

## Utilizando método Browser

### Transcrição

[0:00] Importado aqui a nossa página, vamos fazer o que, declarar a variável desse SigninPage para a gente acessar os métodos que a gente vai criar no nosso page object.

[0:13] Então aqui declaração normal de variável, let signin.

[0:21] Vamos utilizar o mesmo nome, só que ao invés de com as letras maiúsculas sempre, eu utilizo no estilo Camel case aqui.

[0:36] E vou tipificar ele, aqui mudou para Inglês a configuração.

[0:42] Vamos voltar para Português a configuração do meu teclado, se acontecer isso também, vem aqui no Windows 10, ele já tem um POR PTB, Português de Brasil e o EN.

[0:57] Vou dar dois pontos aqui e vou tipar ele como SigninPage, ou seja, essa variável vai ser do tipo SigninPage.

[1:09] Essa classe que a gente criou do page object.

[1:14] Pronto.

[1:16] BeforeEach, como havia falado também, antes de inicializar todos os nossos testes, todos os nossos its, ele executa.

[1:27] Então eu vou fazer, estanciar uma nova classe de SigninPage.

[1:34] Agora sim eu consigo acessar todos os métodos de SigninPageObject.ts.

[1:41] Então vamos para o SigninPage object aqui, e vamos criar então uma coisa básica utilizando a API do Protractor para acessar a página principal deles, que no caso a gente quer acessar a página de signin.

[2:00] Vou dar um enter aqui e uma coisa muito legal é a gente acessar a API do Protractor.

[2:08] Eu vou vir aqui no Google Chrome, vou vir aqui no site do Protractor, que a gente já viu antes, que é protractortest.org, e aqui nessa reference, nesse menu de referência tem um Protractor API.

[2:25] E aí tem todos os métodos que a gente consegue bater nele, e ele vai retornar um resultado para a gente, ou a gente consegue utilizar alguma outra função.

[2:36] No caso aqui a gente tem esse método, a gente tem esse endpoint que a gente pode chegar nele, que é o browser.

[2:43] Esse browser aqui que a gente vai utilizar agora vai fazer com que o próprio, digamos assim, Protractor consiga manipular o browser que você estiver utilizando.

[2:56] No caso aqui, eu tenho funções de restart, de fazer o get da URL, enviar para qual URL ele deseja chegar, enfim, ele consegue fazer bastante coisa.

[3:11] E a gente vai utilizar bem esse get aqui, que na verdade nada mais é do que a gente colocar ele e dizer browser.get, e eu quero que ele vá até a URL tal.

[3:26] Então, vamos fazer isso no nosso page object.

[3:30] Eu vou aqui, aperto enter para descer esse class SigninPage e vou fazer a importação do browser.

[3:42] Como?

[3:43] Eu dou um import.

[3:44] De novo mudou para Inglês aqui.

[3:46] Eu dou um import, abro chaves, dou um from e digito o módulo que eu quero.

[3:53] No caso, eu estou dizendo que é o módulo de Protractor.

[3:59] E dentro dessas chaves eu digo qual que é o end point que eu quero bater, que é esse endpoint aqui.

[4:07] Eu dei um Alt Tab para trocar de tela e venho aqui, vou dizer o quê?

[4:11] Que eu quero browser.

[4:14] Ele já me trouxe o browser.

[4:17] CONTROL S, sempre lembra de dar um CONTROL S para sempre salvar o seu código para caso aconteça alguma catástrofe natural você não perca seu código.

[4:29] Aqui a gente vai colocar os métodos que a gente precisa para começar a acessar a URL, começar a testar alguns inputs.

[4:38] Então eu vou criar um método aqui que vai ser, a gente quer navegar para a home, então AcessarHome.

[4:48] Criei esse método e dentro dele, está mudando toda a hora aqui o meu idioma, eu já mudo.

[4:56] E dentro dele vou abrir chaves, eu vou dar um return porque eu vou chamar lá do signin.e2e, eu dou um return desse método que ele vai retornar o que para o signine2e?

[5:09] Browser.get e vai uma URL aqui dentro.

[5:16] Nesse caso, a gente vai passar uma string vazia.

[5:20] Esse get esperava um string, mas nesse caso a gente não vai digitar nada diferente dessa documentação que eles mostram para a gente, que tem a URL específica aqui, nesse caso a gente não precisa.

[5:36] Por quê?

[5:37] Se você voltar no protractor.config.js, a base URL dele já está aqui.

[5:46] Se a gente precisar indicar alguma URL, alguma página em específico, nós vamos colocar simplesmente a terminologia dele, a página dele.

[5:56] No caso, se realmente quisermos, se colocarmos home poderia home, se colocarmos login poderia ser, mas nesse caso não precisamos.

[6:09] Como é que vamos saber realmente se precisamos ou não disso?

[6:14] Eu vou dar um CONTROL P para navegar entre os arquivos do VS Code e vou digitar routing para pegar todas as rotas que tem configuradas no projeto de ângulo.

[6:31] E vou aqui nesse app, primeiro vou no app.

[6:35] Por que eu vou no app?

[6:37] Porque dentro dele está indicando que o home.module tem uma rota, e essa rota é o quê? Essa rota é vazia.

[6:51] Então agora eu consigo acessar, porque caso alguma coisa estivesse aqui, por exemplo, path.home, então a partir dele teria que ter home.alguma coisa, se tivesse alguma coisa nesse home.module nessas rotas que a gente vai procurar de home.

[7:09] Vamos lá agora procurar por home.routing, da mesma maneira que eu naveguei entre as pastas, que é CONTROL P.

[7:16] Home.routing e vou acessá-lo.

[7:22] Vamos procurar esse componente de signin, ele já está aqui e ele me diz que a rota é realmente vazia, que a gente não precisa colocar nada nele.

[7:33] Então eu posso fechar porque agora eu já sei que a rota da home não tem nenhuma, digamos assim, palavra mais, nenhuma página mais na URL.

[7:49] Vou chegar, não fiz alteração aqui, posso não salvar e aqui eu vou salvar esse signin.po.

[7:57] Falta a gente chamar esse e2e para nós rodarmos esse teste.

[8:02] Só uma coisa que está dando erro.

[8:03] Vamos ver qual erro que ele está dando.

[8:05] Aqui ele já me mostra.

[8:07] A importação não pode ser terminada com o final TS.

[8:11] Beleza.

[8:13] Vamos corrigir aqui, pronto.

[8:15] Só tirei o TS e ele já não está dando nenhum alarme aqui, nenhum alerta para a gente.

[8:20] Vamos configurar esse it aqui, que a gente criou na outra aula.

[8:25] Você pode dar enter aqui para ficar um espaçamento, um respiro entre os métodos, e como nós vamos acessar agora a tela da home, pode deixar aqui nesse it, você deve escrever deve ir para a home.

[8:48] Um teste básico.

[8:49] E o que eu vou fazer?

[8:50] Eu vou chamar esse signin page, dei um CONTROL C, CONTROL V, mas vamos digitar aqui SigninPage.

[9:00] Se eu der o ponto, ele já mostra quais são os métodos que tem naquele page object, que no caso é acessar home.

[9:07] É o único método que a gente tem.

[9:12] Vamos ver se vai funcionar mesmo agora?

[9:14] Eu vou abrir o Prompt de comando, que eu já tinha aberto aqui, e vou dar o comando que é para nós executarmos o teste.

[9:24] Qual será que é o comando que ele faz para executar o Protractor?

[9:28] É bem simples, vamos procurar aqui na pasta principal da nossa aplicação, e dentro desse package.json ele já está indicando todos os scripts do nosso projeto.

[9:42] No caso, NPM start eu executo o projeto, o build para fazer o build do projeto para ele fazer deploy e tudo o mais.

[9:51] Para fazer teste, esse não é um teste que a gente quer. A gente quer o e2e.

[9:57] Só explicando a diferença, esse teste aqui é o teste unitário que ele executa em cada componente que foi desenvolvido, e esse e2e é o teste automatizado de integração que a gente está fazendo com o Protractor.

[10:12] Então, o que vamos rodar aqui?

[10:15] Eu posso fechar esse package.json e vou dar um alt tab, vou no meio prompt de comando, no diretório que está as nossas pastas e dou um ng e2e.

[10:31] Vamos ver se ele vai executar?

[10:34] Entendendo só o que está acontecendo aqui, ele primeiro executa o nosso projeto para que ele fique alocado na porta quatro mil e duzentos, local host quatro mil e duzentos, e depois ele vai fazer aqui o nosso teste.

[10:51] Mas aparentemente ele deu um erro aqui, qual que será o erro?

[10:55] Process exit with error, o processo terminou com erro de código cento e noventa e nove.

[11:04] Se a gente vier mais para cima, ele já deixa um pouco em vermelho, e ele já fala cannot find Chrome binary.

[11:11] Ele não acha o binário do Chrome.

[11:15] O que acontece?

[11:15] Algumas vezes quando a gente instala o nosso Protractor, ele já instala para nós algumas coisas referentes ao selenium, aquele Web Driver, só que o Web Driver às vezes pode dar incompatibilidade de versão seja com Windows ou com o próprio Chrome.

[11:35] Até mesmo com o Node.

[11:35] Por isso que a gente precisou instalar o Node numa versão específica.

[11:39] E nesse caso, para resolver esse problema eu preciso indicar para ele qual que é o diretório do executável do binário do Chrome que ele se encontra, para ele chegar até aquele diretório e executar o Chrome para a gente, para que ele possa verificar os nossos testes e executá-los.

[12:02] Como a gente vai fazer isso?

[12:04] Eu vou dar um Alt Tab, vou vir aqui no protractor.config.js e dentro desse capabilities, que eu indiquei que o browser que a gente vai utilizar vai ser o Chrome, eu vou dar uma vírgula, vou dar um enter aqui e vou dizer Chrome, vou digitar Chrome options.

[12:26] Dois pontos, abro chaves aqui.

[12:28] Aqui eu vou indicar, vou primeiro já digitar o binary, que é o erro que ele me mostrou.

[12:36] Dentro desse binary, vou passar aonde está o EXE, o executável do Chrome instalado na nossa máquina.

[12:46] No caso, para facilitar a vida, a gente aperta o Windows aqui, digita chrome.exe.

[12:58] Aqui ele já está indicando que existe esse arquivo, eu vou vir aqui em cima dele apertar o botão direito e ele já vai dizer abrir local de arquivo.

[13:09] Ele abriu aqui aonde está, eu vou vir aqui, dou um CONTROL C nesse caminho de pastas, porque tem muita pasta.

[13:17] É melhor, às vezes, copiar o caminho dela porque senão pode dar algum erro.

[13:22] Vou voltar aqui para o VS Code, abre aspas, e colar esse caminho que ele está.

[13:30] Vou dar mais uma barra aqui e vou copiar esse chrome.exe para acessar realmente aquele executável.

[13:40] Só uma coisa, vamos mudar essas barras aqui, essas barras invertidas, porque senão não vai funcionar.

[13:45] Uma coisa que facilita muito a vida também, que é um comando do VS Code, que eu seleciono essa barra aqui, aperto CONTROL D, aperto de novo, de novo, de novo e de novo.

[13:57] Ele vai funcionar como se fosse um CONTROL F específico para aquele elemento que está selecionado.

[14:04] Ele vai identificar qual que é o outro e ele vai selecionar todos os elementos que são parecidos com aquele inicial que você selecionou.

[14:10] E eu vou apertar a outra barra invertida, ao contrário.

[14:15] Porque assim ele vai funcionar.

[14:19] Aperta um CONTROL S para salvar, vou voltar para o Prompt de comando.

[14:24] Só para limpar um pouco essa tela vou digitar CLS, limpei, e vou executar novamente o ng e2e.

[14:33] Vamos ver se vai dar certo agora?

[14:36] Vamos aguardar só para ele executar o nosso projeto em si, tranquilo, subiu localhost, subiu, e abriu o nosso Chrome tranquilamente.

[14:50] Viu que legal?

[14:51] Ele já abriu e ele já disse testando tela home.

[14:56] Essa mesma tela, ou melhor, eu vou dar um Alt Tab, vir aqui em sign in spec, testando tela home.

[15:05] Ele já vem mostrando aqui.

[15:07] Todos os acertos que ele vier, por exemplo, deve ir para a home, ele vai mostrar com verdinho, ou seja, esse teste passou.

[15:15] Mas ele está dizendo que algumas coisas falharam aqui.

[15:20] E o que será que ele falhou?

[15:23] A gente pode ver algumas dessas coisas que eles falharam, mas basicamente é o que esse erro?

[15:31] É porque a gente não deletou os outros elementos que eles deixaram para a gente.

[15:37] Por isso que ele vai falar, porque a gente não tem esses componentes de app.e2e e esse PO.

[15:43] Então vou deletar ele, que esses arquivos vieram na hora da instalação e de inicialização do projeto do próprio Angular e se executarmos novamente só para mostrar, ou melhor, vou digitar ng e2e, ele vai executar o nosso projeto como ele sempre faz e vai rodar e indicar que os nossos testes foram concluídos perfeitamente, porque agora não tem nenhum resquício do app.e2e ou ponto PO.

[16:15] Ele vai executar agora nosso projeto, pronto.

[16:19] Deu, informou para a gente que tudo foi certo.