

JavaScript: JSON e LocalStorage



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

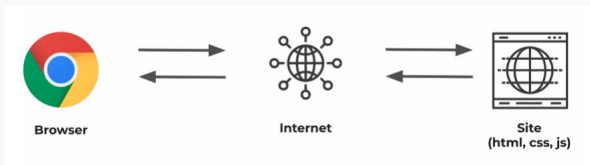
AGENDA

- JSON
- LocalStorage

Porque estudar?

A internet funciona por meio de uma arquitetura **cliente-servidor**.

Na prática, significa que quando eu (*cliente*) acesso um site ele buscará informações em uma base de dados (*servidor*)



Para que essa “comunicação” aconteça é preciso que as **informações sejam transmitidas** em um determinado formato, sendo o mais comum o JSON.

```
1 {  
2   "name": "Smith",  
3   "nickname": "smt",  
4   "id": 475,  
5   "age": 25,  
6   "height": 150,  
7   "lastOverScore": [6, 2, 1, 4, 1, 4]  
8 }
```

O que é?

JSON (*Javascript Object Notation*) é uma **formatação** utilizada para estruturar dados em formato de texto e transmiti-los de um sistema para outro

JSON: Sintaxe

JSON é uma maneira de se representar objetos, mas não exatamente como os objetos no JavaScript.

```
{
  "user": {
    "id": 12345,
    "name": "John Doe",
    "email": "john.doe@gmail.com",
    "active": true,
    "posts": [
      {
        "id": 12345,
        "title": "Hello World",
        "body": "This is a test post 1"
      },
      {
        "id": 12345,
        "title": "Hello World 2",
        "body": "This is a test post 2"
      }
    ]
  }
}
```

```
// Objeto JavaScript
const user = {
  id: "12345",
  name: "John Doe",
  email: "john.doe@gmail.com",
  active: true,
  posts: [
    {
      id: "12345",
      title: "Hello World",
      body: "This is a test post 1",
    },
    {
      id: "67890",
      title: "Hello World 2",
      body: "This is a test post 2",
    },
  ],
};
```

Regras

Regras de sintaxe:

- Os dados são em pares
nome:valor
- O nome é uma string, mas o valor pode ser qualquer tipo de dado
- Dados são separados por vírgula
- Chaves guardam objetos
- Colchetes guardam arrays
- Necessário usar aspas duplas

Podemos receber informações JSON por meio da URL de uma **API**.

Por exemplo, nas API disponibilizada pelo **IBGE** podemos ver os nomes mais comuns do CENSO:

<https://servicodados.ibge.gov.br/api/v2/censos/nomes/ranking>

```
[{"localidade": "BR", "sexo": null, "res":  
  [{"nome": "MARIA", "frequencia": 11734129, "ranking": 1},  
  {"nome": "JOSE", "frequencia": 5754529, "ranking": 2},  
  {"nome": "ANA", "frequencia": 3089858, "ranking": 3},  
  {"nome": "JOAO", "frequencia": 2984119, "ranking": 4},  
  {"nome": "ANTONIO", "frequencia": 2576348, "ranking": 5},  
  {"nome": "FRANCISCO", "frequencia": 1772197, "ranking": 6},  
  {"nome": "CARLOS", "frequencia": 1489191, "ranking": 7},  
  {"nome": "PAULO", "frequencia": 1423262, "ranking": 8},  
  {"nome": "PEDRO", "frequencia": 1219605, "ranking": 9},  
  {"nome": "LUCAS", "frequencia": 1127310, "ranking": 10},  
  {"nome": "LUIZ", "frequencia": 1107792, "ranking": 11},  
  {"nome": "MARCOS", "frequencia": 1106165, "ranking": 12},  
  {"nome": "LUIS", "frequencia": 935905, "ranking": 13},  
  {"nome": "GABRIEL", "frequencia": 932449, "ranking": 14},  
  {"nome": "RAFAEL", "frequencia": 821638, "ranking": 15},  
  {"nome": "FRANCISCA", "frequencia": 725642, "ranking": 16},  
  {"nome": "DANIEL", "frequencia": 711338, "ranking": 17},  
  {"nome": "MARCELO", "frequencia": 693215, "ranking": 18},  
  {"nome": "BRUNO", "frequencia": 668217, "ranking": 19},  
  {"nome": "EDUARDO", "frequencia": 632664, "ranking": 20}]]
```

JSON: Convertendo Dados

JSON => JS

Apesar de parecer com um objeto JS, ao receber dados JSON do servidor, o objeto será recebido no formato de *string*.

Por essa razão, para usá-lo no código usamos a função **JSON.parse()**

JS **JSON.parse()**
and
JSON.stringify()



JS => JSON

Já para enviar dados de volta do JS precisamos converter a informação em uma **string**.

Fazemos isso usando a função **JSON.stringify()**

Desafio

Baixar o programa Postman para Windows/Linux e fazer uma requisição GET para o seguinte endereço:

<https://jsonplaceholder.typicode.com/todos/1>

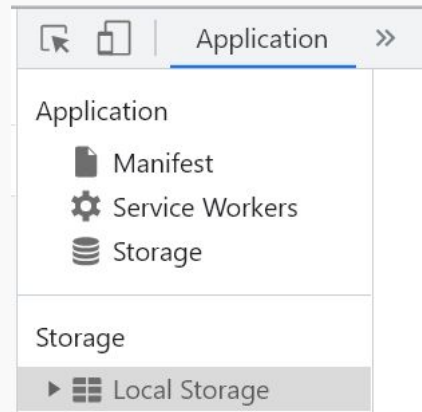
**Analise e entenda a resposta que é entregue em formato JSON.
Em seguida, explique, com suas palavras,
o que entendeu a respeito.**

Dica (em ingles): <https://medium.com/@YvonneFPan/using-postman-run-api-tests-1fe430bb31cf>

Porque estudar?

Muitas vezes precisamos **gravar informações** no próprio **browser**, sem enviar para o servidor, seja um valor, um código ou um texto.

Para isso foi criado o **localStorage**.



O que é?

O localStorage é uma propriedade que permite armazenar e recuperar dados **localmente** no próprio navegador do usuário.

LocalStorage: Exemplo

Quando acessamos um e-commerce (loja online) e queremos realizar uma compra, nós adicionamos produtos ao “carrinho”.

Enquanto não efetuamos a compra, os produtos **ficam “salvos” no browser**, mesmo saindo e voltando da página.

Exemplo: e-commerce exemplificativo no [GitHub](http://hpshop-rosana.surge.sh/): <http://hpshop-rosana.surge.sh/>

[illegible]

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



LocalStorage: Sintaxe

- Sintaxe para utilização do localStorage

- Armazenar:

```
localStorage.setItem('myCat', 'Tom');
```

- Ler:

```
cat = localStorage.getItem('myCat');
```

- Remover:

```
localStorage.removeItem('myCat');
```

- Remover todos:

```
localStorage.clear();
```

Exemplo: JSON + LOCALSTORAGE

```
//declara elementos em variaveis para utilizarmos a qualquer momento no futuro;
const $input = document.querySelector("#texto")
const $adicionar = document.querySelector(".enviar")

//cria uma variavel que comportará a lista de informações enviadas pelo usuário e também salvas no storage
const RepositorioEmVariavelEspelhoDoStorage = [];

//adiciona ação de clique ao botão
$adicionar.addEventListener("click", function () {
  if ($input.value) {
    RepositorioEmVariavelEspelhoDoStorage.push(input.value)
    localStorage.setItem("array", JSON.stringify(RepositorioEmVariavelEspelhoDoStorage))
  }
});

if (localStorage.getItem("array")) {
  RepositorioEmVariavelEspelhoDoStorage = JSON.parse(localStorage.getItem("array"));
}
```

Crie um site com os seguintes requisitos:

- um campo de texto (input) onde o usuário irá digitar uma mensagem
- três botões:
 - um botão “**adicionar**”, que no momento do clique insira a mensagem na tela
 - um botão “**salvar mensagem**”, que no momento do clique salve as informações no localStorage
 - opcional: se o texto estiver vazio (length), deve mostrar uma mensagem ao usuário solicitando a inserção de algum texto.
 - um botão “**mostrar mensagem**”, que após o recarregamento da página, no momento do clique busque a mensagem no localStorage e exiba na tela
 - opcional: se não houver mensagem cadastrada no localStorage, deve mostrará na tela: “não há itens salvos”

- [JSON \(JavaScript Object Notation - Notação de Objetos JavaScript\)](#)
- [Trabalhando com JSON - Aprendendo desenvolvimento web | MDN](#)
- <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>
- [Orientação a Objetos x Programação Funcional - Fórum DevMedia.](#)

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>