**Como criar seu primeiro teste automatizado com Selenium WebDriver?**

Bom, falando aqui de iniciante para iniciante eu vou tentar diminuir os gargalos de aprendizagem que tive durante o estudo do Selenium Webdriver em Java. Mas antes disso, vamos esclarecer algumas coisas aqui.

**O que eu consigo automatizar com o Selenium Webdriver?**

Então, em geral Testes de Interface do usuário, ou seja, você consegue simular a manipulação da página como se fosse um usuário normal.

**Tá, mas para que eu vou querer automatizar isso se eu já faço manualmente?**

Então, automatizando os principais fluxos de execução da sua aplicação, você traz uma certa tranquilidade para a sua equipe, visto que, toda vez que tiver um build do projeto, a automação vai rodar e garantir que aquilo que já funcionava antes continua funcionando e isso dá mais segurança a toda equipe. Além disso, vale lembrar que, nós como Homo Sapiens, estamos sujeitos a temperamentos, as vezes você pode não estar 100% no dia, ou simplesmente se sente tão confortável com o sistema sempre indo tudo certinho que você deixa de testar uma parte importante, bom, é ai que entra o problema, isso pode gerar certos prejuízos a equipe.

**O que é recomendado que eu automatize com o Selenium Webdriver?**

O que o usuário deveria fazer no seu sistema? Criar um questionário? Fazer uma compra? Enviar um arquivo? Enfim, depende muito de sistema para sistema, mas eu vou considerar aqui o seguinte: Abrir a página, realizar o login(ou tentar), clicar em um botão, preencher um campo e capturar um texto que são os comandos mais utilizados e a automação desse tipo de teste gira basicamente nisso, cabendo a você somente saber organizar.

**Quais conteúdo eu já devo ter uma noção para poder compreender e desenvolver com facilidade a automação?**

· Conhecimento básico na linguagem Java

· Conhecimento básico em HTML e CSS

· Ter a JDK do Java já instalada

· Um pouquinho de inglês também ajuda, mas nada fora do normal

**Preparando o ambiente**

No seu Google Chrome vá na opção Ajuda e Sobre o Google Chrome, lá vai informar a versão do seu Chrome, o meu por exemplo está na versão 84.0.4147

Sabendo isso, vá agora no site do chromedriver <https://chromedriver.chromium.org/downloads> e realize o download da versão do chromedriver igual à do seu. Irá vir um arquivo .rar, você extrai e terá um executável.

Para facilitar você vai criar uma pasta no seu diretório C: com o nome que você preferir, eu aqui irei chamar de “projetinho”, dentro dessa pasta você coloca o executável do chromedriver.

**Criando um projeto na sua IDE de escolha**

File > New > Project

Na tela que abrir você vai em Maven, lá em Project SDK precisa já estar selecionado a SDK, se não tiver você vai lá e seleciona. Caso não apareça você vai precisar instalar. Aperta em Next, em Name você põe o nome do seu projeto, em Location você localiza a pasta onde você quer que o projeto seja criado, aqui eu recomendo em C:\...Nome do seu projeto e aperte Finish.

Feito isso seu primeiro projeto será criado, provavelmente você já irá cair na tela do “pom.xml”, esse arquivo ajuda você a configurar e importar o que você vai precisar nos testes.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Dentro do pom.xml você precisará incluir as dependências do Junit e selenium-java que se encontram no link abaixo, é só procurar pelo nome da extensão que quer usar Ex.JUnit ou Selenium.

<https://mvnrepository.com/>

Obs. Versões alpha e beta são versões mais instáveis então procure sempre versões mais estáveis e que tenham mais pessoas usando como no exemplo abaixo:

Interface gráfica do usuário

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Para facilitar, irei pôr o código aqui já, mas caso tenha atualização de versões é bom vocês buscarem no link acima citado as versões mais atualizadas.

Texto

Descrição gerada automaticamente

Feito isso vai aparecer na tela um botão do maven que é o Load Maven Changes, você precisa rodar esse load par a ele carregar as dependências.

Ou caso esteja no Eclipse deverá ir em seu projeto mandar executar um maven install clicando com o botão direito sobre o mesmo.

Texto

Descrição gerada automaticamente

Carregando as alterações do Maven

Pronto, o ambiente já está configurado.

**Abrindo o navegador pela primeira vez**

Para criar a primeira classe de testes você vai em “src > test > java” crie um package chamado testes e dentro dele uma nova classe. ATENÇÃO, ESSA NOVA CLASSE DEVE TERMINAR COM O TEXTO “Test”, exemplo “primeiroTesteTest”. Isso vale também para qualquer outra classe que você vá rodar os testes, por exemplo “DashboardTest”, “LoginTest” etc. Você também deve por @Test acima do método de teste que você criou.

Agora dentro da classe você vai criar o método de teste que aqui eu nomeei como entrar no Facebok, mas você pode criar como quiser. Vou pôr o código aqui comentado explicando cada linha. Sugiro que você remova os comentários!

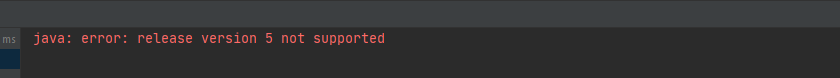
Texto

Descrição gerada automaticamente

Abrindo o navegador

Caso você aperte no Run, aquele botão verde que fica ao lado do código ele já irá executar a o teste abrindo o navegador.

Se você receber esse erro quando tente executar o teste siga os seguintes passos:



Erro da versão

File > Setting > Build, Execution, Deployment > Compiler > Java Compiler

Em “Project bytecode version” selecione o “11”. Mais abaixo em “Target bytecode version” selecione “11” também. Aplique e salve.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Corrigindo o erro1

Agora em File > Project Structure > Project

Em Project SDK selecione 11.

Texto

Descrição gerada automaticamente

Corrigindo o erro 2

Saindo de Project passe para a aba baixo “Modules” em Language level selecione o 11.

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Corrigindo o erro 3

Pronto, é só rodar novamente no Run que agora irá executar.

**Localizando elementos**

Existem 8 formas de encontrar elementos no Selenium WebDriver. Pode ser pelo nome da classe, css selector, id, name, link text, tag name e xpath. Aqui eu vou utilizar pelo id, xpath, link text e nome da classe. Caso queira saber mais sobre a localização de elementos acesse <https://www.selenium.dev/documentation/legacy/selenium_ide/#locating-elements>

Para achar os elementos na sua página você deve clicar com o botão direito em cima do elemento que deseja capturar e clica em Inspecionar. Observe as imagens:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Inspecionando os elementos

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Inspecionando os elementos

**Aqui estão alguns dos principais métodos que você irá utilizar para localizar os elementos e realizar ações durante a escrita dos scripts:**

findElement() — Você indica pelo que deve ser buscado, dentro dos () você informa os parâmetros que podem ser algum dos 8 que informei acima.

click() — Você informa para ser realizada ação de click

sendKeys() — Você informa que deve ser enviado um texto

getText() — Você informa que deve ser capturado o texto

assertEquals() — Você faz uma comparação de igualdade entre os parâmetros informados.

**Escrevendo os primeiros comandos**

Texto

Descrição gerada automaticamente

Preenchendo campo login e senha

No seu caso você só irá utilizar uma forma de encontrar o elemento, eu coloquei três para ficar um pouco mais claro. Não é em todos os casos que o elemento vai ter um Id ou um Name para encontrar, nesses casos você pode utilizar o Xpath que parece ser um pouco mais difícil, mas ajuda bastante nessas situações. Como se pode observar dentro dos parâmetros você indica ali o Id, Name ou o caminho do Xpath. Dentro do sendKeys() você indica o texto que você quer inserir ou, você pode por uma variável que já contenha o texto, aí você é quem decide.

Agora vamos encontrar o botão Entrar

Texto

Descrição gerada automaticamente

Clicando no botão entrar

Como se pode observar neste caso escrevi 4 formas diferentes de encontrar o botão Entrar. Escolha a que melhor se aplica ao seu projeto.

Para finalizar vamos utilizar o getText() e o assertEquals(). Aqui eu vou comparar um texto com a captura do erro quando você não insere nenhuma informação para login.

Texto

Descrição gerada automaticamente

Capturando erro de campos obrigatórios

Repare que, depois de tentar entrar com um login e senha errado, eu informei para clicar novamente em entrar, porque nesse caso a página de login e senha estarão em branco na segunda tentativa, assim surgir o erro.

**Extra — Executando o teste sem abrir o Navegador**

Algo que será muito utilizado mais a frente é executar o código sem abrir o navegador, isso será necessário para implementar o código em uma possível integração contínua caso queira. É muito simples, mas deu um trabalho encontrar, então vamos lá.

Texto

Descrição gerada automaticamente

Executando o projeto em Headless

Pronto, inserindo essas duas primeiras linhas de código, o navegador não irá mais abrir, sendo assim, o teste será executado em modo Headless. Caso você queira voltar para o navegador aparecer, mude de true para false.

Para aprender mais sobre o Selenium WebDriver recomendo que vocês acessem a documentação e também tem um curso ótimo do Júlio de Lima, que além de aprender a criar os scripts de automação ele ensina como organizar seu código com boas práticas, afinal, mais importante que escrever o código, é escrevê-lo bem!

Link da documentação.

<https://www.selenium.dev/documentation/>

**Curso de Júlio de lima:**

<https://www.youtube.com/watch?v=lELENTQ-PWc>