SWE 4604

Software Testing and Quality Assurance Lab

Lab 6

Prepared By Maliha Noushin Raida, Lecturer, CSE
Islamic University of Technology

Selenium: Introduction

- Selenium is a of testing automation tools for web-based applications: **Selenium IDE, Selenium WebDriver** etc.
- The operations provided by this tool are very much flexible and afford many options for comparing UI element for expected behavior.
- Supportive of:
  - Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript
  - Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.
  - Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
- Easy to use
- Open Source Tool

Selenium IDE

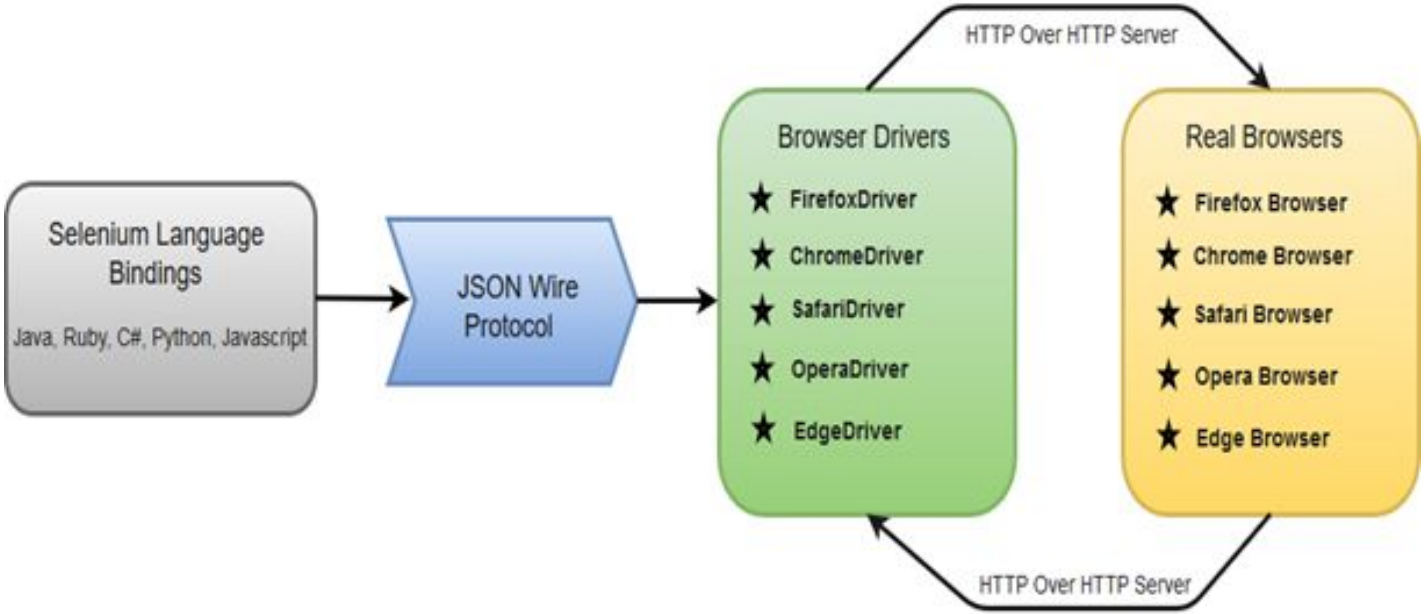
- Rapid prototyping for building test scripts
- Plugins for many web browsers
- Can be used by testers with no programming experience to write simple test scripts.
- Can record user activities and add is to the test case
- Provides CLI within the plugin
- Export tests to code
- Link: Selenium IDE · Open source record and playback test automation for the web

Selenium Webdriver

- The browser is controlled directly from OS (Operating System) level. The basic requirements to run a test script on WebDriver are:
  - An IDE (Integrated Development Environment) with any of the supported programming language like Java, C#, etc.
  - A Browser to execute the instructions generated by the test script.

- Selenium WebDriver performs much faster
- Selenium WebDriver API provides communication facility between languages and browsers.

Selenium Webdriver Architecture



Selenium Webdriver

ChromeDriver:
https://chromedriver.chromium.org/
Code Example:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

System.setProperty("webdriver.chrome.driver", <location of the extracted driver folder>);
WebDriver driver=new ChromeDriver();
```

Selenium Webdriver

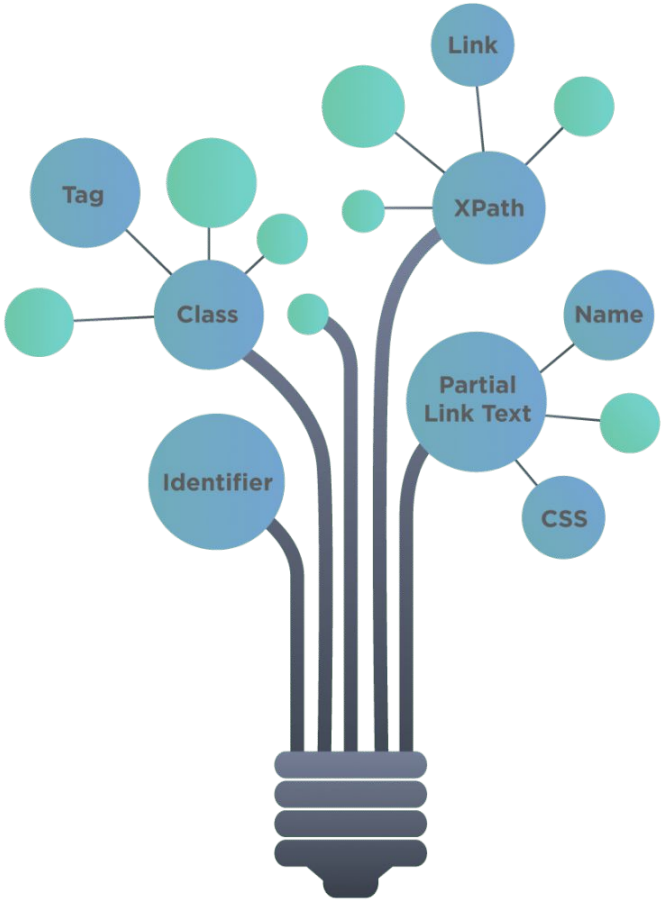
FirefoxDriver:
https://github.com/mozilla/geckodriver/releases
Code Example:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;

System.setProperty("webdriver.gecko.driver",Path_of_Firefox_Driver");
DesiredCapabilities capabilities = DesiredCapabilities.firefox();
capabilities.setCapability("marionette",true);
Webdriver driver= new FirefoxDriver(capabilities);
```

Selenium Webdriver Locators

- **ClassName** – A ClassName operator uses a class attribute to identify an object.
- **cssSelector** – CSS is used to create style rules for pages and can be used to identify any web element.
- **Id** – Similar to class, we can also identify elements by using the 'id' attribute.
- **linkText** – Text used in hyperlinks can also locate element
- **name** – Name attribute can also identify an element
- **partialLinkText** – Part of the text in the link can also identify an element
- **tagName** – We can also use a tag to locate elements
- **xpath** – Xpath is the language used to query the XML document. The same can uniquely identify the web element on any page.



Selenium Webdriver Command(Mostly Used)

#get() method:

- `get()`
- `getTitle()`
- `getCurrentUrl()`
- `getPageSource()`
- `getAttribute()`
- `getText()`

#Close method:

- `close()`
- `quit()`

#link method:

- `click()`

#Selecting drop dropdown:

- `selectByValue("greenvalue"); or deselectByValue(“greenvalue”)`
- `selectByVisibleText("Red") or deselectByVisibleText(“Red”)`
- `selectByIndex(2) or deselectByIndex(2)`

Selenium Webdriver Command(Mostly Used)

#Navigate method

- `driver.navigate().to(<url string>);`
- `driver.navigate().back();`
- `driver.navigate().forward();`

#input method

- `sendkeys(<string to enter>)`

#Wait methods:

- `driver.manage().timeouts().pageLoadTimeout(500, SECONDS);` //after get()
- While searching for an element:

```
driver.manage().timeouts().implicitlyWait(1000, TimeUnit.SECONDS);
```

- While waiting for an element to be visible:

```
WebDriverWait wait = new WebDriverWait(driver, 10);
WebElement element = wait.until(ExpectedConditions.
visibilityOfElementLocated (By.xpath("//input[@id='name']")));
```