06

Mãos à obra: Usando o método setAllowedFields

Para proteger contra o ataque de **Mass assignment** nós criamos uma classe chamada **UsuarioDTO** que poderia ser utilizada em qualquer framework. Como estamos trabalhando com o Spring, nós podemos utilizar os próprios recursos que o Spring nos oferece e fazer essa configuração de uma forma mais simples. Para isso, vá até a classe UsuarioController e coloque o método:

```
@InitBinder
public void initBinder(WebDataBinder binder){
    binder.setAllowedFields("nome","email","senha","nomeImagem");
}
```

Com isso nós estamos especificando que somente esses quatro atributos é que tem permissão de serem manipulados pelo usuário protegendo assim o atributo **roles** presente na classe Usuario. Volte agora ao método **registrar** e retorne para a associação do formulário ser feita com a classe Usuario:

```
@RequestMapping(value = "/registrar", method = RequestMethod.POST)
    public String registrar(MultipartFile imagem,
            @ModelAttribute("usuarioRegistro") Usuario usuarioRegistro,

....
```

Com isso, podemos remover a linha que havíamos feito na etapa anterior:

```
Usuario usuarioRegistro = usuarioDTO.montaUsuario();
```

Tente cadastrar um novo usuário, por exemplo o Rodrigo fazendo a mesma manipulação do HTML realizadas pela Joviane e o Pedro:

```
<input type="hidden" name="roles[0].name" value="ROLE_ADMIN"/>
```

Qual é o perfil que foi registrado para o usuário Rodrigo? Ele consegue acessar a parte administrativa da Alura Shows?