01

Criando classe UsuarioDTO

Transcrição

[00:00] Nós cadastramos na aplicação a Joviane e a ela fez a alteração do atributo roles da classe usuário para que ela fosse cadastrada na nossa aplicação como um usuário administrador. Então vamos verificar o porquê que essa nossa aplicação do Alura shows está apresentando essa vulnerabilidade e como que nós vamos conseguir corrigir.

[00:18] Para isso, vamos voltar na nossa parte de formulário para registrar esse novo usuário. Então nós temos o quê? Nós temos nessa aba de registro do usuário esses campos do nome, do e-mail, a senha e a imagem do perfil desse nosso usuário. Essas informações serão passadas para aquele método registrar que está na nossa classe usuário controller.

[00:41] Então vamos nossa classe usuário controller, só para eu colocar Ctrl + Shift + R", usuário controller, então nós temos aqui esse nosso método registrar. Então nesse nosso método registrar, nós estamos fazendo o quê? Associação desses dados que serão passados no formulário com quem? Com essa nossa classe modelo usuário.

[01:03] Essa classe modelo usuário tem quais atributos? O e-mail, a senha, o nome, o nome da imagem, até aí tudo bem, de fato nós estamos esperando receber esses dados do formulário, mas nós também temos como atributo dessa classe usuário quem? Os roles. Então essas roles não devem ser manipuladas de nenhuma forma pelo nosso usuário.

[01:19] Mas pelo fato de nós estarmos fazendo associação direta dessa classe usuário com os dados que são recebidos do formulário, nós temos o quê? Nós temos aquele caso do Mass Assignment onde a Joviane conseguiu fazer a alteração, a manipulação desse atributo roles para que ela se cadastrasse na nossa aplicação como se ela fosse um usuário administrador.

[01:39] Como é que nós conseguimos corrigir isso? Existem algumas formas que nós poderíamos estar protegendo essa nossa aplicação contra esse ataque do Mass Assignment. Uma das formas seria a seguinte, o que nós poderíamos fazer? Nós poderíamos criar uma classe intermediária, somente com os atributos que nós esperamos estar recebendo do formulário.

[02:02] Então o que nós esperamos receber do formulário? O nome, o e-mail, a senha e a imagem do perfil. Então nós vamos montar uma classe só com esses atributos que nós esperamos estar recebendo do formulário e nessa classe intermediária nós vamos passar esses dados, nós vamos transferir esses dados recebidos do formulário, para essa nossa classe modelo usuário para que seja criado esse nosso objeto usuário sem estar expondo esse atributo roles para o formulário da nossa aplicação.

[02:27] Esse é um padrão de projeto que recebe o nome de Data Transfer Object ou pela abreviação DTO. Então essa classe que nós vamos criar só é responsável por estar realizando essa transferência de dados para montar esse nosso objeto do tipo usuário, evitando expor esses atributos no caso dos atributos roles diretamente no formulário da nossa aplicação.

[02:58] Então vamos no nosso pacote de modelo e nós vamos criar essa classe usuário DTO. Então "Ctrl + N", nós colocamos class e vamos criar essa nossa classe usuário DTO.

[03:12] Essa classe usuário DTO deve ter só aqueles quatro atributos que é o que nós devemos receber informação do formulário. Então vamos aproveitar aqui, vamos voltar na nossa classe usuário, nós vamos copiar esses quatro parâmetros que é o que nós esperamos estar recebendo no nosso formulário.

[03:28] Uma vez que nós fizemos isso, nós temos que configurar os setters, para setar os valores vindos do formulário nesses nossos atributos aqui da classe. Nós não vamos precisar dos getters, nesse caso, nós só precisamos dos setters para acertar os valores desses parâmetros no formulário nesses atributos e depois nós vamos transferir para nossa para montar o nosso objeto do tipo usuário.

[03:52] Para configurar os setters, nós colocamos aqui "Ctrl + 3", Ggs, para Generate Getters and Setters, e basta nós clicarmos nessa aba para selecionar somente os setters que já vai atender o que nós precisamos. Então nós colocamos "Ok". E aqui agora nós já temos os atributos aqui com os dados vindos do formulário. Então agora, uma vez que nós recebermos esses dados vindos do formulário, nós temos que passar, temos que transferir esses dados para construir esse nosso objeto usuário.

[04:24] Aqui dentro dessa nossa classe usuário DTO, eu vou criar outro método aqui para passarmos esses dados para criar esse objeto do tipo usuário. Então vou colocar aqui public e esse método vai retornar um objeto aqui do tipo usuário e nós vamos chamar esse método de monta usuário.

[04:39] Então nesse aqui, dentro desse método, o que nós queremos fazer? Nós queremos passar esses atributos para montar esse nosso objeto usuário. Então vamos instancear a nossa classe usuário e nós vamos passar como, aqui no construtor, nessa classe, os atributos dessa nossa classe usuário DTO, que são o e-mail, a senha, nós temos um nome e imagem.

[05:05] Vamos só colocar aqui, ponto e vírgula, e nós temos essa reclamação. Isso por quê? Porque nós não temos, na classe usuário, esse construtor com esses parâmetros, então nós temos que vir aqui "Ctrl + 1", e nós queremos criar esse construtor aqui na nossa classe usuário.

[05:24] Vamos só mudar o nome dos parâmetros aqui para ficar exatamente igual ao que nós temos nessa nossa classe usuário, e agora nós queremos fazer o quê? Nós vamos chamar simplesmente "this.email", vai receber aqui um e-mail aqui que nós possamos contar como argumento nesse construtor, o "this.senha", que vai receber a senha, o "this.nome=nome" e o "this.nomeimagem = nomeimagem".

[05:52] E agora nós voltamos para nossa classe usuário DTO e nós retornamos, nós podemos vir aqui e retornar esse nosso objeto usuário aqui. Então com isso nós configuramos já essa nossa classe usuário DTO que só vai ter como atributo quem? O e-mail, a senha, o nome e o nome da imagem do perfil, somente são as informações que nós estamos realmente esperando receber do formulário.

[06:21] Agora nós temos que fazer o quê? Temos que voltar na nossa classe usuário controller e passar, nesse método registrar, que agora as informações do formulário vão ser associadas com essa classe intermediária que nós criamos que é a usuário DTO.

[06:38] Nós vamos aqui e colocamos que agora esses dados serão associados com essa classe usuário DTO, vamos só mudar o nome também do parâmetro, usuário DTO, e aqui nós podemos chamar o método que nós configuramos nessa classe usuário DTO para criar esse nosso objeto usuário.

[06:56] Nós vamos aqui, usuário DTO, e chamamos o método "montaUsuário" que vai nos retornar quem? Um objeto do tipo usuário e nós aqui, internamente, estamos utilizando essa variável usuário registro, nós podemos vir aqui e colocar esse nome dessa nossa variável como usuário registro e nós devemos ter tudo funcionando.

[07:15] Vamos testar lá? Então vamos na nossa aplicação agora e nós vamos verificar se nós conseguimos fazer essa manipulação do atributo roles. Se tudo deu certo, o que é que nós esperamos? Bom, a classe usuário DTO não tem mais esse atributo roles, então se não tem mais esse atributo roles, nós não estamos expondo ele para o formulário e nós devemos ter conseguido proteger nossa aplicação. É isso que nós esperamos, vamos só confirmar se é isso que de fato vai acontecer.

[07:42] Só esperar aqui, inicializar o Tomcat, só mais alguns segundos. O Tomcat terminou de inicializar. Então vamos voltar para nossa aplicação aqui e nós vamos registrar um outro usuário que vai ser o Pedro. Então coloca aqui o nome do Pedro, nós colocamos o e-mail do Pedro, pedro@gmail.com, e vamos cadastrar aqui uma senha do Pedro, vamos colocar, por exemplo, 1 2 3 4 5 6, 1 2 3 4 5 6 e a imagem do perfil dele, daqui do Pedro, está aqui e aí o Pedro vai tentar fazer aquela mesma alteração, aquela mesma manipulação que a Joviane tinha feito.

[08:24] Então vai vir aqui e com o botão direito do mouse, inspecionar, ele clica aqui na tag formulário do registro e ele vem aqui edit HTML e ele vai colocar aquela mesma configuração que a Joviane tinha posto. Então ele vai vir aqui, "<input type"hidden" e ele vai e tenta manipular o atributo roles. Então ele vem aqui, "name = "roles(o).name" e ele coloca o valor como sendo o quê? De role admin. E ele fecha essa tag aqui.

[09:04] Está a mesma configuração que a Joviane tinha feito e a Joviane tinha conseguido fazer essa manipulação, porque nós estávamos fazendo diretamente a associação da nossa classe usuário aqui para ser mapeado com esses parâmetros que são passados do formulário.

[09:19] Agora vamos verificar se, ao criar essa pasta, usuário DTO, se nós vamos ter conseguido proteger a nossa aplicação. Então aqui o Pedro fez a mesma coisa, ele vai registrar, vamos só voltar aqui para a aplicação. Então Pedro já foi registrado, ele está aqui. Agora vamos voltar para MySQL Workbench e vamos verificar qual que é o perfil aqui que foi dado para o nosso usuário Pedro.

[09:44] Então vamos aqui só rodar essa query novamente. O Pedro foi registrado na aplicação, mas olha só qual que é agora o único perfil dele? Ele foi cadastrado aqui com o perfil role user. Então com isso, a tentativa agora do Pedro não surtiu mais efeito. Por quê? Por que nós estamos mapeando quem para a nossa parte do formulário?

[10:07] É a classe usuário DTO. E a nossa classe usuário DTO não tem mais esse atributo roles. O atributo roles continua na nossa classe usuário que nós não estamos mais expondo diretamente essa nossa classe usuário para os parâmetros que serão recebidos do formulário. Nós conseguimos já proteger a nossa aplicação.

[10:25] Vamos só confirmar que o Pedro não vai ter acesso à parte administrativa? Vamos voltar na aplicação e vamos só fazer logout. E o Pedro vai tentar fazer o login na parte administrativa. Então ele coloca o e-mail dele, pedro@gmail.com, e a senha era 123456, nós colocamos aqui login, e permissão negada.

[10:46] Então de fato nós já conseguimos proteger a nossa aplicação. Então nós vimos aqui que com essa configuração dessa classe DTO, nós já conseguimos fazer a proteção contra o Mass Assignment. Só que como nós estamos usando spring, nós podemos facilitar um pouco esse processo. Nós vamos ver na sequência como ficaria esse processo utilizando aqui o recurso do Spring mvc.