08

Atualizando Totais do Carrinho na View com JavaScript

Transcrição

Começaremos a modificar o valor e quantidade total de itens no carrinho na tela via JavaScript, mas primeiro, veremos uma situação em que removemos a quantidade de itens do carrinho até que ela fique zerada.

Quando zeramos a quantidade de itens no carrinho devemos remover esse item do banco de dados, portanto voltaremos ao `ItemPedidoRepository.cs`, onde criaremos um método novo para realizar essa remoção quando necessária.

Na interface adicionaremos o método `RemoveItemPedido()` que retornará um `void`. Esse método terá como parâmetro `int itemPedidoId`

```
namespace CasaDoCodigo.Repositories
{
    public interface IItemPedidoRepository
    {
        ItemPedido GetItemPedido(int itemPedidoId);
                void RemoveItemPedido(int itemPedidoId);
    }
```

Assim feito, implementaremos o método na classe concreta: selecionamos `IItemPedidoRepository` e utilizando o atalho "Ctrl + ." escolheremos a opção "Implementar interface".

```
public class ItemPedidoRepository : BaseRepository<ItemPedido>, IItemPedidoRepository
    {
        public ItemPedidoRepository(ApplicationContext contexto) : base(contexto)


        {
        }
```

Desas forma, ao final do código teremos:

```
public void RemoveItemPedido(int itemPedidoId)
{
    throw new NotImplementedException();
}
```

Para removermos o item precisaremos obte-lo no banco de dados, faremos isso por meio do método `GetItemPedido()`, que receberá como parâmetro `itemPedidoId`. Iremos remover esse `itemPedidoId` do `dbSet`, que é o conjunto de itens do banco de dados. O `dbSet` possui um método para excluir itens do banco, chamado `Remove()`, que por sua vez receberá `GetItemPedido(itemPedidoId)`.

```
public void RemoveItemPedido(int itemPedidoId)
{
    dbSet.Remove(GetItemPedido(itemPedidoId));
}
```

Iremos até `PedidoRepository.cs` e chamaremos o método `RemoveItemPedido()` para excluir `ItemPedido` quando necessário. Adicionaremos a condicional `if`, e quando o `Quantidade` foi igual a `0`, chamaremos o `itemPedidoRepository.RemoveItemPedido(itemPedido.id)`.

```
public UpdateQuantidadeResponse UpdateQuantidade(ItemPedido itemPedido)
        {
            var itemPedidoDB = itemPedidoRepository.GetItemPedido(itemPedido.Id);

            if (itemPedidoDB != null)
            {
                itemPedidoDB.AtualizaQuantidade(itemPedido.Quantidade);

                if (itemPedido.Quantidade == 0)
                {
                    itemPedidoRepository.RemoveItemPedido(itemPedido.Id);
                }
```

No código JavaScript, atualizaremos o valor total do carrinho. Procuraremos o html que contém o elemento com a quantidade total de itens em nosso carrinho. Ao inspecionarmos o código html veremos que essa quantidade é armazenada em um `<span>`, precisamente no atributo `numero-itens`.

```
<span numero-itens> Total: 1 itens</span>
```

Localizaremos esse tributo por meio de uma filtragem do jQuery. No código JavaScript inseriremos o seletor para localizado atributo `numero-itens`. Quando esse elemento for encontrado, trocaremos o html com a função `html()`, que receberá `Total: 12345 itens`

```
postQuantidade(data) {
        $.ajax({
            url: '/pedido/updatequantidade',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify(data)
        }).done(function (response) {
            let itemPedido = response.itemPedido;
            let linhaDoItem = $('[item-id=' + itemPedido.id + ']')
            linhaDoItem.find('input').val(itemPedido.quantidade);
            linhaDoItem.find('[subtotal]').html((itemPedido.subtotal).duasCasas());

                    $('[numero-itens]').html('Total: 12345 itens');
                    debugger;
            });
```

Iremos intercalar a string `Total: 12345 itens` de forma que possamos substituir os caracteres numéricos pela quantidade total de itens no em `carrinhoViewModel`. Acessaremos a lista de `itens` e em seguida obteremos a quantidade de elementos da lista, que conseguimos acessar por meio da função JavaScript `lenght`.

```
postQuantidade(data) {
        $.ajax({
```

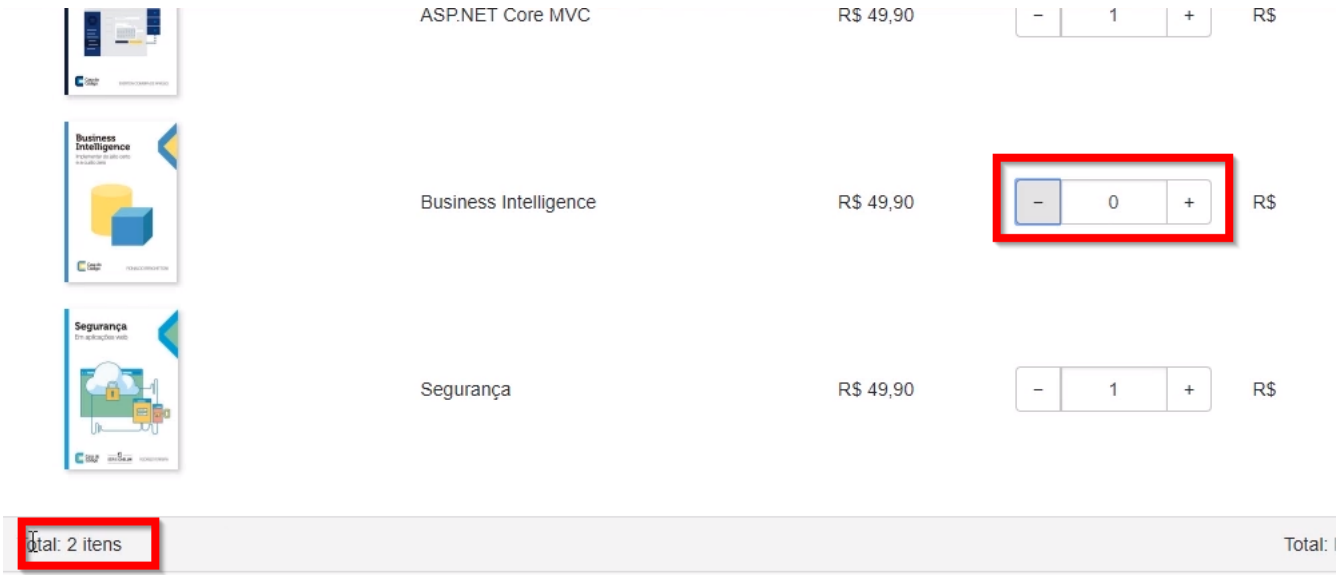
```
            url: '/pedido/updatequantidade',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify(data)
        }).done(function (response) {
            let itemPedido = response.itemPedido;
            let linhaDoItem = $('[item-id=' + itemPedido.id + ']')
            linhaDoItem.find('input').val(itemPedido.quantidade);
            linhaDoItem.find('[subtotal]').html((itemPedido.subtotal).duasCasas());

                        $('[numero-itens]').html('Total:' + carrinhoViewModel.itens.length + 'itens'
                        debugger;
                });
```

Contudo, `carrinhoViewModel` ainda não foi declarado, faremos isso na linha acima.

```
postQuantidade(data) {
        $.ajax({
            url: '/pedido/updatequantidade',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify(data)
        }).done(function (response) {
            let itemPedido = response.itemPedido;
            let linhaDoItem = $('[item-id=' + itemPedido.id + ']')
            linhaDoItem.find('input').val(itemPedido.quantidade);
            linhaDoItem.find('[subtotal]').html((itemPedido.subtotal).duasCasas());
                        let carrinhoViewModel = response.carrinhoViewModel;
                        $('[numero-itens]').html('Total:' + carrinhoViewModel.itens.length + 'itens'
                        debugger;
                });
```

Pressionaremos o atalho "F5" para compilar a aplicação. Depois, iremos até a página inicial e colocaremos três itens diferentes no carrinho de compras. Reduziremos a quantidade de um dos itens até que na caixa de texto seja exibido o valor `0`. A quantidade total de itens foi atualizada, contudo o item que apresenta `0` de quantidade não foi removido.



Isso aconteceu porque não configuramos para que o item seja removido da tela. Faremos isso nesta etapa. De volta ao código JavaScript, iremos verificar a quantidade de itens que foram alterados. Incluiremos a condicional `if` para fazer a verificação de `itemPedido`, caso ela seja igual a `0`, `linhaDoItem` será removida por meio do método `remove()`.

```
postQuantidade(data) {
        $.ajax({
            url: '/pedido/updatequantidade',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify(data)
        }).done(function (response) {
            let itemPedido = response.itemPedido;
            let linhaDoItem = $('[item-id=' + itemPedido.id + ']')
            linhaDoItem.find('input').val(itemPedido.quantidade);
            linhaDoItem.find('[subtotal]').html((itemPedido.subtotal).duasCasas());
            let carrinhoViewModel = response.carrinhoViewModel;
            $('[numero-itens]').html('Total: ' + carrinhoViewModel.itens.length + ' itens');


            if (itemPedido.quantidade == 0) {
                linhaDoItem.remove();
            }

            debugger;
        });
    }
}
```

Dessa forma, quando um item estiver com uma quantidade igual a `0`, ele será removido da tela e da quantidade total de itens. Contudo, falta atualizarmos o valor total do pedido. Para isso, inseriremos `$('[total]').html((carrinhoViewModel.total)`, e para que a formatação seja correta, inseriremos a função `duasCasas()` que envolverá toda a expressão.

```
postQuantidade(data) {
        $.ajax({
            url: '/pedido/updatequantidade',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify(data)
        }).done(function (response) {
            let itemPedido = response.itemPedido;
            let linhaDoItem = $('[item-id=' + itemPedido.id + ']')
            linhaDoItem.find('input').val(itemPedido.quantidade);
            linhaDoItem.find('[subtotal]').html((itemPedido.subtotal).duasCasas());
            let carrinhoViewModel = response.carrinhoViewModel;
            $('[numero-itens]').html('Total: ' + carrinhoViewModel.itens.length + ' itens');
            $('[total]').html((carrinhoViewModel.total).duasCasas());

            if (itemPedido.quantidade == 0) {
                linhaDoItem.remove();
            }

            debugger;
        });
    }
```

```
}
```

Com isso, o valor total e quantidade de itens são atualizados.