04

Usando Filtros jQuery Para Acessar Elementos

Transcrição

Realizamos o primeiro teste de interação entre o usuário e botão "+", que por sua vez chama função JavaScript por meio do evento `onclick` , entenderemos o porquê devemos sempre inserir o script dentro de uma view do Asp.NET Core dentro de uma sessão de `Script` .

```
@{
    ViewData["Title"] = "Carrinho";
}


@section Scripts
{

    <script type="text/javascript">
        function clickIncremento() {
            debugger;
        }
    </script>
}
```

Voltemos até o browser, na página de carrinho que exibe todos os itens que compramos. Iremos analisar o código html gerado da página, para isso, basta clicar com o botão direito do mouse e selecionar a opção "Exibir código fonte da página" ou pressionar o atalho "Ctrl + U". Ao final do código, veremos a função `clickIncremento()` inserida. Temos os botões `Adicionar Produtos` e `Finalizar Pedido` ,em seguida temos o rodapé da nossa página que foi gerado pelo `layout.cshtml` , a view que envolve todo o código html. Dois, temos os scripts utilizados na página, e ao final teremos o script específico para nossa página.

```
<!DOCTYPE html>

<****!****>


<div class="row">
    <div class="col-md-12">
        <div class="pull-right">
            <a class="btn btn-success" asp-action="carrossel">
                Adicionar Produtos
            </a>
            <a class="btn btn-success" asp-action="resumo">
                Finalizar Pedido
            </a>
        </div>
    </div>
</div>

        <hr />
        <footer>
            <p>&copy; 2018 - Casa do Código</p>
        </footer>
```

```
        </footer>
    </div>

        <script src="/lib/jquery/dist/jquery.js"></script>
        <script src="/lib/jquery-validation/dist/jquery.validade.js"></script>
        <script src="/lib/jquery-validation-
unobtrusive/jquery.validade.unobtrusive.js"></script>
        <script src="/lib/bootstrap/dist/js/bootstrap.js"></script>
        <script src"/js/site.js?v=8ZRc1sGeVrPBx41D717BgRaQeKyh78QKV9SKsdt638U">
</script>

    <script type="text/javascript">
        function clickIncremento() {
            debugger;
        }
    </script>
}

</body>
</html>
```

O que Asp.Net Core faz é lançar a tag de script para o final da página, caso a ordem fosse outra, seria, executados os outros scripts principais da aplicação, e isso geraria uma série de erros e teremos dependências que ainda não forma executadas. Por isso é muito importante inserir o código JavaScript dentro de uma sessão.

De volta ao código html da página, notaremos os itens do carrinho que são exibidos como uma série de `<div>` s e `<span>` s, e não mencionam exatamente quais são os itens que estamos trabalhando.

```
<!DOCTYPE html>

<****!****>

<div class="col-md-3">PostgreeSQL</div>
<div class="col-md-2 text-center">R$ 49,90</div>
<div class="col-md-w text-center">
    <div class="input-group">
        <span class="input-group-btn">
            <button class="btn btn-default">
                <span class='glyphicon-minus"></span>
            </button>

<****!****>
```

A única informação que possuímos é o nome, no caso `PostgreeSQL` , mas isso não é relevante. O que precisamos saber é qual o item que foi clicado no botão "+". Para identificarmos cada um dos itens iremos colocar dentro de um elemento `<div>` um atributo com o identificador de cada item.

Na view `Carrinho.cshmtl` localizaremos a `<div>` que envolve cada um dos itens. Essa `<div>` possui a classe `linha produto` .

```
@{
    ViewData["Title"] = "Carrinho";
}
```

```
@foreach (var item in Model)
        {

            <div class="row row-center linha-produto">
                <div class="col-md-3">
                    <img class="img-produto-carrinho" src="/images/produtos/large_@(item.Produto.Coc
                </div>

<****!****>
```

Adicionaremos outro atributo, que chamaremos de `item-id`, que será igual ao id do item pedido. Inseriremos, portanto, `@item.Id`.

```
@{
    ViewData["Title"] = "Carrinho";
}


@foreach (var item in Model)
        {

            <div class="row row-center linha-produto" item-id="@item.Id">
                <div class="col-md-3">
                    <img class="img-produto-carrinho" src="/images/produtos/large_@(item.Produto.Coc
                </div>

<****!****>
```

Desse modo estamos inserido um código que será obtido do banco de dados, ou seja, estamos lançando uso do Razor para que ele renderize o número de identificação do item correspondente.

Salvaremos as modificações realizadas, atualizaremos a página e novamente iremos verificar o código fonte. Notaremos a presença do novo atributo `item-id`.

```
<!DOCTYPE html>

<****!****>

            <div class="row row center linha-produto" item-id"8">
                <div class="col-md-3">
```

Na view `Carrinho.cshtml` passaremos o `item-id` para o código JavaScript, de forma que esse item possa ser trabalhado e o servidor possa receber a informação e atualizar a quantidade de itens no carrinho. No evento `onclick` teremos a função `clickIncremento()`, para esta passaremos o contexto por meio da palavra `this`,.

```
@{
    ViewData["Title"] = "Carrinho";
}

                    <input type="text" value="@(item.Quantidade)"
                           class="form-control text-center" />
```

```
                    <span class="input-group-btn">
                        <button class="btn btn-default">
                                                    onclick="clickIncremento(this)">
                            <span class="glyphicon-plus"></span>
                        </button>
                    </span>
```

Além disso, devemos modificar a assinatura da função JavaScript, passando para `clickIncremento()` um parâmetro `btn`, isto é, o botão que foi clicado pelo usuário. Em seguida, removeremos o comando `debugger`.

```
@{
    ViewData["Title"] = "Carrinho";
}

@section Scripts
{

    <script type="text/javascript">
        function clickIncremento(btn) {

        }
    </script>
}
```

Para acessarmos o atributo id a partir desse elemento, usaremos a biblioteca **jQuery**, que é um componetente que acompanha a instalação do Asp.NET Core. O jQuery fornece diversas ferramentas que são úteis para, por exemplo, fazer requisições ao servidor. Inseriremos o caractere `$` e passaremos o elemento a ser manipulado, no caso trata-se do `btn`.

```
@{
    ViewData["Title"] = "Carrinho";
}

@section Scripts
{

    <script type="text/javascript">
        function clickIncremento(btn) {
            $(btn)
        }
    </script>
}
```

E seguida, acessaremos a função `attr`, uma das várias disponibilizadas pelo jQuery, e ela receberá o nome do atributo que desejamos trabalhar, ou seja, `item-id`. Todos esses valores serão atribuídos a uma variável que chamaremos de `idemId`. Finalmente, adicionaremos novamente o comando `debugger`.

```
@{
    ViewData["Title"] = "Carrinho";
}

@section Scripts
{
```

```
    <script type="text/javascript">
        function clickIncremento(btn) {
            var itemId = $(btn).attr('item-id');
            debugger;
        }
    </script>
}
```

Atualizaremos a página no browser, abriremos o código fonte e clicaremos sobre o botão "+". Nos atentaremos ao seguinte trecho do código:

```
<!DOCTYPE html>

<****!****>

    <script type="text/javascript">
        function clickIncremento(btn) { btn = button.btn.btn-default
            var itemI = $(btn).attr('item-id'); itemId = undefined
            debugger;
        }
    </script>

</body>
</html>
```

Note que o valor trazido é indefinido (`undefinded`), não foi possível encontrar o atributo `item-id` . O que fizemos foi clicar no botão, mas não é no botão que está o atributo.

Ao analisarmos o código fonte mais detalhadamente, perceberemos que o atributo `item-id` foi armazenado em uma `<div>` , e mais abaixo no código teremos a função `clickIncremento()` , e `<button class="btn btn-default>` foi passado como evento `onclick` . Porém, ao analisarmos a estrutura do código hmtl perceberemos que `item-id` está bem acima.

```
<!DOCTYPE html>

<****!****>

<div class="row row-center linha-produto"item-id="8">
                <div class="col-md-3">
                    <img class="img-produto-carrinho" src="/images/produtos/large_036.jpg" />
                </div>
                <div class="col-md-3">@Business Intelligence</div>
                <div class="col-md-2 text-center">R$ 49,90</div>
                <div class="col-md-2 text-center">
                    <div class="input-group">
                        <span class="input-group-btn">
                            <button class="btn btn-default">
                                <span class="glyphicon-minus"></span>
                            </button>
                        </span>
                        <input type="text" value="1"
                               class="form-control text-center" />
                        <span class="input-group-btn">
                            <button class="btn btn-default">
```

```
                                                onclick="clickIncremento(this)">
                                <span class="glyphicon-plus"></span>
                            </button>

    <****!****>
```

Nas próximas aulas veremos como localizar o `item-id="8"` por meio da hierarquia de elementos html por funções do jQuery.