

Usando Jasmine

Transcrição

[0:00] Para continuar nosso projeto, eu acho interessante a gente adotar uma prática.

[0:05] O que é?

[0:06] Vamos executar agora o nosso projeto em si, vou dar um NPM start, vou deixar ele rodando para que a gente sempre analise tudo o que tem naquela página que vamos começar a criar nossos testes para verificar, por exemplo, na tela de login eu tenho o campo input de texto.

[0:29] Ele vai receber um e-mail, vai receber um usuário, e o campo de senha, que vai ser um campo do type password, que no HTML se a gente colocar o input type password, ele tem que deixar o input como se fosse, digamos assim, sem que ele seja um texto aberto.

[0:54] Precisamos fazer essas verificações, vamos fazer esses testes, mas lembrando, como a gente não tem uma documentação, vamos ficar sempre verificando o que tem na nossa aplicação.

[1:10] Vou para o nosso navegador, local host dois pontos quatro mil e duzentos, ele foi direto, e temos aqui a nossa página de login.

[1:21] Aparentemente, a gente já tem um erro aqui na nossa URL.

[1:27] Na verdade, nem é um erro, mas indo para o nosso código colocamos aqui deve ir para a home.

[1:35] Se a gente colocasse aqui, fizesse um teste para fazer uma comparação do que a gente realmente precisasse, talvez não batesse.

[1:43] Vamos fazer esse teste aqui primeiro.

[1:47] Mas antes de fazer esse teste, vamos terminar aqui de navegar todos os inputs.

[1:52] Eu tenho uma label de login, eu tenho um campo texto e username, eu tenho um password, eu tenho um nora user, não é o usuário, registre-se agora.

[2:06] Eu tenho um botão de login, tranquilo, eu tenho uma imagem, eu tenho um please login, que apertando não faz nada.

[2:15] Eu tenho também um menu.

[2:18] Esses são os componentes que a gente precisa testar, na verdade, tudo o que a gente precisa testar na page object de signin.

[2:29] Vamos ver, aqui também tem o register now, que fica abaixo de home também, por isso que a gente deixou home.

[2:37] Tem esse nome, register to embrace a new world, registre aqui, tem um e-mail, tem um nome completo, tem o nome do usuário, tem o password e tem o registro.

[2:51] E também tem um botão que vai linkar para voltar para a outra tela de login.

[3:00] Então como eu havia falado, vamos fazer primeiro esse teste aqui, que a gente não colocou nenhum parâmetro para colocar aqui na rota e aparentemente, a rota é outra.

[3:13] Vamos fazer o que, então?

[3:14] Vamos declarar esse it.

[3:16] Vamos dizer aqui it, já vou dizer aqui, deve verificar a URL, e vou chamar o método que a gente vai criar lá.

[3:32] Vou para a página de `signin.po.ts` e vou apertar enter, deixar um espaçamento legal aqui entre os métodos, verificar URL.

[3:52] E vamos utilizar o browser, só que ao invés de dar um get que nem esse camarada aqui de cima, a gente vai dar um outro método que vai ser `get current URL`.

[4:10] Ele vai retornar esse método para a gente.

[4:14] E no `signin.e2e` o que a gente vai criar?

[4:16] A gente vai dizer aqui `expect`, ele vai esperar que aquele método retorne alguma coisa para a gente, o `signin.page`, verificar URL.

[4:29] Eu vou acessar esse método e ele vai dar o retorno da URL.

[4:34] E ele tem que ser igual, esse `expect` aqui tem um `to be`, espera-se que seja string tal.

[4:48] Bom, vou voltar para a home aqui e verificar que ele tem uma terminologia assim.

[4:56] Localhost, na verdade, toda a minha URL eu tenho que jogar lá.

[5:01] Mas ele tem uma terminologia diferente do que a gente colocou no outro.

[5:05] Vou dar um `CONTROL A`, copiar essa URL, vou voltar para o meu código e colocar nesse `to be` a minha URL.

[5:15] O que ele está dizendo aqui?

[5:18] Que esse verificar URL que está no PO vai retornar o get URL.

[5:24] Esse get URL que vai me trazer exatamente essa URL que ele estiver navegando.

[5:30] Voltei para o código aqui, vou salvar.

[5:33] Vamos rodar o nosso código.

[5:35] Eu vou cancelar aqui, agora que a gente já carregou a nossa aplicação vou cancelar, vou dar um `CLS` para limpar o meu `PROMPT` de comando e vou dar um `ng e2e`.

[5:49] Vamos ver o que ele vai dizer para a gente, se realmente é aquilo que está sendo esperado.

[5:56] Está carregando, ele sempre vai carregar um novo projeto.

[6:02] Vai fazer os nossos testes, vamos ver.

[6:07] Local host e passou os dois.

[6:09] Por que será que passou?

[6:13] Está bom, porque aqui eu já testei com a URL certa.

[6:17] Correto?

[6:19] Se eu quisesse fazer assim, eu vou salvar, eu vou retirar esse daqui e vou voltar para o PROMPT de comando.

[6:29] Na verdade, eu retirei a URL que a gente colocou, salvei porque aqui o que eu estou fazendo?

[6:34] Eu vou dizer que essa string aqui que eles estão recebendo, esse to be, essa string vazia é a mesma que esse PO.

[6:42] Vamos ver se vai estar certo o que ele está fazendo, na verdade.

[6:47] Ng e2e, vamos testar ele, verificar se está tudo OK realmente.

[6:56] Ele sempre carrega a aplicação assim, então por isso que é bom sempre criar os testes depois testar tudo, mas aqui como a gente está indo passo a passo é interessante a gente ficar testando.

[7:08] E ele já deu erro.

[7:12] E ele mostra testando tela home, aí, verificar URL.

[7:15] Ele apontou um erro aqui.

[7:19] A URL tal é diferente desse daqui.

[7:23] Agora você deve estar se perguntando por que esse camarada aqui, se eu acessar a home, ou melhor, se eu não passar nada para ele, esse acessar a home, se eu não passar nada para ele, ele acessa a home de qualquer jeito?

[7:38] E aqui, quando ele vai navegar definitivamente para a tela de signin, ele não funciona se eu passar uma string vazia.

[7:47] Vou dar um CONTROL P, vou dar (ININTELIGÍVEL) em routing, vou dar em home, e ele está dizendo aqui que o paf vazio realmente não precisa acessar nada, ou seja, dizendo se eu não colocar nada no local host. 4200, se eu não colocar nada no final realmente ele vai seguir o caminho, que é a rota default dele.

[8:18] Que no caso está aqui nesse app routing.

[8:22] Ele dizendo aqui paf vazio, paf home, ele já carrega isso daqui.

[8:27] Ou seja, esse paf vazio dá um redirect para a home e essa home, pef home está linkada a esse home.module.

[8:35] Tá certo?

[8:35] Então a gente colocou aqui uma coisinha que aparentemente para acessar a home dá certo, mas para testar se ele está na home iria dar errado.

[8:47] Por isso que se colocar ele vazio vai dar errado.

[8:52] Então eu vou dar um CONTROL Z para voltar o código que a gente tinha colocado, vou salvar e esse código vai estar OK agora.

