

Mãos na massa: adicionando os pacientes na tabela

Já conseguimos realizar a requisição HTTP, ir ao servidor e trazer os dados dos pacientes, que até mesmo imprimimos no console e vimos que a resposta é um grande texto formatado. Agora neste exercício vamos transformar este texto em um objeto Javascript e adicionar cada um destes pacientes na tabela.

1- Se você observar o que é impresso no console, você vai ver que o servidor nos retorna um JSON, um formato de texto muito comum hoje em dia na web. Como não queremos trabalhar com texto e sim com objetos Javascript, vamos parsear este texto para um objeto Javascript equivalente. Use a função `JSON.parse()` na resposta para transformar:

```
var botaoAdicionar = document.querySelector("#buscar-pacientes");

botaoAdicionar.addEventListener("click", function() {
  var xhr = new XMLHttpRequest();

  xhr.open("GET", "https://api-pacientes.herokuapp.com/pacientes");

  xhr.addEventListener("load", function() {
    var resposta = xhr.responseText;
    var pacientes = JSON.parse(resposta);

  });
  xhr.send();
});
```

2- Nossa variável `pacientes` agora é um array de objetos `paciente`, com nome, peso, altura, gordura e IMC. Agora precisamos iterar por este array e adicionar cada um destes pacientes na tabela, mas antes de começar a implementar isto na mão, vamos refatorar um pedaço de código que já havíamos feito para facilitar nossa vida.

3- Vamos refatorar um pedaço do código do `form.js` para aproveitar o código que adiciona uma paciente na tabela, exportando este código para uma função externa que aproveitaremos no nosso `buscar-pacientes.js`, afinal também queremos adicionar um paciente lá.

Em seu `form.js` crie a função `adicionaPacienteNaTabela`, que recebe um paciente. Ela vai fazer o que já fazíamos depois de montar um paciente do form. Só que agora, portado para uma função externa.

```
// Crie esta função em seu form.js
function adicionaPacienteNaTabela(paciente) {
  var pacienteTr = montaTr(paciente);
  var tabela = document.querySelector("#tabela-pacientes");
  tabela.appendChild(pacienteTr);
}
```

4- Agora no evento de click do `form.js`, faça as substituições e chame a função `adicionaPacienteNaTabela` aqui também. Seu código final do evento deve ficar assim:

```
botaoAdicionar.addEventListener("click", function(event) {
  event.preventDefault();
```

```
var form = document.querySelector("#form-adiciona");
var paciente = obtemPacienteDoFormulario(form);
var erros = validaPaciente(paciente);
if (erros.length > 0) {
    exibeMensagensDeErro(erros);
    return;
}

adicionaPacienteNaTabela(paciente);

form.reset();
var mensagensErro = document.querySelector("#mensagens-erro");
mensagensErro.innerHTML = "";
});
```

Cuidado ao fazer esta refatoração! Se for necessário copie e cole este código para que não haja erros!

5- Voltando para o `buscar-pacientes.js`, agora que temos uma função especialista em adicionar pacientes na tabela, vamos aproveitá-la para percorrer o array de pacientes e adicionar cada um deles:

```
var botaoAdicionar = document.querySelector("#buscar-pacientes");

botaoAdicionar.addEventListener("click", function() {
    var xhr = new XMLHttpRequest();

    xhr.open("GET", "https://api-pacientes.herokuapp.com/pacientes");

    xhr.addEventListener("load", function() {
        var resposta = xhr.responseText;
        var pacientes = JSON.parse(resposta);

        pacientes.forEach(function(paciente) {
            adicionaPacienteNaTabela(paciente);
        });
    });

    xhr.send();
});
```

Clique no botão de buscar pacientes e veja que agora os pacientes são carregados!