

## Iterando com element

### Transcrição

[0:00] Bom, agora falta testar a funcionalidade de fotografia.

[0:03] Que nada mais é que um (CROOD) muito simples, se a gente for ver.

[0:08] Aqui temos a parte do upload, certo?

[0:13] Que podemos carregar imagens, que tem uma integral.

[0:18] Conseguimos colocar um comentário para ela, que provavelmente vai ser para pesquisarmos.

[0:23] Então, aqui: integral. Vou digitar, aqui, integral.

[0:27] E, temos dois botões o upload e o cancel.

[0:29] Então, conseguimos cadastrar, normalmente.

[0:35] E, se pesquisarmos aqui, integral, ele vai retornar a imagem certinho.

[0:40] Clicando aqui, podemos curtir, comentar também, comentário, apertar o publish q ele comenta para nós.

[0:54] E, como excluir. Perfeito.

[0:56] Então, essa é a funcionalidade que falta para testarmos, ok?

[1:01] Voltando para o código, então. Precisamos estruturar só esse pege objects aqui.

[1:09] Porque nós temos aqui a parte de home, page objects da home e vamos ter agora a parte de page objects das fotos.

[1:19] O ideal é que possamos dividir em pastas, certinhas, certo?

[1:26] Aqui eu vou dar um new folder, dentro da pasta de page objects e vou dividir, então, com as pastas da home com os page objects jogados para dentro dela.

[1:37] Certo?

[1:40] E vou criar também uma nova pasta chamada “fotos”.

[1:44] Que é o componente que nós precisamos criar as page objects, certo?

[1:50] E essas telas aqui, essas paginas podemos fechar por que não vamos usar agora.

[1:56] Ok? Perfeito.

[1:59] Então, voltando para pagina de page objects dentro dela que temos as fotos agora.

[2:04] vamos criar a primeira page object de fotos, que no caso podemos fazer a page object de adicionar fotos.

[2:14] Então, vamos aqui em new file, vou colocar o que?

[2:18] Foto-form.po.ts, por que form?

[2:23] Por que aqui em baixo, nosso projeto (ANGULER) temos um componente photo form que ele que faz o cadastro das fotos para gente, ok?

[2:33] Então, vamos preparar a estrutura desse photo form e vamos trabalhar em cima dele.

[2:38] Perfeito.

[2:39] Import from protractor.

[2:45] Vamos importar aqui o browser.

[2:50] O elemento.

[2:52] E o by.

[2:56] Aqui começamos exportar nossa classe, export class photo form page.

[3:05] Como sempre navegar para photo form.

[3:15] Criamos a função, demos um return um browser get e a url que precisamos.

[3:22] Se a gente for ver aqui, no nosso projeto que já esta sendo executado, temos aqui a url que é hashtag user e o nome de usuário.

[3:33] Como estamos usando o usuário do Flavio por (DEFOW) aqui, por padrão, vamos colocar / hashtag / user e Flávio.

[3:44] Certo?

[3:48] O que a gente precisa fazer agora?

[3:50] Ir para tela de upload de arquivo, no caso, bem lembrado, nós não vamos usar a url do Flávio.

[4:03] Vamos usa a /p/add, que essa funcionalidade que vamos acessar, não é de listagem mas, sim de formulário. Ok.

[4:13] Indo para a tela aqui, o que podemos ver nela?

[4:18] Um input, que vamos subir a foto, um campo descrição e dois campos de upload e de cancelar, voltando para o código, precisamos do que, então?

[4:30] Carregar uma imagem, para ver se realmente ele está carregando a imagem e selecionar o botão upload

[4:40] E ver também, o comportamento dele, se ele vai voltar para tela de listagem, por que, lembrando no começo do vídeo, testamos a funcionalidade e ela voltou para a ela inicial.

[4:52] Então, vamos fazer o que?

[4:53] Carregar imagem, temos essa função.

[5:00] e como que vamos fazer isso daqui?

[5:01] Se você já mexeu com algum projeto em Angular, normalmente, para acessar alguma pasta do seu diretório que você está trabalhando utilizamos, digamos assim, um modo inativo do próprio angular se chama path.

[5:22] Perfeito?

[5:23] Aqui vamos utilizar também esse path, então, require path.

[5:30] Então, aqui ficou como?

[5:31] Declarei uma constante, conste pah, declarei P A T H = require abro parênteses abro aspas path e fecho ele.

[5:46] Perfeito?

[5:47] Vamos utilizar agora ele, nesse método carregar imagem. Como?

[5:52] Vamos dizer que o path vai chamar o método resolve e também o node, o node normalmente, quando queremos pegar o diretório que estamos executando o código, fazemos o que?

[6:08] Digitamos underline, underline, (\_\_) duas vezes e colocamos dirname, que traduzindo, diretório name.

[6:19] E qual que é a localização da imagem que queremos cadastrar, no caso aqui, vamos colocar a imagem.

[6:27] Mas, a onde está nossa imagem aqui?

[6:29] Se formos ver também no nosso projeto inteiro, vamos aqui dentro da pasta search, há uma pasta assets que tem uma pasta imagem para nos, e temos a home jpeg.

[6:42] Podemos utiliza-la para fazer cadastros, se você quiser utilizar alguma outra imagem, só vou entrar só para fazer testes, tranquilo, para ver também se ele carrega imagens mais pesadas.

[6:52] Tudo bem. Tudo tranquilo.

[6:54] Como vamos fazer isso?

[6:56] Junto com a const path podemos declarar uma constante também, const que se chama imagine.

[7:03] E ela vai conter o diretório do nosso arquivo, certo?

[7:07] Que é a onde?

[7:08] Vamos ver aqui, quantas subníveis ele vai ter que retornar, certo?

[7:12] 1, 2, 3, 4. 4 então, como eu vou retornar essas pastas?

[7:22] um ponto, dois pontos, barra retornei o nível.

[7:26] Então, vamos fazer isso 4 vezes.

[7:27] .././../././

[7:30] Perfeito.

[7:31] E começamos a colocar os nomes das pastas. Como que é?

[7:36] `Src/assets/img/` e o nome da imagem que no caso é `home` é `home.jpeg`.

[7:50] Aqui eu usei o diretório que já estava sendo utilizado pelo projeto em angular, só para que a gente tenha mais praticidade em utilizar a imagem que já possuía aqui. Ok?

[8:03] Então, aqui, voltando para o método `carregar imagem` eu tinha o `path.resolve` e o nome do diretório, o caminho do diretório que eu precisava acessar.

[8:14] E nele eu vou passar o `dir name` e também...

[8:18] Opa, apertei botão errado.

[8:20] E aqui, agora aqui, como segundo parâmetro eu vou passar o `image` a constante que eu declarei também.

[8:27] Como que ficou isso, então?

[8:28] `Path.resolve` e dentro dele eu vou passar dois parâmetro que é o `dir name` e o `image`.

[8:36] Perfeito?

[8:37] Mas como a gente vai fazer isso agora?

[8:40] Como é que eu vou passar dentro daquele `input` essa imagem?

[8:45] Simples. Vamos dar um `enter` aqui e vamos repetir o mesmo processo que a gente repetia antes, que é: elemento eu dou um `by css` e vou passar pelo `input`.

[9:00] Aqui, o que acontece?

[9:04] Vamos para parte do `photo form` vou dar um `control P`, procurar por `photo-form.html`.

[9:12] No `html` dele, a gente consegue ver o quê?

[9:22] Vou dar um `control B` só para ficar mais legível, com mais área útil para gente ver o código, e vamos vir aqui no formulário para subir a imagem.

[9:35] E temos o que aqui?

[9:37] Temos o `input`, `formcontrolname`, `file` e também tenho o tipo dele `file`, certo?

[9:45] E é ele que vamos utilizar para procurar esse `input` para o nosso banco e subir a nossa imagem.

[9:53] Então, voltando para o nosso `photo-form.pageobject`, vamos fazer o que?

[10:00] `Input`, abro esses colchetes aqui e vou dizer que o `type` dele é igual, qual que era o outro tipo dele mesmo? Voltando para o `html`, `file`

[10:15] Certo? Perfeito.

[10:18] E aqui, depois que fecharmos os parênteses no `send Keys` mandamos esse `path`.

[10:25] O que podemos fazer aqui antes, voltamos nesse `path`, para não passar ele e ficar muito extenso nossa linha, declaramos uma constante também, dizendo que é de `image`, diretório da imagem.

[10:38] Tá certo?

[10:39] E passamos para dentro desse send Keys.

[10:41] Aqui, ele já vai informar para o nosso input, oh input, eu tenho um caminho e eu preciso que você acesse ele no meu computador para passar essa imagem para você poder cadastrar. Perfeito?

[10:55] E é isso que ele vai fazer.

[10:57] O que precisamos fazer também, para cadastrar a imagem é procurar o botão de upload.

[11:03] Voltamos então para o photo – form html e vemos aqui que temos dois botões aqui em baixo na linha 54, que é qual?

[11:14] Que é o botão de upload e o botão de cancelar, esse botão aqui oh.

[11:19] Então não queremos o cancelar, vamos testar o upload.

[11:25] Voltamos para o page object, de photo-form e vamos criar um novo método chamado como?

[11:28] Pegar botão upload. Certo?

[11:36] Esse método damos um return nele.

[11:40] Que é o que?

[10:41] Ele vai procurar um elemento um elemento botão teste, certo?

[11:48] Abro parênteses, e passo aquele nome, que é do upload, perfeito?

[11:57] Com isso nós conseguimos já fazer um teste, carregar, só que vamos carregar primeiro a listagem e os page objects, para testar a funcionalidade dele como um todo.

[12:09] Certo?

[12:10] Então, mais para frente agora vamos fazer a listagem de imagens para ver se a imagem que a gente cadastrou aqui ela vai estar carregada, vai ser carregada na lista do usuário. Certo?