05

Usando método setAllowedFields

Transcrição

[00:00] Nós criamos aqui a classe usuário DTO somente com os atributos que nós estamos querendo associar com o nosso formulário de registro. E com isso, nós evitamos dispor atributos que não devem ser manipulados pelo usuário.

[00:14] E nós conseguimos fazer essa configuração, e funcionaria independentemente do framework com o qual nós estejamos trabalhando. Mas como nós estamos aqui no nosso projeto trabalhando com Spring, ele já vai nos ajudar nessa tarefa de uma maneira um pouco mais fácil até do que configurar essa classe do usuário DTO.

[00:30] Então o que é que nós poderíamos fazer? Aqui na nossa classe usuário controller, nós poderíamos usar aquela lotação do init binder para configurar quais são os atributos da nossa classe que nós permitimos que sejam manipulados pelo usuário.

[00:44] Vamos fazer isso? Então eu vou colocar aqui aquela anotação do init binder, essa anotação vai indicar que sempre esse método que nós vamos descrever agora deve ser executado quando uma requisição cair nesse controller e nós vamos falar que somente aqueles quatro atributos que nós temos na classe usuário é que vão poder ser editados, aquele atributo roles não vai ter permissão de ser editado pelo usuário.

[01:09] O que nós podemos fazer? Nós colocaríamos “void initBinder", que vai receber um objeto do tipo “(WebDataBinder binder)”, e o que é que nós queremos fazer? Nós só queremos permitir determinados campos, determinados atributos da nossa classe usuário de serem manipulados pelo usuário quando ele fizer o registro no formulário.

[01:32] Quais seriam eles? Seria o nome, o e-mail, o nome da imagem do perfil e a senha. Então podemos vir aqui e chamar esse método dizendo “SetAllowedFields”, quais seriam os campos, os atributos, da nossa classe que nós permitimos que sejam configurados, que sejam manipulados pelo nosso usuário quando ele fizer o registro, que seriam eles, como nós falamos, aqueles quatro: nome, e-mail, nós temos a senha e nós temos aqui por último o nome da imagem.

[02:01] Então aqui, se nós voltarmos para nossa classe, “Ctrl + Shift + R”, nossa face modelo, usuário, nós, com isso, só estamos permitindo o quê? Só está permitindo justamente a manipulação desses quatro atributos. Esse último atributo, roles, nós não estamos configurando para ele ser um campo, um atributo, permitido de ser editado no formulário.

[02:23] Com isso, nós poderíamos vir aqui e voltar a associar diretamente à nossa classe usuário, nós não precisaríamos utilizar a nossa classe usuário DTO. Eu posso até voltar aqui o nome desse parâmetro como sendo usuário registro, que é o que nós estamos trabalhando aqui, nós podemos vir comentar essa linha que nós tínhamos feito para chamada do nosso objeto usuário DTO.

[02:46] Com isso, nós já devemos ter o mesmo comportamento que nós tivemos anteriormente com a classe usuário DTO. Vamos só confirmar se tudo continua funcionando como deveria? Vou aqui, reiniciar o Tomcat, só esperar alguns segundos o Tomcat inicializar, e nós vamos tentar fazer esse mesmo registro de usuário fazendo aquele mesmo teste para verificar se nós conseguimos manipular o atributo roles.

[03:13] Então o Tomcat já terminou de ser inicializado, então vamos voltar para a nossa aplicação da Alura shows. Vamos na aba usuário, e nós vamos registrar esse novo usuário agora que vai ser o Rodrigo. Então nós colocamos aqui Rodrigo, o e-mail do Rodrigo é rodrigo@gmail.com, a senha do Rodrigo, eu vou colocar aqui 0123456789. E nós colocamos a foto do perfil do Rodrigo e ele vai fazer aquele mesmo teste que a Joviane e o Pedro já fizeram.

[03:45] Então vamos clicar com o botão direito do mouse, inspecionar e aqui nós vamos procurar o nosso formulário de registro, clicamos com o botão direito do mouse, edit html e nós vamos colocar aquela tag para tentar manipular o atributo roles, que a Joviane e o Pedro já fizeram. É aquela input type hidden, nós coloca name e nós pegamos o atributo roles, posição zero, queremos mudar o atributo name, da classe role, para que o valor dela seja um role admin, exatamente igual ao que nós já fizemos.

[04:21] Deixa eu só confirmar que está tudo certo. Então aqui nós fizemos essa configuração e agora nós vamos tentar fazer o registro desse nosso usuário, Rodrigo. Registrar. O Rodrigo foi registrado na Alura shows.

[04:42] Agora vamos voltar para o banco e vamos verificar como é que está o perfil, a role, que foi atribuída para o Rodrigo. Então vamos clicar com o botão direito do mouse aqui nessa tabela usuário role, select rows e o Rodrigo foi cadastrado aqui na nossa aplicação. Mas olha só, a única role que tem agora para o Rodrigo continua sendo quem? A role user.

[05:09] Com isso, com essa chamada desse método, do “setAllowedFields”, nós estamos falando quais são os atributos da nossa classe que podem ser manipulados pelo usuário, mesmo que o usuário tentou, que nem o Rodrigo agora, manipular o formulário html para mudar o atributo roles que tudo bem, tem aqui na nossa classe usuário, nós estamos configurando somente esses quatro atributos da nossa classe usuário é que podem ser editados.

[05:33] Então vamos só confirmar se o Rodrigo não consegue acessar de fato a parte administrativa. Então vamos só fazer o logout, e nós vamos vir aqui e o Rodrigo vai tentar fazer o login na parte administrativa da Alura shows.

[05:49] Ele coloca o e-mail dele e a senha dele era 0123456789. E nós temos aqui login, e permissão negada. Então de fato, nós conseguimos proteger a nossa aplicação contra esse ataque do Mass Assignment, agora utilizando esse recurso do Spring mvc. E nós conseguimos ter uma forma um pouco mais fácil de fazer essa proteção.