

Criando Método POST no Servidor para Receber Requisição

Transcrição

Temos em nosso código JavaScript que obtém as informações necessárias para realizar a alteração da quantidade de itens no carrinho. Devemos passar as informações para o servidor, mas antes de fazer a requisição devemos prepará-lo para receber essa requisição.

Faremos a preparação em `PedidoController.cs`. Ao final do código, criaremos um novo método público, e a princípio não declararemos nada. O nome desse método será `UpdateQuantidade()`

```
namespace CasaDoCodigo.Controllers
{
    <****!****>

    public IActionResult Resumo()
    {
        return View(pedidoRepository.GetPedido());
    }

    public void UpdateQuantidade()
    {
    }
}
```

O método `UpdateQuantidade()` irá tratar a requisição, e receberá como parâmetros um valor interio (`int`) de `itemPedidoId` e a nova quantidade de itens, que chamaremos de `quantidade`.

```
namespace CasaDoCodigo.Controllers
{
    <****!****>

    public IActionResult Resumo()
    {
        return View(pedidoRepository.GetPedido());
    }

    public void UpdateQuantidade(int item PedidoId, int quantidade)
    {
    }
}
```

Adicionaremos um break point, e faremos uma requisição diretamente no browser pela barra de endereços, dessa forma podemos testar nosso método. Na barra de endereço digitaremos:

<http://localhost:50040/pedido/updatequantidade>

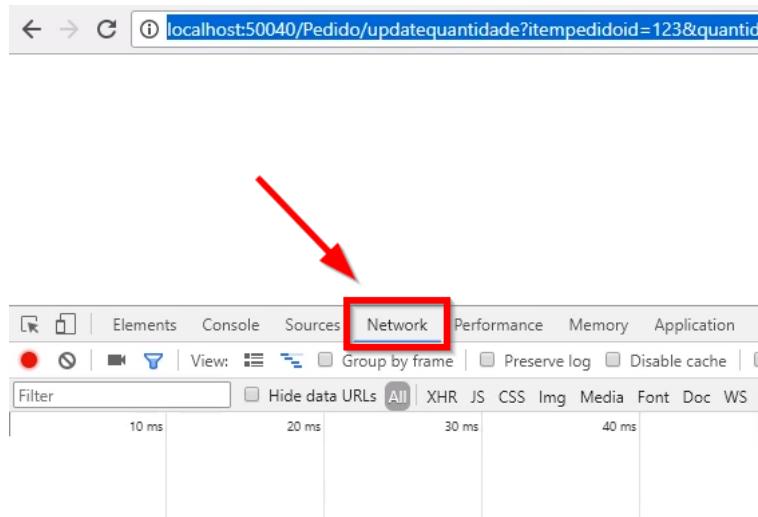
Veremos que a aplicação parou no break point do método `updatequantidade()`, passando como parâmetro `itemPedidoId`, que é 0 e `quantidade` também 0, pois estes são os valores default já que não passamos nada para esses parâmetros.

Faremos uma nova requisição diretamente no browser, mas dessa vez passando parâmetros. Utilizaremos a ?, que marca o começo da dos parâmetros, depois inseriremos `itemPedidoId` e configuraremos um valor qualquer de 123. Em seguida inseriremos o segundo parâmetro `quantidade` de valor 7, e que será dividido do primeiro parâmetro pelo caractere &,

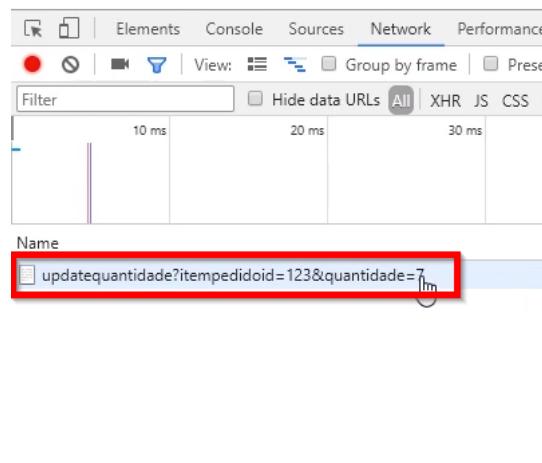
<http://localhost:50040/pedido/updatequantidade?itempedidoId=123&quantidade=7>

Conseguimos passar com sucesso os parâmetros para o método. No entanto, quando formos realizar de fato a requisição não a passaremos diretamente na **query string**, isto é, todo o conteúdo após a ? que contém os parâmetros da chamada.

Queremos que esses dados sejam passados no corpo da chamada. Executaremos novamente a aplicação e acessaremos o corpo do desenvolvedor pressionando o atalho "F12". Antes de fazermos a chamada para a requisição, selecionaremos a opção "Network" no browser Chrome.



Agora que a aba network foi selecionada, faremos a requisição novamente. Voltando ao browser, veremos a requisição registrada.



Ao clicarmos sobre ela, teremos acesso a alguns detalhes, como sua URL e seu método, no caso, "GET".

The screenshot shows a browser developer tools Network tab. A single request is listed with the name "updatequantidade?itempedid...". The Headers tab is selected, showing the following details:

- Request URL: `http://localhost:50040/Pedido/updatequantidade?itempedidoid=123&quantidade=7`
- Request Method: `GET`
- Status Code: `200 OK`
- Remote Address: `[::1]:50040`
- Referrer Policy: `no-referrer-when-downgrade`

Below the Headers tab, there are sections for Response Headers and Request Headers.

GET é um verbo http, isto é, uma das formas de realizar uma requisição http. Os métodos mais comuns são GET, POST, PUT, DELETE e alguns outros. O método de requisição é a maneira como ela é realizada para o servidor.

Como já havíamos dito, queremos que as informações no corpo da requisição e reforçaremos essa regra no nosso método. Ao invés de realizarmos uma requisição GET, iremos fazer uma do tipo POST, que exige que os parâmetros sejam enviados no corpo da requisição. Além disso, uma requisição da natureza http POST é ideal para realizar alterações em alguma informação do sistema, enquanto a GET é mais interessante para simplesmente obter dados.

Em PedidoController adicionaremos um atributo de método que chamaremos de `HttpPost`.

```
namespace CasaDoCodigo.Controllers
{
    <*****!*****>

    public IActionResult Resumo()
    {
        return View(pedidoRepository.GetPedido());
    }

    [HttpPost]
    public void UpdateQuantidade(int itemPedidoId, int quantidade)
    {
    }
}
```

Ao realizarmos novamente a requisição no browser, veremos que a aplicação não conseguiu identificar uma página para responder ao endereço `updatequantidade`.

localhost:50040/Pedido/updatequantidade?itempedidoid=123&quantidade=7



Não foi possível encontrar a página deste localhost

Nenhuma página da web foi encontrada para o endereço da
Web:<http://localhost:50040/Pedido/updatequantidade?itempedidoid=123&quantidade=7>

Pesquise [localhost 50040 Pedido update quantidade](#) no Google

HTTP ERROR 404

Nas próximas aulas veremos como fazer uma requisição com o método http POST a partir do código JavaScript.