

Mão na massa: lidando com erros da requisição

Nossa requisição AJAX funciona, porém não estamos levando em conta a possibilidade de acontecer um erro na requisição. Neste exercício vamos detectar este problema e exibir o erro para o usuário caso aconteça algo:

1- Para detectarmos se ocorreu algo, devemos utilizar o código de status da requisição HTTP, que pode ser obtido através da propriedade `.status` do `XMLHttpRequest`. Vamos testar se o código é 200, que significa que a requisição foi OK, caso contrário, vamos exibir os erros para os usuários:

```
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() {

        if (xhr.status == 200) {
            var resposta = xhr.responseText;
            var pacientes = JSON.parse(resposta);

            pacientes.forEach(function(paciente) {
                adicionaPacienteNaTabela(paciente);
            });
        } else {
            //Exibiremos o erro aqui!
        }
    });

    xhr.send();
});

xhr.send();
```

2- Caso a requisição mostre qualquer outro código que não o 200, devemos exibir uma pequena mensagem de erro para o usuário. Esta mensagem erro será uma tag `` que começará com a classe invisível, e que caso dê algum problema ela ficará visível. Crie a tag `` que conterá a mensagem de erro acima da tag button:

```
</table>

<span id="erro-ajax" class="invisivel">Erro ao buscar os pacientes</span>

    <button id="buscar-pacientes" class="botao bto-principal">Buscar Pacientes</button>
</section>
</main>
```

3- O primeiro passo é selecionar o `` dentro do nosso código:

```
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() {
    //Adição aqui
    var erroAjax = document.querySelector("#erro-ajax");

    if (xhr.status == 200) {
      erroAjax.classList.add("invisivel");
      var resposta = xhr.responseText;
      var pacientes = JSON.parse(resposta);

      pacientes.forEach(function(paciente) {
        adicionaPacienteNaTabela(paciente);
      });
    } else {
      erroAjax.classList.remove("invisivel");
    }
  });

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

4- Se a requisição retornar qualquer erro, devemos remover a classe `invisivel` do ``, tornando a mensagem de erro visível para o usuário. Remova a classe `invisivel` caso o código não seja 200, e volte a adicionar caso a requisição não dê nenhum erro:

```
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 erroAjax = document.querySelector("#erro-ajax");

    if (xhr.status == 200) {
      erroAjax.classList.add("invisivel");
      var resposta = xhr.responseText;
      var pacientes = JSON.parse(resposta);

      pacientes.forEach(function(paciente) {
        adicionaPacienteNaTabela(paciente);
      });
    } else {
      erroAjax.classList.remove("invisivel");
    }
  });

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

Simule um erro, por exemplo removendo um pedaço da URL do servidor, e verifique se a mensagem de erro da tag `` irá aparecer.