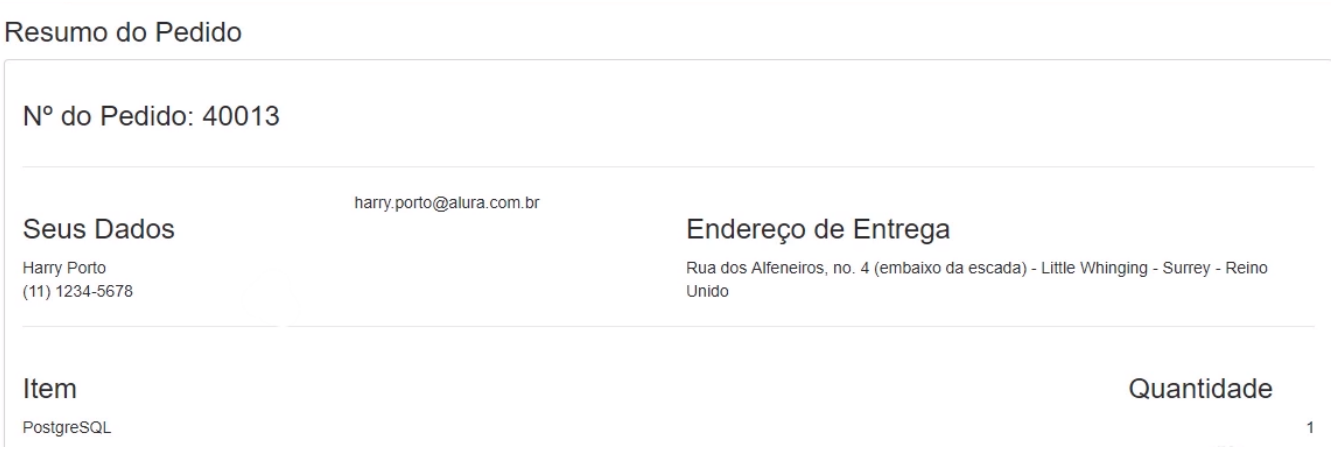
02

Protegendo uma Action com ValidateAntiForgeryToken

Transcrição

Testaremos a gravação das informações cadastrais no banco de dados. No browser, preencheremos todos os campos obrigatórios na página de cadastro e ao final clicaremos sobre o botão "Finalizar Pedido".

Uma vez que finalizarmos o cadastro, seremos encaminhados para a view de resumo. Neste ponto, todo o processo de transferência de dados do cadastro para o repositório e gravação no banco de dados já foi realizado.



De volta ao Visual Studio, acessaremos os dados da tabela `dbo.Cadastro` para conferir os dados gravados. Essa arquivo encontra-se na pasta `Tabelas` do projeto. Veremos todas as informações que foram preenchidas nos campos de cadastro.

Id	Bairro	CEP	Complemento	Email	Endereco	Municipio	Nome	Telefone	UF
40009									
40010									
40012									
40013	Vila Mariana	15151515151	fundos	fulano@gmail.com	Rua Vergueiro,...	sao paulo	fulado de tal	12345678	SP
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

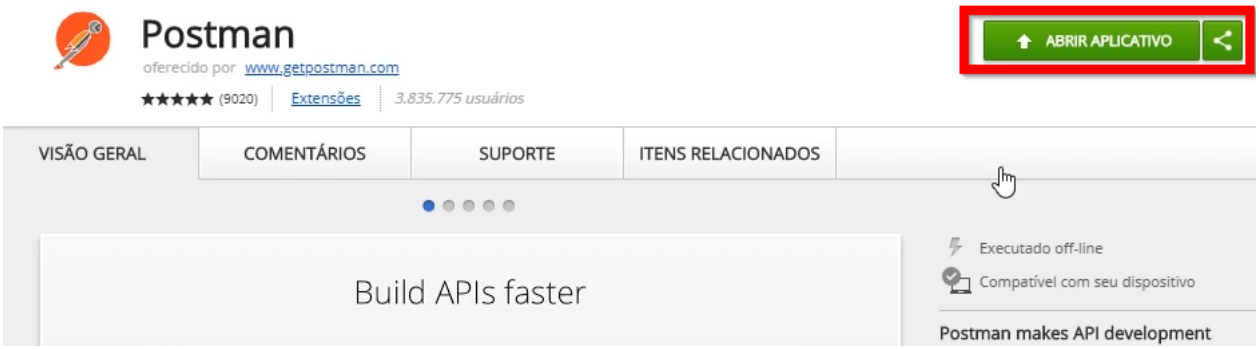
Contudo devemos refletir: **será que nosso sistema está bem protegido contra acessos externos? **

Marcamos a action com o acesso do tipo "POST", o que impede que qualquer usuário acesse a action de resumo por meio da URL, mas isso não é o suficiente, pois podemos sofrer um ataque conhecido como **CSRF**, ***Cros-site request forgery***, em uma tradução livre "Falsificação de solicitação entre sites". Quando recebemos uma requisição proveniente de fora do nosso site, existe a possibilidade de que se trata de um agente malicioso, com interesse no furto de dados ou algo do gênero.

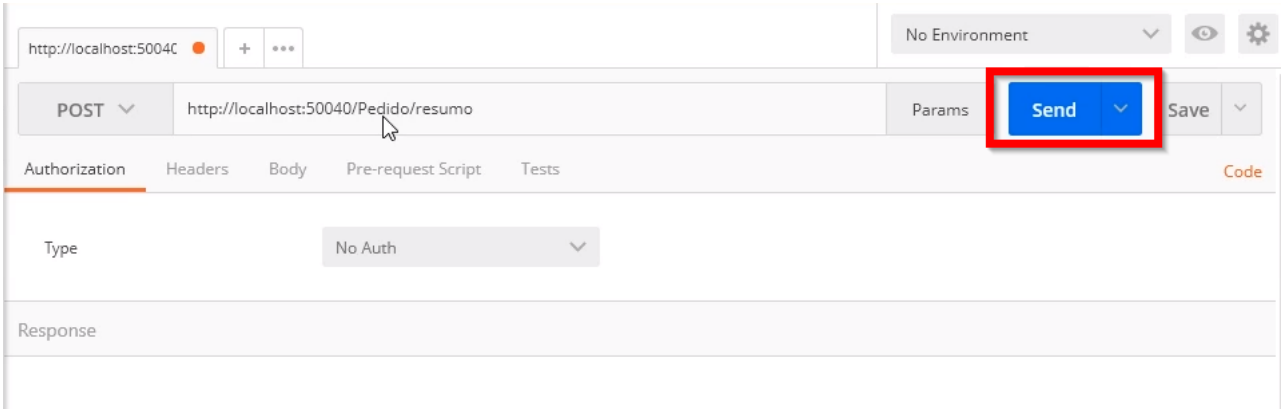
De volta a tela de cadastro veremos a existência de vários campos, mas temos um campo oculto. Ao inspecionarmos a página de cadastro, veremos um `<input>` marcado com o valor `hidden` , isto é, "oculto". Em seguida, teremos uma informação criptografada, conhecida por **token de verificação**. Essa informação precisa ser validada pelo servidor, caso contrário podemos sofrer um ataque externo.

```
<input name="__RequestVerificationToken" type="hidden" value=
"CfDJ8DGA9YtWRddDkFJM71llQFM7JaUU49wEH-y8MjecfFu7xM9EMQTS39cHlzZDBJI5_QppOv3dSWg5fGyvg11QWqImS4IOPR6
```

Simularemos um ataque por meio de uma ferramenta chamada **Postman** para Chrome. Basta procurar o termo no Google e realizar o download e depois abrir o programa.



Este é um recurso bem interessante para simularmos requisições. Na página inicial do programa, inseriremos no campo "POST" - que configuramos como sendo o tipo de acesso que utilizamos - o endereço da action de resumo: `http://localhost:50040/Pedido/resumo` . Depois, clicaremos sobre o botão "Send".



Obteremos a resposta da action de resumo.

Em `PedidoController.cs` inseriremos um break point na linha `if (ModelState.IsValid)` .

```
[HttpPost]
public IActionResult Resumo(Cadastro cadastro)
{
    if (ModelState.IsValid)
    {
        return View(pedidoRepository.UpdateCadastro(cadastro));
```

```
        }
        return RedirectToAction("Cadastro");
    }
```

Faremos novamente a requisição no programa Postman clicando sobre o botão "Send". Seremos redirecionados novamente para a action `Resumo` . Isso pode ser um agente mal intencionado querendo acessar o site. Nesses casos, devemos utilizar o token criptografado na view de cadastro para garantirmos a segurança da aplicação.

Pararemos o programa e inseriremos na action o atributo `ValidateAntiFogeryToken` que valida o tolken no momento em que recebemos uma requisição.

```
[HttpPost]
[ValidateAntiFogeryToken]
public IActionResult Resumo(Cadastro cadastro)
{
    if (ModelState.IsValid)
    {
        return View(pedidoRepository.UpdateCadastro(cadastro));
    }
    return RedirectToAction("Cadastro");
}
```

Pressionaremos o atalho "F5" para compilarmos a aplicação e seremos direcionados para a página principal, escolheremos um produto qualquer e preencheremos o formulário de cadastro.

Iremos até o Postman e tentearemos fazer uma nova requisição. Dessa vez,o programa acusou o status da requisição como "400 Bad Request", afinal não foi permitido o acesso pelo servidor, pois não fornecemos o token por meio do Postman.