

Validando arquivo passado

Transcrição

[00:00] Nós conseguimos cadastrar um novo usuário na Alura shows mas, ao invés de nós passarmos um arquivo que seria da imagem do perfil desse usuário, nós passamos o quê? Nós passamos esse arquivo aqui, que é uma jsp, e dentro dessa jsp, nós vamos ter um código JAVA, que vai ser responsável por remover o diretório aqui na nossa máquina do Windows que é onde o Tomcat está executando.

[00:22] Nós devíamos fazer o quê? Nós devemos fazer uma validação de que tipo de arquivo é esse que o nosso usuário vai estar passando para que nós possamos ter uma garantia maior de que o arquivo que ele passou é de uma imagem mesmo.

[00:34] Então nós poderíamos pensar em fazer essa análise inicialmente verificando qual é a extensão do arquivo, se é extensão de um arquivo de uma imagem válida como jpeg ou png. Mas se nós fizermos isso, o que é que pode acontecer? O usuário poderia vir aqui, vamos só voltar para o Atom, ele poderia muito bem abrir uma outra aba, copiar todo esse código que ele já fez para jsp, ele colaria aqui nessa outra aba, ele salvaria esse arquivo como se fosse um arquivo de imagem.

[01:02] Por exemplo, ele poderia vir aqui e falar que é uma imagem falsa .jpeg e nós teríamos aqui, esse nosso arquivo imagem falsa.jpeg, parecendo que é uma imagem, mas na verdade tem o quê? É o mesmo código JAVA que nós já tínhamos na nossa jsp. Nós não queremos isso, nós queremos de fato verificar se o arquivo que ele passou é um arquivo de imagem mesmo.

[01:23] Por isso que é importante nós fazermos justamente a verificação do conteúdo desse arquivo para que nós possamos ter uma certeza maior de que o arquivo que ele está passando é de uma imagem mesmo.

[01:34] Vamos voltar aqui para o eclipse para que nós possamos iniciar esse processo de validação da imagem e só depois que nós validarmos a imagem, nós seguimos com o respectivo cadastro do nosso usuário. Então como nós vimos, quando o usuário clica no botão registrar, essa informação, essa requisição vem aqui para nossa classe usuário controller, para esse nosso método aqui, que é o método registrar.

[01:59] O método registrar está recebendo como parâmetro quem? Esse arquivo que nós acreditamos que é uma imagem do tipo MultipartFile. Nós estamos passando essa nossa imagem para o método aqui tratar imagem para que nós façamos todo o tratamento dessa imagem para poder disponibilizar depois no perfil do nosso usuário.

[02:18] É importante antes fazermos esse tratamento da imagem para disponibilizar no perfil do usuário, é importante nós fazermos a validação, se essa imagem de fato, se esse arquivo de fato corresponde a uma imagem ou não. Então como nós vamos precisar fazer uma lógica aqui, vamos isolar esse nosso método para fazer a respectiva validação e tratamento da imagem numa classe própria, que nós vamos chamar de imagem validator.

[02:42] Vamos aqui no nosso pacote validator. Nós vamos criar essa classe que vai ser responsável por primeiro fazer a validação e depois o tratamento dessa imagem. Nós vamos chamar essa classe de imagem validator. Então vamos só levar esse nosso método que inicialmente estava aqui na nossa classe usuário controller para essa nossa classe que nós acabou de criar, que é a classe imagem validator.

[03:06] Então aqui, antes de nós seguirmos com esse fluxo aqui para fazer o tratamento da imagem e mostrar no perfil do usuário, nós vamos fazer essa avaliação se o arquivo passado é ou não de uma imagem. Então perceba o que nós

estamos falando, se é ou não uma imagem, então nós esperamos que o quê? Que o retorno desse nosso método seja do tipo booleano e já vamos aproveitar também e mudar aqui o modo de acesso para público.

[03:29] Agora, para nós podermos fazer essa verificação, nós vamos instancear a classe aqui `ByteArrayInputStream`. Então nós colocamos aqui “`new ByteArrayInputStream()`”, para que nós possamos passar aqui no construtor quem? Os bytes que compõem esse arquivo passado pelo nosso usuário.

[03:50] Nós vamos aqui e colocamos a imagem e chamamos o método `Getbytes`, para que nós possamos pegar os bytes que compõem esse arquivo passado para o nosso usuário para nós podermos fazer essa respectiva análise.

[04:02] Nós colocamos aqui ponto e vírgula “`Ctrl + U`”, para assinalar a variável local que nós vamos chamar aqui de `bytesImagem`. Agora que já temos a informação dos bytes que compõem esse arquivo passado pelo nosso usuário, nós precisamos fazer essa validação se esses bytes correspondem a um arquivo de imagem ou não.

[04:22] E para isso, nós vamos pedir ajuda da classe `URLConnection`. A classe `URLConnection` tem esse método estático chamado `guessContentTypeFromStream`. Qual que vai ser a ideia desse nosso método?

[04:38] Nós vamos passar para ele os bytes que compõem esse arquivo passado pelo nosso usuário e esse método `guessContentTypeFromStream` vai tentar adivinhar que arquivo é esse e vai nos indicar qual que é o tipo mime desse arquivo, se é uma imagem jpeg, se é uma imagem png.

[04:55] Caso esse método `guessContentTypeFromStream` não consiga identificar que arquivo é esse passado pelo nosso usuário, ele vai retornar um nulo, e com isso nós sabemos que o arquivo que o usuário passou não corresponde uma imagem.

[05:09] Então aqui, o que é que nós vamos passar? Nós vamos passar para esse método fazer análise dos bytes da imagem que o nosso usuário passou na hora de fazer o registro. E ele vai nos retornar qual que é o tipo mime desse arquivo, se ele conseguir identificar, se é uma imagem jpeg, se é uma imagem png, se é um arquivo de imagem válido.

[05:27] Se ele não conseguir identificar qual que é, que arquivo é esse, ele retorna o nulo. E nós sabemos que não é um arquivo de imagem válido. Então nós vamos verificar. Se “`if(mime==null)`” aqui for nulo, nós sabemos que o arquivo que o usuário passou não é válido. E nós vamos o quê? Nós retornamos aqui o falso.

[05:43] E aqui, se o mime for diferente de nulo, é porque o método `guessContentTypeFromStream` conseguiu identificar o tipo mime desse arquivo, se é uma imagem jpeg, uma imagem png, por exemplo, e nós sabemos que o arquivo que o usuário passou de fato é de uma imagem, e nós podemos seguir com todo esse fluxo normalmente como nós estávamos fazendo anteriormente. Nós só retornamos aqui o true.

[06:07] Com isso nós estamos agora fazendo a respectiva validação desse arquivo passado pelo nosso usuário e nós estamos indicando se o arquivo que ele passou corresponde ou não a uma imagem válida. E assim como nós fizemos no retrofit, nós podemos `@component` para ser um componente gerenciado por essa classe c, um componente gerenciado pelo spring.

[06:29] Nós colocamos aqui, “`Ctrl + S`” para salvar, e agora nós podemos voltar na nossa classe usuário controller e como nós colocamos a anotação `@component`, nós podemos vir aqui e pedimos para o spring fazer a injeção, colocando anotação `@Autowired`, nós colocamos aqui `private`, e nós colocamos `ImagenValidator`, e nós vamos colocar a imagem desse nosso atributo também de `ImagenValidator`.

[06:57] Então com isso, nós podemos voltar aqui nesse nosso método registrar e nós chamamos o nosso atributo `imagem validator` e dentro da nossa classe `Imagen validator`, nós temos quem? Nós temos nosso método tratar `imagem`, que vai fazer essa validação e vai nos retornar a informação `true`, se no caso o arquivo passado pelo usuário é de imagem, ou `false`, se o arquivo passado pelo usuário não é de uma imagem.

[07:23] Nós temos que fazer aqui “Ctrl + U” para pegar essa informação e nós queremos verificar o quê? Se é imagem. Então se é imagem, se essa variável for true, nós seguimos com todo esse fluxo para fazer o cadastro do nosso usuário. Se não, nós retornamos uma mensagem para ele dizendo que a imagem que ele passou não é uma imagem válida.

[07:41] Eu vou vir aqui e nós colocamos se a imagem for true, nós seguimos com todo esse fluxo para fazer o registro do nosso usuário. Se não, é porque nós sabemos que o arquivo que o nosso usuário passou não é de uma imagem válida, então nós chamamos aqui redirect, o método addFlashAttribute e na nossa jsp, já tem a chave configurada com o nome mensagem.

[08:07] Nessa chave mensagem que já estava na jsp, nós vamos passar aqui a mensagem personalizada quem nós queremos mostrar para o nosso usuário. Então poderia ser a mensagem que nós quisermos. Vamos colocar, por exemplo, que a imagem passada não é válida. E nós colocamos aqui para retornar para nossa jsp usuário.

[08:32] Então com isso, agora, nós estamos primeiro fazendo a validação e só depois é que nós vamos decidir se esse arquivo passado pelo usuário for de uma imagem, nós seguimos com esse fluxo aqui para fazer o cadastro. Se não, nós retornamos essa mensagem, que a imagem passada pelo nosso usuário não é válida.

[08:49] Vamos verificar se a nossa aplicação agora está se comportando como nós esperamos, fazendo a devida validação. Então eu vou só reiniciar o Tomcat aqui, só esperar alguns segundos para o Tomcat reiniciar.

[09:05] Vamos voltar para a máquina do hacker, no Kali Linux, e nós vamos tentar cadastrar um novo usuário, que vai ser a Vanessa. Então aqui no nosso browser, vamos acessar a aplicação da Alura shows, então nós colocamos o endereço IP da máquina do Windows, a porta de comunicação do Tomcat, 8080, e nós colocamos Alura shows para acessar essa nossa aplicação.

[09:28] Nós vamos aqui, usuário, e nós vamos aqui, registrar esse novo usuário, que vai ser a Vanessa. Então inicialmente, vamos passar aquela mesma jsp que a Priscila conseguiu se cadastrar na aplicação. Então vou colocar aqui o e-mail da Vanessa, que vai ser vanessa@gmail.com e aqui na imagem, a senha da Vanessa, ficou faltando, vamos colocar 1 2 3 4 5, e nós colocamos aqui essa imagem que inicialmente vai ser aquela jsp que a Priscila tinha conseguido se cadastrar inicialmente.

[09:58] Eu vou colocar aqui a jsp. Então esse arquivo aqui não é um arquivo de imagem válido. Então nós esperamos que a nossa aplicação esteja fazendo essa validação e não deixa o usuário ser cadastrado aqui na aplicação da Alura shows.

[10:14] Eu vou colocar aqui registrar e vamos verificar. Olha só o que aconteceu. Justamente o quê? Aquela nossa mensagem que nós havíamos configurado, que a imagem passada não é válida. Então nós conseguimos já fazer a validação deste arquivo jsp, que de fato não é um arquivo de imagem.

[10:28] Então vamos voltar aqui para a nossa aba registrar e vamos fazer o registro agora de novo da Vanessa tentando passar aquela manipulação que nós fizemos da extensão do arquivo, que nós trocamos aquela jsp para ela ser salva parecendo que é um arquivo de imagem jpeg. Então vamos verificar se a nossa aplicação também está protegida contra isso.

[10:47] Eu vou colocar a senha da Vanessa, 12345, e nós colocamos aqui uma imagem falsa .jpeg, que também tem o mesmo conteúdo, que é o código JAVA na jsp, então nós vamos vir aqui registrar, e olha lá o que aconteceu? A imagem passada não é válida.

[11:04] Agora nossa aplicação está conseguindo fazer a validação que mesmo que o usuário faça essa troca, essa manipulação da extensão, nós conseguimos fazer a devida validação e não registrar o usuário na aplicação da Alura shows.

[11:19] Só para garantir que de fato está tudo funcionando, vamos agora fazer o registro da Vanessa como imagem válida para ver se a nossa aplicação está aceitando essa imagem válida.

[11:28] Então eu vou colocar aqui o e-mail da Vanessa, vanessa@gmail.com, a senha da Vanessa é 12345, e nós vamos colocar a imagem de fato da Vanessa, que é uma imagem válida. Então nós colocamos aqui a imagem da Vanessa e nós clicamos no botão registrar.

[11:43] O que é que nós esperamos? Nós esperamos que seja uma imagem válida, então a nossa aplicação deve aceitar. Vamos só clicar aqui, registrar, e vamos ver. A nossa aplicação aceitou agora esse arquivo que de fato corresponde a um arquivo de imagem válida.