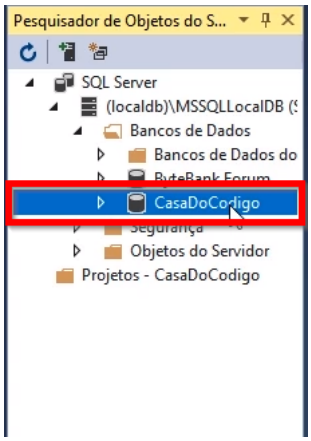
03

Chamando Função JavaScript a Partir de um evento HTML

Transcrição

Usaremos o mesmo projeto iniciado no curso Asp.NET Core 2.0 parte I, portanto copiaremos esse arquivo e o utilizaremos na segunda etapa do curso. Havíamos criado um banco de dados na primeira parte do curso, mas ele será apagado nesta etapa. No Visual Studio, clicaremos sobre "Exibir" no cabeçalho de ferramentas, e selecionaremos a opção "Pesquisador de Objetos no SQL Server". Assim, localizaremos o banco de dados `CasaDoCodigo` e o excluiremos. Basta clicar sobre ele com o botão direito e selecionar a opção "Excluir".



Deletamos esse banco de dados porque nós fizemos uma cópia do arquivo e trocamos o nome do *name space*, isso pode gerar problemas no momento em que precisarmos trabalhar com o banco.

Depois que deletamos o banco de dados iremos executar a aplicação.A aplicação rodará no *browser* ,e teremos o catálogo de livros da Casa do Código e a opção de adiciona-los ao carrinho de compras por via do botão "Adicionar". Na primeira opção apresentada, o livro "ASP.NET Core MVC", clicaremos sobre o botão "Adicionar".

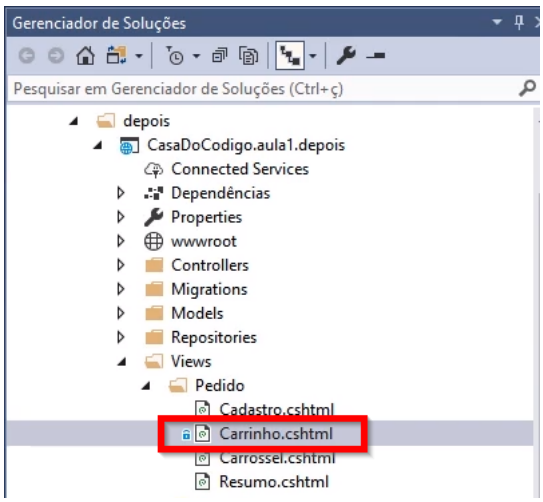
Adicionaremos mais alguns livros diferentes no carrinho. Ao realizarmos essa ação, somos direcionados para a página que exibe todos os itens que estamos comprando. Contudo, temos a opção de "Quantidade" de livros, e esta não pode ser alterada

diretamente ao clicarmos no botão "+", ficando sempre com o valor `1` fixo.

Nosso objetivo é fazer com que a aplicação responda ao clique no botão "+" para podermos aumentar as quantidades de itens no carrinho.

Até o momento trabalhamos com o código apenas do lado do servidor, o chamado Server Side. Neste ponto do curso, iremos fazer uma implementação do lado do cliente, utilizaremos a linguagem JavaScript, que somente pode ser executada o *browser*.

Abriremos novamente o Visual Studio e acessaremos a view de carrinho. Na área de gerenciamento de soluções, clicaremos sobre "Views > Carrinho.cshtml".



Iremos até o final da view e inseriremos o código JavaScript que será executado do browser. Não podemos simplesmente adicionar o script no código , primeiramente precisamos adicionar a tag `<script>` que por sua vez necessita de um tipo (`type`), no caso, `text/javascript`

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

<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>

<script type="text/javascript">

</script>
```

Dentro dessa tag iremos inserir o código JavaSript. Porém, há um detalhe: não podemos inserir uma tag `<script>` dentro de uma view do Razor, como é o casso de `Carrinho.cshtml` . Precisamos criar uma sessão para incluir esse script, para isso, deveremos criar uma diretiva `@section` que terá o nome de `Scripts` . Essa sessão terá duas chaves, e dentro delas inseriremos a tag `<script>` .

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

<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>

@section Scripts
{

    <script type="text/javascript">

    </script>
}
```

Com isso, podemos realmente começar a escrever nosso código JavaScript.

O objeto é fazer com que aplicação responde ao clique do usuário no botão "+", tal responde deve ser criada por meio de uma função (`function`), que terá o nome de `clickIncremento()` .

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


@section Scripts
{

    <script type="text/javascript">
```

```
        function clickIncremento(){

        }

    </script>
}
```

Iremos até o trecho de código responsável pelo botão "+" na view, que é `button class="btn btn-default"` .

```
@{
    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>
                <div class="col-md-3">@(item.Produto.Nome)</div>
                <div class="col-md-2 text-center">R$ @(item.PrecoUnitario)</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="@(item.Quantidade)"
                               class="form-control text-center" />
                        <span class="input-group-btn">
                            <button class="btn btn-default">
                                <span class="glyphicon-plus"></span>
                            </button>
                        </span>
                    </div>
                </div>
                <div class="col-md-2">
                    R$ <span class="pull-right" subtotal>
                        @(item.Quantidade * item.PrecoUnitario)
                    </span>
                </div>
            </div>

        }
```

Adicionaremos um atributo ao elemento `<button>` para o evento do click. Esse atributo é chamado de `onclick` , e ele receberá a função JavaScript que acabamos de criar, a `clickIncremento()` .

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

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

Voltaremos ao browser a atualizaremos a página utilizando o atalho "Ctrl + F5" e testaremos se a nossa modificação foi bem sucedida. Ao clicaremos sobre o botão de "+" veremos que nenhuma ação é realizada. Isso ocorre porque não solicitamos nenhuma ação, nós iremos inserir um *break point* no código JavaScript da view. Inseriremos o comando `debugger`.

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

@section Scripts
{

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

    </script>
}
```

Ao testarmos novamente o botão "+", notaremos que nada ocorre. Para que o comando `debugger` funcione, devemos entrar nas ferramentas do desenvolvedor utilizando o atalho "F12" no browser. Uma nova área irá surgir na parte inferior da tela. Clicaremos no botão "+" e teremos acesso ao seguinte trecho de código:

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

</body>
</html>
```

Estamos parados da linha do comando `debugger`. Veremos nas próximas aulas como implementar o código JavaScript em nossa aplicação.