

Validando Campos do Formulário no Cliente

Transcrição

Conseguimos passar `modelo` para a view, e de volta recebemos no `PedidoController` as informações do cadastro por meio da action de `resumo`, contudo precisamos prepará-la para lidar corretamente com esse tipo de requisição.

Ao entrarmos no browser, perceberemos que é possível acessar a action por meio de uma url digitada na barra de endereço: `localhost:50040/Pedido/resumo`.

Contudo, essa não deveria ser uma possibilidade, pois deveríamos conseguir chegar até `resumo` a partir do envio de um formulário, feito por uma requisição `POST`.

Por conseguinte iremos restringir o acesso à action inserindo uma notação com o atributo `http` na `PedidoController.cs`, o que irá impedir a chamada direta do browser.

```
[HttpPost]
public IActionResult Resumo()
{
    return View(pedidoRepository.GetPedido());
}
```

Outra modificação a ser realizada na action é a inserção de um parâmetro para recebermos os dados de cadastro.

Inseriremos o parâmetro `cadastro` e criaremos um break point na linha `return View(pedidoRepository.GetPedido());`.

```
[HttpPost]
public IActionResult Resumo(Cadastro cadastro)
{
    return View(pedidoRepository.GetPedido());
}
```

Pressionaremos a tecla "F5" para compilar a aplicação. Feito isso, iremos abrir a página principal no browser e selecionaremos um produto qualquer, seremos direcionados à página de carrinho e clicaremos sobre o botão "Preencher Cadastro". Com isso, seremos direcionados para a página de formulário.

Iremos preencher alguns campos do cadastro, são eles "Nome do Cliente", "Email" e "Telefone". Preenchidos os campos, clicaremos sobre o botão "Finalizar Pedido", para avaliarmos o resultado.

Quando clicamos no botão de finalização de pedido, não conseguimos acessar a action de resumo. Isso ocorre porque temos uma série de validações sendo feitas pelo cliente. Caso inspecionemos a caixa de texto "Endereço" por exemplo, veremos que há um elemento hmtl input com uma série de atributos de validação. Tais atributos são utilizados pelo JavaScript do jQuery Validation Unobtrusive - um dos frameworks utilizados no projeto - como condição para que o cliente avance nas próximas etapas de compra.

```
<input type="text" class="form-control input-validation-error" id="endereco" placeholder="Endereço">
```

Precisamos exibir para o usuário uma mensagem de que os campos de cadastro não estão corretamente preenchidos, para isso incluiremos alguns elementos html na view de cadastro. Abriremos `Cadastro.cshtml` e para cada campo do formulário, incluiremos um elemento html que indicará que o campo possui um problema de preenchimento.

Logo depois do `<input>`, incluiremos um elemento `` que terá uma classe CSS denominada `text-danger`, que por sua vez irá marcar o texto com a cor vermelha. Feito isso, transformaremos a classe em uma tag helper, incluindo o atributo específico `asp-validation-for` que receberá o valor `@Model.Nome` .

```

<div class="form-group">
    <label class="control-label" for="telefone">Telefone</label>
    <input type="text" class="form-control" id="telefone" placeholder="Telefone"
           asp-for="@Model.Telefone">
    <span asp-validation-for="@Model.Telefone" class="text-danger"></span>
</div>
</div>
<div class="col-md-4">
    <div class="form-group">
        <label class="control-label" for="endereco">Endereço</label>
        <input type="text" class="form-control" id="endereco" placeholder="Endereço"
               asp-for="@Model.Endereco">
        <span asp-validation-for="@Model.Endereco" class="text-danger"></span>
    </div>
<*****!*****>

```

No caso de `UF` temos um `select`, portanto depois dele iremos inserir o `` com a propriedade `@Model.UF` .

```

<div class="form-group">
    <label class="control-label" for="UF">UF</label>
    <select class="form-control single-select" name="UF"
           asp-for="@Model.UF">
        <option selected value="">[Selecione o estado]</option>
        <option value="AC">Acre</option>
        <option value="AL">Alagoas</option>
        <option value="AP">Amapá</option>
        <option value="AM">Amazonas</option>
        <*****!*****>
    </select>
    <span asp-validation-for="@Model.UF" class="text-danger"></span>

```

Com as modificações realizadas em todos os campos, iremos atualizar a página e clicaremos na opção "Finalizar Pedido", na página de cadastro. Veremos que todos os campos que não foram preenchidos surgem com a mensagem em vermelho "The field is required", dessa forma orientamos o usuário a preencher os campos corretamente.

