**LAB 5: THỰC HÀNH SELENIUM  
 AUTOMATION VỚI BASIC TESTCASE**

**I/ Yêu cầu thực hành**

1. Học viên đã có kiến thức cơ bản về java và TestNG.
2. Học viên đã có kiến thức cơ bản về web element, selenium.
3. Thực hành viết code automation theo TestCase được cho trước.
4. Thực hành với TestNG và đọc kết quả testcase
5. Lưu lại code.

**II/ Thiết lập project.**

**2.1** Tạo Maven project.

<Tham khảo: REF\_1\_CREATE\_MAVEN\_PROJECT.docx>

**2.2**.Cài đặt TestNG

<Tham khảo: REF\_2\_Setup\_TestNG.docx>

**2.3.** Cài đặt Selenium và Chrome driver.

<Tham khảo: REF\_4\_Setup\_Selenium.docx>

**2.4**. Các hàm thông dụng trong Selenium webdriver.

<Tham khảo: REF\_5\_basic\_func\_webdriver.docx>

**III/ Ví dụ mẫu**

**1. Test case**

| **1. Khởi chạy Chrome browser.**  **2. Open link: https://accounts.lambdatest.com/register**  **3. Cài đặt quá trình tải trang không quá 15s.**  **4. Tìm kiếm và click vào “I agree to LambleTest’s...”**  **5. Tìm kiếm và click vào “Sign In”**  **6. Đóng trình duyệt** |
| --- |

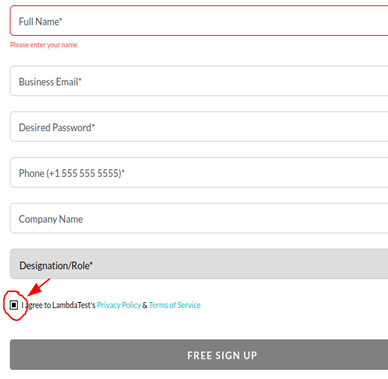
***Giải pháp:***

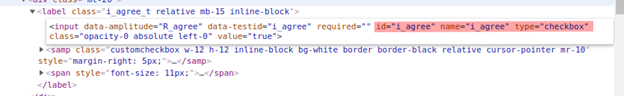
**· Cài đặt Maven project và các thư viện**

**· Truy cập vào đường dẫn để xác định vị trí các phần tử , tìm và lựa chọn các yếu tố phù hợp (id, link, class.. xpath..) để có thể định danh cho phần tử mà chúng ta cần thao tác.**

**Các thành phần này là các thành phần định danh, duy nhất và không thay đổi mỗi lần reload trang.**

**#Xác định thành phần“I agree to LambleTest’s…”**

****

****

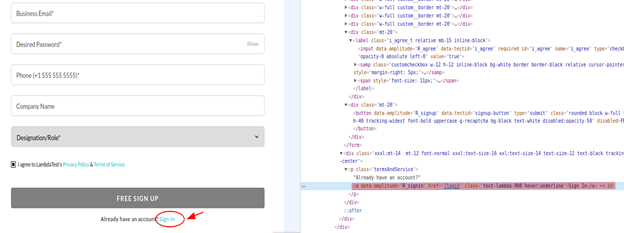
**=> Xác định được các elements có thể sử dụng đó là id, name thậm chí là type (vì cả page chỉ có mình element này có type là checkbox)**

* **Tuy nhiên thì id và name là 2 trường phù hợp nhất.**
* **Có thể lấy ra web Element dựa vào một trong 2 element này:**

| **WebElement element = driver.findElementByName("i\_agree");**  **hoặc**  **WebElement element = driver.findElementById("i\_agree");** |
| --- |

**Lưu ý: Nếu trong trang không có box “I agree to LambleTest’s…”, bạn có thể bỏ qua phần này.**

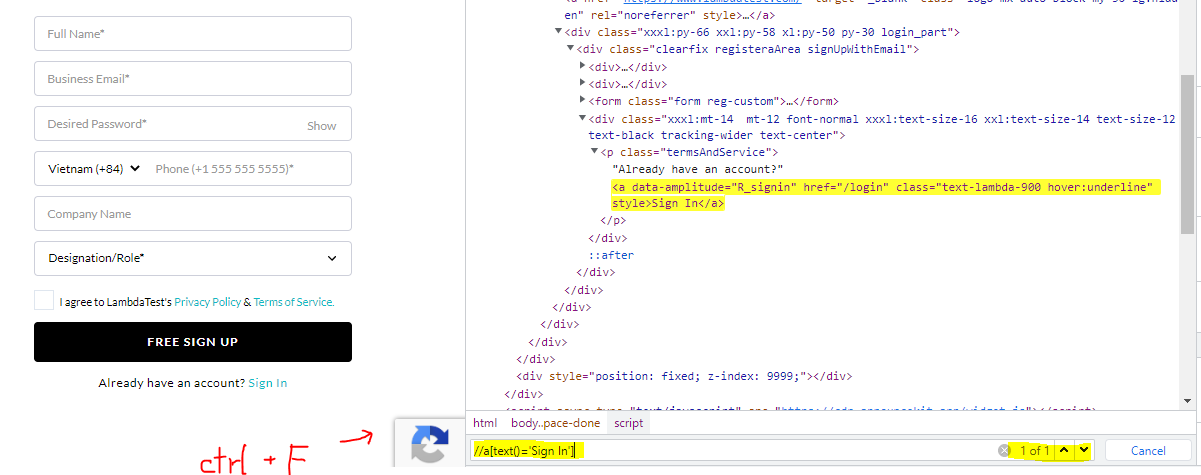
**#Xác định thành phần“Sign In”**

****

**Với “Sign In”, chúng ta không tìm thấy các element phù hợp để định danh**

**duy nhất như: Id, name ..**

* **Sử dụng XPath ?**

**  
Chúng ta sẽ thấy “sign In” được thể hiện bởi thẻ tag <a và có nội dung text = “sign In”.  
Do đó, chúng ta có thể xác định xpath dựa trên so sánh nội dung text.**

* **So sánh tuyệt đối: tìm kiếm thành phần web bắt đầu bằng thẻ tag <a và có nội dung text = “Sign In” : //a[text()='Sign In']**
* **Hoặc so sánh tương đối: tìm kiếm thành phần web bắt đầu bằng thẻ tag <a và có nội dung chứa từ khóa “Sign In” : //a[contains(text(),'Sign In')]]**

**Trong đó hàm text() : lấy ra nội dung của thành phần web, hàm contains(a, b) kiểm tra chuỗi a có chứa chuỗi b hay không.**

**Sử dụng WebDriverWait để cài đặt thời gian tải trang không quá 15s.**

**Tìm web element bằng Xpath**

| **WebDriverWait wait = new WebDriverWait(driver, 15);**  **WebElement element = wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("//a[text()='Sign In']")));**  **hoặc**  **WebElement element = wait.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*("/html/body/div[2]/div[1]/div[2]/div[2]/div/div/div[3]/p/a");** |
| --- |

* **Tạo một thao tác click chuột.**

**Sau khi lấy ra được web element, sử dụng hàm click() để tạo 1 click chuột. element.click();**

* **Để đóng trình duyệt, sử dụng : driver.close()**
* **Thực thi testcase**

**#Đáp án.**

| **import org.openqa.selenium.WebElement;**  **import org.openqa.selenium.chrome.ChromeDriver;**  **import org.testng.annotations.AfterClass;**  **import org.testng.annotations.BeforeClass;**  **import org.testng.annotations.Test;**  **public class Example1Test {**  **private String testUrl = "https://accounts.lambdatest.com/register";**  **private ChromeDriver driver;**  **@BeforeClass**  **public void init() {**  **System.*out*.println("ini: open browser");**  **System.*setProperty*("webdriver.chrome.driver","D:/chromedriver");**  **driver = new ChromeDriver();**  **}**  **@Test**  **public void TC01\_VerifyAgree() {**  **System.*out*.println("START TC01");**  **driver.get(testUrl);**  **WebDriverWait wait = new WebDriverWait(driver, 15);**  **WebElement element = wait.until(ExpectedConditions.*visibilityOfElementLocated(By.id(*"i\_agree")));**  **element.click();**  **System.*out*.println("FINISH TC01");**  **}**  **@Test**  **public void TC002\_VeirySignIn() {**  **System.*out*.println("START TC02");**  **driver.get(testUrl);**  **WebElement element = driver.findElementByXPath("//a[text()='Sign In']");**  **element.click();**  **System.*out*.println("FINISH TC02");**  **}**  **@AfterClass**  **public void terminate() {**  **System.*out*.println("finish: close browser");**  **driver.close();**  **}**  **}** |
| --- |

**III. Thực hành**

**1.** Yêu cầu viết chương trình tự động hóa quá trình sau.

* Gọi trình duyệt Google Chrome.
* Mở URL: *https://www.google.com.vn/*
* Cài đặt thời gian tải trang không vượt quá 15s.
* Nhấp vào hộp văn bản Tìm kiếm của Google.
* Nhập giá trị “**selenium**”
* Bấm vào nút Tìm kiếm.(“Tìm trên google”)
* Đóng trình duyệt.

*Gợi ý:*

1. Tạo Maven project, testNG và selenium
2. Xác định Element locate/ Xpath của Ô văn bản tìm kiếm & Nút Tìm kiếm/Tìm trên google.



Có thể là id, name, Css class, xpath

1. Viết Class **Lab5Example.java**

* @BeforeClass  
   - Thiết lập setProperty đến nơi lưu file chromeDriver.exe.

- Khởi tạo đối tượng ChromeDriver

* Viết hàm TC01\_GoogleSearch() và đánh @Test

- Mở trình duyệt và điều hướng đến : [*https://www.google.com.vn/*](https://www.google.com.vn/)

- Sử dụng WebDriverWait để cài đặt thời gian tải trang không quá 15s.

Sau khi trang được tải thành công thì lấy ra ô nhập văn bản tìm kiếm thông qua WebDriverWait

| WebDriverWait wait = **new** WebDriverWait(**driver**, 15);  WebElement **inputSearch** = wait.until(ExpectedConditions.visibilityOfElementLocated(By….**"**)));  //By.id()  //By.name()  //By.xpath() |
| --- |

- Thực hiện đẩy từ khóa “selenium” vào ô tìm kiếm.

Sử dụng câu lệnh sau: **inputSearch.sendKeys("selenium");**

Trong đó **inputSearch** là WebElement vừa tìm được.

- Tiếp tục tìm kiếm (id, name, css, xpath…) tương ứng với Nút Tìm kiếm/ Tìm trên google.

| WebElement **btnSearch** = **driver.findElementByXPath(…**)  WebElement **btnSearch** = **driver.findElementById(…**)  WebElement **btnSearch** = **driver**.findElementByName(...) |
| --- |

Sau khi lấy ra được WebElement, sử dụng hàm click() để tạo ra 1 click chuột. **btnSearch.**click**();**

@AfterClass

Thực hiện đóng trình duyệt thông qua hàm: **driver.close()**