

Configurando o Protractor

Transcrição

[0:01] Agora que a gente já entendeu o projeto, vamos entender qual que é a estrutura do Protractor em si para a gente começar a trabalhar com ele.

[0:10] Bom, aqui está rodando o nosso projeto do Angular. Vou dar um Ctrl C aqui para parar ele. Ctrl C.

[0:19] Ele vai perguntar: deseja finalizar os arquivos? Eu coloco um S.

[0:24] No caso se o seu estiver em inglês, ele vai pedir e você coloca um Y, de yes.

[0:30] E aqui eu vou utilizar o VSCode para codificar os nossos projetos.

[0:35] Você fica livre para utilizar qualquer ferramenta que você quiser.

[0:40] O VSCode aqui, ele disponibiliza para a gente um tipo de um atalho aqui no CMD que independente da pasta que a gente estiver, se a gente digitar code, C O D E espaço ponto e der enter, ele vai abrir aquela página para a gente. Como ele abriu aqui, sem problema nenhum.

[1:05] Aqui, como eu executei o code na pasta do Angular, ele já mostrou tudo o que tem debaixo daquela pasta.

[1:16] No caso aqui ele tem esse e2e, o node modules, que são todos os módulos do Node que a gente instalou naquele processo de NPM i. Ele tem a pasta SRC que está todo o projeto em Angular.

[1:31] E tem as configurações, por exemplo, o gitignore, se você for utilizar o git para fazer versionamento do seu código. É muito interessante isso daqui.

[1:41] O Angular json, que são todas as configurações do Angular, o package json, package lock.

[1:47] Bom, mas o que importa para a gente mesmo é essa pasta de e2e, que se a gente abrir aqui, o Angular em si, quando você inicializa um projeto, ele já vem com algumas configurações já, digamos assim, para que a gente ganhe tempo na hora de criar os nossos testes.

[2:09] Mas essas configurações são importantes a gente saber porque às vezes dá algum errinho ou outro e a gente fica meio perdido.

[2:17] Então, vamos entender o que esse Protractor.config.js tem a nos oferecer.

[2:24] Bom, eu vou dar um Ctrl B aqui. Esse Ctrl B, se você viu, tem a hierarquia de pastas aqui. Eu só tirei elas para que a gente tenha espaço útil na tela para a gente codificar e ver o código.

[2:40] Ele já tem um exports de um módulo jasmine-spec-reporter.

[2:49] Esse módulo aqui, ele está aqui embaixo, sendo executado. Por quê? O Protractor também ele utiliza algumas coisas do Jasmine.

[2:58] Até você vai ver mais para frente quando a gente começar a codificar, tem algumas funções, tem algumas coisas que a gente vai utilizar tudo do Jasmine.

[3:06] Então, é importante também que você tenha uma pequena noção de Jasmine, que você pode ver lá no curso que a gente tem aqui no Alura, de Jasmine. Certo.

[3:17] Mas voltando para cá, esse tem esse exports, config.

[3:21] Se você já fez alguma coisa em Node, você vai se familiarizar muito isso, que ele exporta essas configurações para algum módulo. É como se fosse um exports module alguma coisa.

[3:33] Bom, algumas coisas que você precisa saber: esse specs aqui, é muito importante que você saiba disso, que ele vai dizer o quê?

[3:41] Abaixo dessa hierarquia de src. SRC é o quê? Dentro dessa pasta e2e, claro, na pasta SRC, ele vai ler todas as pastas, por isso esses dois asteriscos e qualquer nome que seja desde que tenha o sufixo, a terminologia, com e2e-spec.ts.

[4:06] No caso aqui, olha, se a gente for olhar o que ele já criou para a gente, ele já tem dois arquivos, que são o quê? Esse app.e2e-spec.ts, coincidentemente, é a mesma coisa que aqui.

[4:22] Não, não é coincidência. É o que eles fizeram mesmo, porque todos os arquivos que estiverem com essa terminologia, ele vai executar.

[4:32] E ele também tem esse PO. Esse PO se chama page object, na verdade, é uma abreviação de page object, que ele deixa encapsulado todas as funções que tem naquela página específica.

[4:44] Por exemplo, se a gente dividir em vários componentes alguma coisa que tenha em alguma página, por exemplo, nessa app page.

[4:53] Essa app page pode ter um componente de suite, outro tem o modal, o outro tem vários inputs e cada um deles pode ser um componente específico. No caso aqui eu vou deixar tudo encapsulado, tudo nessa page object para a gente conseguir acessar eles. Basicamente isso.

[5:16] Então, a gente já sabe que ele só roda esse tipo de arquivo.

[5:22] Esse capabilities também é muito importante a gente saber, que ele vai dizer o quê?

[5:26] Uma das funções mais importantes é: qual que é o navegador que a gente vai utilizar? Por quê?

[5:31] O Protractor em si, como ele utiliza um, digamos assim, uma parte do Selenium, ele vai executar automaticamente os códigos para a gente.

[5:42] Então, ele vai precisar de algum browser. A gente pode fazer teste tanto para, digamos, para plataforma mobile, que seja Android, que seja iOS, sistema iOS.

[5:57] Você consegue configurar aqui. E, claro, colocar outros parâmetros.

[6:02] Mas no caso ele já deixa só o browser name e por padrão vem o Chrome. Eu vou deixar o Chrome. Você pode colocar o Firefox. Não tem problema nenhum.

[6:12] E algumas coisinhas que a gente precisa também saber.

[6:15] Esse baseUrl, que é URL base que ele vai utilizar. Por exemplo, se eu quisesse testar algum site que não fosse o que eu estivesse rodando aqui, por exemplo, localhost 4200, eu poderia, mas aqui ele já deixou para o padrão o que o próprio Angular já utiliza.

[6:32] Ele utiliza o framework Jasmine, como eu falei, para executar algumas funcionalidades da própria biblioteca Jasmine, certo, do próprio framework Jasmine.

[6:44] No caso, mais para frente vocês vão entender isso daqui. Esse `showColors` aqui, ele vai facilitar nossa visão do que ele está sendo executado de teste ou não.

[6:58] Enfim, é isso basicamente o que tem no nosso `Protractor.config`.

[7:05] E nesse `ts` também, `tsconfig.e2e.json`.

[7:11] Esse `tsconfig` é parecido com o que tem nesse `ts`. Eu eu ia abrir esse aqui: `tsconfig` do Angular em si, né?

[7:22] Tanto que ele usa até algumas configurações que o próprio `tsconfig` do Angular utiliza. Tanto que ele faz um `extends` do próprio `tsconfig`.

[7:32] Mas aqui ele só reforça dizendo: olha, a versão do ECMAScript vai ser o ECMAScript 5. Eu poderia colocar alguma outra. O módulo `normal`, no módulo que ele vai utilizar principal como `js`.

[7:48] Certo. O diretório de saída vai ser o `out-tsc/app`.

[7:54] E os tipos que ele vai executar vai ser `jasmine`, `prawn2` e o `Node`.

[8:02] Essa é a base que o próprio Angular, na hora que você cria o projeto, ele já cria para a gente do Protractor para facilitar o nosso código de testes.