

Ferramenta Selenium Webdriver

Prof. André Takeshi Endo

Selenium Webdriver

- **Exemplo**

```
@Test
public void test02() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("https://github.com/");

    WebElement signUpButton = driver.findElement( By.xpath("/html/body/div[4]/div[1]/div/div/div[2]/div[1]/form/button") );
    signUpButton.click();

    //check msg: "There were problems creating your account."
    WebElement errorMsg = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/div[2]") );
    assertEquals("There were problems creating your account.", errorMsg.getText().trim());

    //check msg: "Login can't be blank"
    WebElement errorMsg02 = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]") );
    assertEquals("Login can't be blank", errorMsg02.getText().trim());

    //fill the username
    WebElement username = driver.findElement( By.id("user_login") );
    username.sendKeys("andreendo22");

    //click on button "create account"
    WebElement caButton = driver.findElement( By.id("signup_button") );
    caButton.click();

    try{
        errorMsg02 = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]") );
        fail();
    }
    catch(NoSuchElementException e) { }
    driver.close();
}
```

Selenium Webdriver

• Exemplo

```
@Test
public void test02() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("https://github.com/");
```

Configuração
(condições de
execução)

```
WebElement signUpButton = driver.findElement(By.xpath("/html/body/div[4]/div[1]/div/div/div[2]/div[1]/form/button"));
signUpButton.click();
```

Encontrar elementos
na página

```
//check msg: "There were problems creating your account."
WebElement errorMsg = driver.findElement(By.xpath("//*[@id=\"signup-form\"]/div[2]"));
assertEquals("There were problems creating your account.", errorMsg.getText().trim());
```

```
//check msg: "Login can't be blank"
WebElement errorMsg02 = driver.findElement(By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]"));
assertEquals("Login can't be blank", errorMsg02.getText().trim());
```

```
//fill the username
WebElement username = driver.findElement(By.id("user_login"));
username.sendKeys("andreendo22");
```

Realizar ações e
verificações

```
//click on button "create account"
WebElement caButton = driver.findElement(By.id("signup_button"));
caButton.click();
```

```
try{
    errorMsg02 = driver.findElement(By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]"));
    fail();
}
catch(NoSuchElementException e) { }
driver.close();
```

```
}
```

Selenium Webdriver

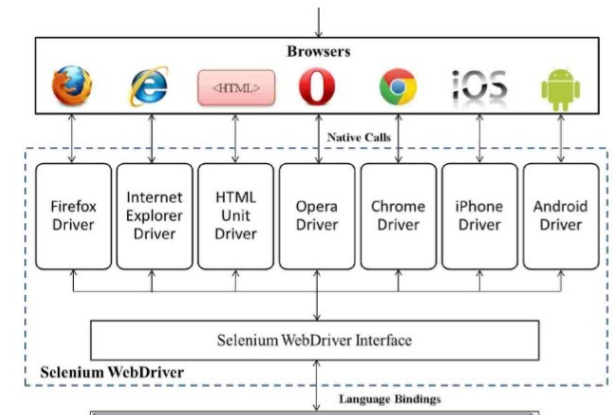
- **Exemplo**

- Veja o código atualizado em:

- <https://github.com/andreendo/software-testing-undergrad-course/tree/master/selenium/test/exemplo2>

Configuração

- Definir o driver de qual *browser*
 - Chrome driver
- Espera implícita
- `Driver.get([URL])`
 - carrega a página da [URL]
- `Driver.close()`
 - fecha o driver e a janela
- Outros métodos
 - `Driver.getTitle()` → retorna o título da página
 - `Driver.getCurrentUrl()` → recupera a url da página atual



findElement()

- `findElement([critério])`
 - Encontra o primeiro elemento que atende ao critério fornecido
 - **NoSuchElementException** → se nenhum elemento é encontrado
- `By` → especifica o critério
 - `By.id([str])`
 - `By.name([str])`
 - `By.linkText([str])`
 - `By.tagName([str])`
 - `By.xpath([str])`

By

- ***Inspeccionar a página (os componentes)***
- By.id(**[str]**) → atributo id do elemento
- By.name(**[str]**) → atributo name do elemento
- By.linkText(**[str]**) → texto do link na página
- By.tagName(**[str]**) → nome da tag do elemento
- By.xpath(**[str]**) → expressão na linguagem de consulta XPath

```
<form name="loginForm">
  <label for="username">UserName: </label> <input type="text"
    name="username" /><br/>
  <label for="password">Password: </label> <input
    type="password" name="password" /><br/>
  <input name="login" type="submit" value="Login" />
</form>
```

Classe *WebElement*

- Representa um elemento/componente HTML
- Elem.click() → efetua um clique no elemento
- Elem.submit() → envia o formulário
- Elem.getText() → retorna o texto visível
- Elem.isDisplayed() → elemento é apresentado?
- Elem.isEnabled() → elemento está habilitado?

Classe *WebElement*

- Elementos HTML

HelloH

Input

Content

textArea



checkBox



Radio button

Choice 1 ▼

Select, (combobox)

Choice 1
Choice 2
Choice 3

Select, (multiple)

Reset

button

Classe *WebElement*

- Elem.isSelected() → elemento está selecionado
 - Checkbox, opções em um select, radio buttons
- Elem.clear() → limpa os valores se for um campo de formulário
- Elem.sendKeys(**[str]**) → simula a entrada de dados no elemento

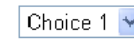
Classe *WebElement*

- **Select**

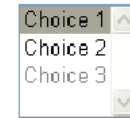
- Instanciar a classe Select

- Métodos:

- `selectBy[Index | Value | VisibleText](par1)`
- `deselectBy[Index | Value | VisibleText](par1)`
- `isMultiple()` → é possível seleção múltipla
- `getOptions()` → retorna as opções



Select, (combobox)



Select, (multiple)

```
import org.openqa.selenium.support.ui.Select;
```

```
@Test
public void test02() {
    (...)
```

```
    Select sexo = new Select( driver.findElement( By.name("sexo") ) );
    sexo.selectByVisibleText("Masc");
```

```
    (...)
```

```
}
```

<https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/Select.html>

Assertivas

- A execução sem lançamento de exceção (e.g., *NoSuchElementException*) já é uma verificação
- Usar assertivas do JUnit para verificar:
 - A presença de elementos
 - Saídas obtidas
 - Qualquer informação de saída esperada que pode ser extraída da página via método getElement()

Scrolling

- Em alguns casos, a interação não acontece porque o componente não está na tela.
- Deve-se executar alguma forma de scrolling
 - Ver <https://www.guru99.com/scroll-up-down-selenium-webdriver.html>

```
js = (JavascriptExecutor) d;
```

```
...
```

```
js.executeScript("window.scrollTo(0,200)");
```

Exercícios (i)

- Elabore 3 CTs usando Selenium e JUnit para o site:
 - <https://www.calculer.net/saude/calcular-imc-calculo-de-imc-ideal-online-aprenda-fazer-a-formula-para-calcular/>
- Faça as assertivas adequadas em JUnit
- Ver exemplo em:
 - <https://github.com/andreendo/software-testing-undergrad-course/blob/master/selenium/test/IMCTest.java>

Exercícios (ii)

- Selecione aplicações web que dispõem de formulários com os componente vistos.
- Elabore CTs usando Selenium e JUnit para esses websites.

Exercícios (iv)

- Considere a busca e busca avançada do Facebook
- Elabore CTs usando Selenium e JUnit.

Exercícios (v)

- Considere com site inicial:
 - <http://www.utfpr.edu.br/>
- Elabore um CT usando Selenium e JUnit que:
 - Navegue através links até chegar e verificar o nome do coordenador atual do curso de engenharia de software

Bibliografia

- [Gundecha15] Unmesh Gundecha, “Selenium Testing Tools Cookbook”, 2ª edição, 2015.
- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

Bibliografia

- [Myers] G. J. Myers, T. Badgett, C. Sandler, “The art of software testing”, 2012.
- [Pezze] M. Pezze, M. Young, “Teste e análise de software: Processos, princípios e técnicas”, 2008.
- [DMJ07] DELAMARO, Márcio Eduardo; MALDONADO, José Carlos; JINO, Mario. Introdução ao teste de software. Rio de Janeiro, RJ: Elsevier, 2007. 394 p. ISBN 9788535226348.
- [UUU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP, DC-UFSCAR e UTFPR-CP.

Bibliografia

- <http://startingwithseleniumwebdriver.blogspot.com.br/2013/11/here-we-try-to-configure-selenium-web.html>
- ..