08

Mãos na massa: conectando com servidor

Começando deste ponto ?

Começando deste ponto? Você pode fazer o download (https://s3.amazonaws.com/caelum-online-public/introducao-javascript/stages/introducao-javascript-capitulo-9.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Neste capítulo nós implementaremos o botão `buscar` , que como vimos no vídeo, deve ir em um servidor externo através de uma requisição `GET` e trazer os dados de novos pacientes.

1- Como de praxe , já que se trata de uma nova funcionalidade vamos começar criando um novo arquivo na pasta `/js` , chamado de `buscar-pacientes.js` :

```
introducao-javascript
   └── js
       ├── calcula-imc.js
       ├── form.js
       ├── remover-paciente.js
       ├── filtra.js
       └── buscar-pacientes.js
```

2- Vamos importá-lo ao fim da tag `<body>` :

```
    </section>

    <script src="js/calcula-imc.js" ></script>
    <script src="js/form.js" ></script>
    <script src="js/remover-paciente.js" ></script>
    <script src="js/filtra.js" ></script>
    <script src="js/buscar-pacientes.js" ></script>
</body>
```

3- Agora vamos criar o botão `Buscar Pacientes` no HTML. Abaixo da `<table>` e acima do fechamento da `<section>` adicione o botão abaixo:

```
        </table>
        // Adicionar aqui!
        <button id="buscar-pacientes" class="botao bto-principal">Buscar Pacientes</button>

    </section>
</main>
```

4- Agora podemos no `buscar-pacientes.js` selecionar o botão e colocar um eventListener de click nele:

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

```
botaoAdicionar.addEventListener("click", function() {

});
```

5- Para executarmos requisições com o Javascript, precisamos de um objeto especialista nisso, que é o `XMLHttpRequest`. Vamos criar um novo objeto deste tipo:

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

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

6- O `XMLHttpRequest` precisa ser configurado, para dizer qual método HTTP queremos utilizar na requisição, e para qual servidor vamos enviá-la. Para configurar o `XMLHttpRequest` utilizamos a função `.open()`:

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

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

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

});
```

7- Por último , para enviar a requisição precisamos chamar o método `.send()`:

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

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

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

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

Se você clicar no botão neste momento, a requisição será enviada, porém não veremos nenhuma alteração na tela ou resultado no console, pois não estamos pegando a resposta. Vamos tratar disto.

8- Para pegarmos a resposta quando a requisiçao HTTP voltar precisamos colocar um escutador de evento no próprio `XMLHttpRequest`, escutando o evento de `load`:

```
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() {
        console.log(xhr.responseText);
    });

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

Clique no botão agora e observe o seu console, você verá que agora conseguimos exibir os resultados!