:

1 - Criar um objeto do tipo Usuário;

```
var obj = {nome : "John", email:"john@gmail.com" , nascimento: "1945-01-01" };
```

2 - Utilizar a função `JSON.stringify()` para converter o objeto numa String;

```
var myJSON = JSON.stringify(obj);
```

3 - Usar método POST do protocolo HTTP e enviar o usuário no corpo da mensagem.

```
xhttp.open("POST", " https://lmsqxd.herokuapp.com/usuarios");  
xhttp.setRequestHeader("Content-type", "application/json");  
xhttp.send(myJSON);
```

Requisição AJAX - POST - /usuários

Passo 1 e 2 = Arquivo: cadastrar.js

```
class Usuario {
    constructor(nome, email, nascimento) {
        this.nome = nome;
        this.email = email;
        this.nascimento = nascimento;
    }
}

function salvar(){
    var _nome = document.getElementById("nome").value;
    var _email = document.getElementById("email").value;
    var _nascimento = document.getElementById("nascimento").value;

    var usuario = new Usuario(_nome, _email, _nascimento);
    var usuarioJSON = JSON.stringify(usuario);
    enviar(usuarioJSON);
}
```

Requisição AJAX - POST - /usuários

Passo 3

```
function enviar(usuarioJSON){
    var elemento = document.getElementById("angel");
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function(){
        if (this.readyState == 4){
            if (this.status >= 200 && this.status < 300){
                elemento.innerHTML = "Objeto cadastrado com sucesso!";
            }else{
                elemento.innerHTML = "Erro ao cadastrar usuário!";
            }
        }
    }

    xhttp.open("POST", "https://lmsqxd.herokuapp.com/usuarios");
    xhttp.setRequestHeader("Content-type", "application/json");
    xhttp.send(usuarioJSON);
}
```

Requisição AJAX - PUT - /usuários

Para atualizar um usuário existente em nossa API, devemos realizar os seguintes passos:

- 1 - Recuperar o objeto do Usuário e alterar seus valores;
- 2 - Utilizar a função `JSON.stringify()` para converter o objeto alterado numa String;
- 3 - Usar método PUT do protocolo HTTP e enviar o usuário alterado no corpo da mensagem.

```
xhttp.open("PUT", " https://lmsqxd.herokuapp.com/usuarios");  
xhttp.setRequestHeader("Content-type", "application/json");  
xhttp.send(myJSON);
```

Requisição AJAX - PUT - /usuários

Passo 1 = Arquivo: atualizar.js

```
function buscarPorId() {  
    var id = document.getElementById("id").value;  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){  
            var resposta = JSON.parse(this.responseText);  
            document.getElementById("nome").value = resposta.nome;  
            document.getElementById("email").value = resposta.email;  
            document.getElementById("nascimento").value = resposta.nascimento.substr(0, 10);  
        }  
    }  
    xhttp.open("GET", "https://lmsqxd.herokuapp.com/usuarios/" + id);  
    xhttp.send(null);  
}
```

Requisição AJAX - PUT - /usuários

Passo 2 = Arquivo: atualizar.js

```
class Usuario {
    constructor(id, nome, email, nascimento) {
        this._id = id;
        this.nome = nome;
        this.email = email;
        this.nascimento = nascimento;
    }
}

function atualizar(){
    var _id = document.getElementById("id").value;
    var _nome = document.getElementById("nome").value;
    var _email = document.getElementById("email").value;
    var _nascimento = document.getElementById("nascimento").value;

    var usuario = new Usuario(_id, _nome, _email, _nascimento);
    var usuarioJSON = JSON.stringify(usuario);
    enviar(usuarioJSON);
}
```


Requisição AJAX - POST - /usuários

Passo 3 = Arquivo: atualizar.js

```
function enviar(usuarioJSON){
    var elemento = document.getElementById("angel");
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function(){
        if (this.readyState == 4){
            if (this.status >= 200 && this.status < 300){
                elemento.innerHTML = "Objeto atualizado com sucesso!";
            }else{
                elemento.innerHTML = "Erro ao atualizar o usuário!";
            }
        }
    }

    xhttp.open("PUT", "https://lmsqxd.herokuapp.com/usuarios");
    xhttp.setRequestHeader("Content-type", "application/json");
    xhttp.send(usuarioJSON);
}
```

Trabalho para entregar no dia 26/11 (Individual)

Recupere um objeto JSON da API: <https://randomuser.me/api/>

Crie uma página WEB a partir dos dados obtidos pela API.

A página deve incluir:

- 1 - Nome da pessoa;
- 2 - Imagem do perfil;
- 3 - Gênero;
- 4 - Idade e Data de Nascimento;
- 5 - Os dois Telefones (fixo e móvel);
- 6 - Endereço Completo e coordenadas do Google Maps;

* Ganhará 1 ponto a mais para quem renderizar o endereço da pessoa na página usando a API do Google Maps.