

JSON

Lembrete da sintaxe de um objeto json:

```
let objetoEmFormatoJSON = '{ "atributo": "valor", "atributo1": 1, "atributo2": [], "atributo3": null, "atributo4": false }';
```

Em seguida, contamos com as funções `JSON.parse()` e `JSON.stringify()`

1. Gerar um objeto JSON que contenha apenas uma propriedade que seja uma string.

Em seguida, transformá-lo em string.

2. Gerar um objeto JSON amigos que contenha um número com a quantidade de amigos e um array com as strings dos nomes deles. Em seguida, transformá-lo em string.

3. Gerar um objeto JSON contendo um array com 3 objetos. Cada objeto deve ter um número e uma string. Em seguida, transformá-lo em string.

a. Quando tiver o objeto JSON, experimentar acessar cada uma das variáveis.

AJAX

Importante: Para os exercícios sobre AJAX, utilizaremos as URLs indicadas no final desta folha de exercícios.

A seguir, temos um exemplo

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
  if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
    console.log(xmlhttp.responseText); console.log(JSON.parse(xmlhttp.responseText));
    //Meu código
  }
};
xmlhttp.open("GET", "http://pilote.techo.org/?do=api.getPaises", true);
xmlhttp.send();
```

Importante 2: Ao fazer um request via POST, adicionar a linha

```
xmlhttp.setRequestHeader("Contenttype", "application/x-www-form-urlencoded");  
depois de invocar open()
```

Depois de uma série de exercícios, faremos um formulário de registro complexo para a organização

Recomendamos fazer esses exercícios em ordem.

4. Gerar um arquivo HTML com um título de registro e um formulário que só tenha um <select> de países. Esse <select>, a princípio, não terá nenhuma opção.
5. Relacionar o HTML a um arquivo JS que, quando o site for carregado, faça um pedido usando o método `getPaíses`. Imprimir em linha de comando o resultado desse pedido em texto e em um JSON.
7. Carregar os <select> de estado e sede utilizando os métodos `getUnidadesOrganizacionais` e `getRegioes`.
8. Modificar o código de forma que, quando o estado for modificado, o <select> de cidades seja atualizado utilizando `getCidades`.
9. Adicionar um botão de “finalizar” que utilize o método `registrarPessoa` mas não é necessário enviar dados a ele. Testar se o pedido é retornado quando é enviado.
10. Mostrar os erros que podem ser retornados por `registrarPessoa` mediante um alert.
11. Modificar o pedido de `registrarPessoa` para que envie os dados do formulário
12. Caso `registrarPessoa` seja bem-sucedido, ocultar o formulário e mostrar uma mensagem de sucesso ao usuário.

Métodos para utilizar via AJAX

Método `getPaíses`

- URL: <http://pilote.techo.org/?do=api.getPaíses>
- Método: GET
- Descrição: Retorna os países disponíveis no banco de dados.
- Parâmetros: Nenhum
- Valores de retorno:
 - “erro”: booleano que indica se houve um error.false, ou seja, que não houve erros
 - “conteúdo”: objeto. Cada atributo é o nome de um país, enquanto que o valor é o id que o representa.

Método getUnidadesOrganizacionais

- URL: [http://pilote.techo.org/?do=api.getUnidadesOrganizacionais?idPais=\[idPais\]](http://pilote.techo.org/?do=api.getUnidadesOrganizacionais?idPais=[idPais])
- Método: GET
- Descrição: Retorna as sedes disponíveis no banco de dados para um país determinado
- Parâmetros: 1
 - idPais: numero. id do país.
- Valores de retorno:
 - “erro”: booleano que indica se houve um error.false, ou seja, que não houve erros
 - “conteúdo”: objeto. Cada atributo é o nome de uma sede, enquanto que o valor é o id que o representa.

Método getRegioes

- URL: [http://pilote.techo.org/?do=api.getRegioes?idPais=\[idPais\]](http://pilote.techo.org/?do=api.getRegioes?idPais=[idPais])
- Método: GET
- Descrição: Retorna os estados/as regiões disponíveis no banco de dados para um país determinado
- Parâmetros: 1
 - idPais: numero. id do país.
- Valores de retorno:
 - “erro”: booleano que indica se houve um error.false, ou seja, que não houve erros
 - “conteúdo”: objeto. Cada atributo é o nome de um estado/região, enquanto que o valor é o id que o representa.

Método getCidades

- URL: [http://pilote.techo.org/?do=api.getCidades?idRegiaoLT=\[idRegiaoLT\]](http://pilote.techo.org/?do=api.getCidades?idRegiaoLT=[idRegiaoLT])
- Método: GET
- Descrição: Retorna as cidades disponíveis no banco de dados para um estado/uma região determinada
- Parâmetros: 1
 - id RegiaoLT: numero. id do estado/da região.
- Valores de retorno:
 - “erro”: booleano que indica se houve um error.false, ou seja, que não houve erros
 - “conteúdo”: objeto. Cada atributo é o nome de uma cidade, enquanto que o valor é o id que o representa.

Método registrarPessoa

● URL: <http://pilote.techo.org/?do=api.registrarPessoa>

● Método: POST

● Descrição: Registra uma pessoa

● Parâmetros: 1

○ escritório: número. id da sede.

○ nomes: string. nomes da pessoa.

○ sobrenome: string. sobrenome da pessoa.

○ e-mail: string. e-mail da pessoa.

○ password: string. senha da pessoa.

○ cpassword: string. confirmação da senha da pessoa.

○ rg: string/número. RG da pessoa.

○ dataNascimento: string. Data de nascimento da pessoa em formato AAAAMMDD. O usuário deve ter mais de 18 anos.

○ sexo: string. Sexo da pessoa. A resposta esperada é "M" ou "F".

○ celular: string/número. Celular da pessoa.

○ nacionalidade: número. Nacionalidade da pessoa. A resposta esperada é o id do país.

○ estado: número. Estado/da região da pessoa. A resposta esperada é o id do estado/da região.

○ cidade: número. Cidade da pessoa. A resposta esperada é o id da cidade.

● Valores de retorno:

○ "erro": booleano que indica se houve um erro.false, ou seja, que não houve erros

○ "conteúdo": array. Se o erro retornar true, o conteúdo terá um array de strings com os erros encontrados.