

Passando código de produto na rota

Transcrição

Quando utilizamos uma *tag helper* de âncora, normalmente acrescentamos a informação da *action* do *controller* também, a qual será chamada ao clicarmos no link. Felizmente, a *tag helper* possui uma conveniência: se você estiver navegando pelo mesmo *controller*, é possível omitir esta informação, e é isto que faremos aqui.

Eliminaremos o atributo `asp-controller` pois estamos sempre no mesmo *controller*, o **pedido**, mantendo apenas a `asp-action` dentro das tags `<div>`:

```
<div class="text-center">
  <a asp-action="carrinho"
    class="btn btn-success">Adicionar</a>
</div>
```

Em seguida, colocaremos o atributo `asp-action` nos demais links das páginas. Em `Carrinho.cshtml`, há o link para a adição de produtos, um elemento âncora do HTML, `href="/"`, que transformaremos em uma *tag helper* por meio do atributo `asp-action`.

```
<div class="pull-right">
  <a class="btn btn-success" asp-action="carrossel">
    Adicionar Produtos
  </a>
  <a class="btn btn-success" asp-action="cadastro">
    Preencher Cadastro
  </a>
</div>
```

Na página `Cadastro.cshtml`, iremos à parte onde se encontram os botões para continuar comprando e finalizar pedido, em que também trocaremos os `href="/"`:

```
<div class="form-group">
  <a class="btn btn-success" asp-action="carrossel">
    Continuar Comprando
  </a>
</div>
```

Vamos rodar a aplicação para verificar seu comportamento. Clicaremos no botão "Adicionar" para incluirmos o produto no carrinho e, nesta página, poderemos pressionar "Adicionar Produtos" para voltarmos à página anterior e continuar vendo o catálogo.

Da mesma forma, conseguimos clicar no botão "Preencher Cadastro" para navegarmos à tela de cadastro. Para continuarmos comprando, existe o botão "Continuar Comprando". E, no momento de adição do produto, queremos passar a informação de código do produto que está sendo comprado à página de carrinho.

De que forma passaremos tal informação utilizando uma *tag helper*?

Acessaremos a *tag helper* de `Carrossel.cshtml` e, onde há `asp-action`, colocaremos outro atributo indicando a informação adicional, no caso, o código do produto. Porém, para passarmos qualquer informação à *tag helper*, é necessário incluir um prefixo chamado `asp-route-`, seguido do nome do parâmetro que estamos passando, `codigo`, e no final colocaremos o valor que está sendo passado, a propriedade `@produto.Codigo`:

```
<div class="text-center">
  <a asp-action="carrinho"
    asp-route-codigo="@produto.Codigo"
    class="btn btn-success">Adicionar</a>
</div>
```

Vamos salvar, voltar ao navegador e atualizar a página. Ao passarmos o mouse sobre o botão "Adicionar" de um determinado produto, o link exibido envolve `/carrinho?codigo=001`, e se passarmos a outro produto, o código na URL gerada será outro. Percebemos que o código está sendo montado como uma *query string*.

É importante passarmos o código, porém existem alternativas para fazê-lo de forma mais elegante. Então, em vez de `?codigo=036`, por exemplo, poderíamos ter simplesmente `/036`. Queremos incorporar o código do produto na rota, portanto pausaremos a aplicação e faremos uma modificação no nosso arquivo `Startup.cs`.

No fim do arquivo, encontraremos a definição da rota, com `controller=Pedido`, `action=Carrossel` e `id`, que seria o atributo passado como rota, o qual substituiremos com `codigo`:

```
routes.MapRoute(
    name: "default",
    template: "{controller=Pedido}/{action=Carrossel}/{codigo?}");
```

Vamos rodar a aplicação mais uma vez. Cairemos novamente na página de carrossel, desta vez com a mudança na hora de clicarmos nos botões de adicionar produto ao carrinho, com a URL que está sendo gerada incorporando o código do produto na rota do ASP.NET Core MVC.