

Ajax na App

Transcrição

O nosso app já faz muitas coisas, mas tem um ponto que deixamos em aberto: no momento em que fazemos um pedido e anotamos o número da mesa, ao clicarmos no botão "Pedir", queremos que o resumo seja enviado para a cozinha. Porém, precisamos pensar na forma de efetuar esta comunicação.

Para isto, usaremos outro serviço - que disponibilizei online, no endereço cozinhapsergioloopes.org (<http://cozinhapsergioloopes.org/>). É o *back-end* da cozinha. Trata-se de um painel simples, que ficará rodando em um monitor grande na cozinha e mostrará os pedidos solicitados, conforme as garçonetes forem confirmando.

Vamos fazer um teste e ver como funciona o painel: [na página de um pedido \(http://cozinhapsergioloopes.org/novo-pedido?mesa=17&pedido=Bolo+de+cenoura\)](http://cozinhapsergioloopes.org/novo-pedido?mesa=17&pedido=Bolo+de+cenoura) que foi feito. Ele irá exibir o número da mesa ("17") e o produto selecionado ("Bolo de Cenoura"). Faremos outro teste, agora a mesa 42, pediu um Bolo com Nutella. Com o [novo pedido \(http://cozinhapsergioloopes.org/novo-pedido?mesa=42&pedido=Bolo+com+Nutella\)](http://cozinhapsergioloopes.org/novo-pedido?mesa=42&pedido=Bolo+com+Nutella), iremos ver outro produto selecionado.

Deixei pronto todo este *back-end*, porque não é o nosso foco. Agora precisamos que o app chame o serviço *Ajax*, ao clicarmos no botão de confirmação do pedido.

No HTML do nosso código, já temos o botão de "Pedir". Quando alguém clicá-lo, precisamos que ele acione o código do Ajax. Para facilitar, iremos incluir a classe `acao-finalizar` no botão.

```
<div class="modal-footer">
  <button class="btn deep-orange waves-effect waves-light modal-close acao-finalizar">
    Pedir
  </button>
</div>
```

No arquivo `app.js` do projeto, vamos pegar o clique no `acao-finalizar`.

```
$('.acao-finalizar').click(function(){
});
```

Quando finalizarmos o pedido, ele fará o Ajax. Para isto, vamos o *jQuery* e ele irá receber alguns parâmetros, o primeiro será a `url` do serviço que apresentei anteriormente.

```
$('.acao-finalizar').click(function(){
  $.ajax({
    url: 'http://cozinhapsergioloopes.org/novo-pedido'

  })
});
```

Precisaremos indicar dois parâmetros: "Mesa" e o "Pedido". Nós não queremos que ele seja fixo, como "mesa: 17" e "pedido: Bolo de Cenoura". Abaixo da `url`, nós iremos adicionar o valor dos *ids* `numero-mesa` e `resumo`, que nos

informava o número da mesa e o resumo do pedido.

```
data: {
  mesa: $('#numero-mesa').val(),
  pedido: $('#resumo').text
}
```

Os `id`s estavam no arquivo HTML:

```
<div class="modal-content">
  <div class="modal-content">
    <h5>Resumo do pedido</h5>
    <input type="number" class="validate" id="numero-mesa">
    <label for="numero-mesa">Número da Mesa</label>
    <blockquote id="resumo"></blockquote>
  </div>
</div>
```

De volta ao arquivo `js`, precisaremos incluir uma mensagem de aviso para usuário do app confirmando a realização do pedido. Faremos isto com o `callback success` ("sucesso"). Ele irá receber a resposta do servidor e podemos apresentar a resposta do *back-end* com um `alert` ou o `Materialize.toast`. Iremos incluir aqui também o tempo de 2 segundos (`2000`).

```
success: function(resposta) {
  Materialize.toast(resposta, 2000);
}
```

Após o pedido ser feito com sucesso, vamos zerar o app para o próximo pedido. Para isto, usaremos o `numero-mesa` com o `val` em branco e removemos todos o número que aparecem no pedido com apagando os `badge`. Vamos aproveitar e incluir o `callback de error`, porque às vezes, as coisas não funcionam. Também iremos criar um aviso de erro com o `Materialize.toast`, definindo o tempo (`3000`) e a cor do texto (`red-text`).

```
success: function(resposta) {
  Materialize.toast(resposta, 2000);

  $('#numero-mesa').val('');
  $('.badge').remove();
},
error: function(erro) {
  Materialize.toast(erro.responseText, 3000, `red-text`);
}
```

Agora saberemos se tivemos sucesso na ação ou não.

Nós usamos um código de jQuery, não iremos nos aprofundar no tema, mas temos um curso específico sobre o assunto no Alura.

Nosso objetivo é observar que estamos chamando uma `url` no *back-end* e que passamos dois parâmetros (`numero-mesa` e `resumo`).

Vamos no navegador, pedimos para atualizar. Temos o *back-end* da cozinha aberto, ao fazermos um novo pedido, anotamos um outro número de mesa ("12") com os produtos selecionados ("Só de Cenoura" e "Com Nutella") e clicamos no botão "Pedir". O pedido da Mesa 12 chegará no Mural dos cozinheiros, enquanto no app, o pedido antigo será apagado.