Selenium Notes

1. *What is Selenium ?*

Selenium is an open source web automation tool.

Limitation of Selenium :

- It doesn't support windows based application directly. However, third party tool **(eg: AutoIt)** can be integrated with selenium to automate windows based applications.

Note :

1. Selenium community developed specific tool called **WINIUM** to automate windows based applications.
2. Selenium community also developed tools to test mobile applications,
    - **Selendroid** - it supports only Android platform
    - **Appium** - it supports Android platform, MAC, Windows etc.

Note :

All the selenium related resources and documents can be found on the below website.

http://www.seleniumhq.org

Here, hq stands for head quarter.

---

2. *Why Selenium is so popular and demanding ?*

Selenium is popular and demanding due to the following features.

1. it is an open source tool freely available on internet
2. No project cost involved
3. No licence required
4. Can be easily customized to integrate with other Test Management tools like ALM, Bugzilla etc.
5. It supports almost 13 different software languages
    - Java
    - C#
    - Ruby
    - Python
    - Perl
    - Php
    - Javascript
    - Javascript (Node JS)
    - Haskell
    - R
    - Dart
    - TCL
    - Objective - C

6. It supports almost all the browsers.(Firefox, Chrome, Internet Explorer etc) and hence, cross browser testing/compatibility testing can be performed using selenium.
7. It supports almost all the Operating System (MAC, Windows, LINUX etc) and hence, cross platform testing can also be performed.

---

3. What are the different flavours of Selenium ?

- **Selenium Core** (Developed by a company called **Thought Works** way back in 2004)
- **Selenium IDE** (supports only Mozilla Firefox - supports record and playback feature)
- **Selenium RC** (Remote Control - Version is 1.x) (Used for parallel execution of automation scripts on multiple remote systems)
- **Selenium WebDriver** (Version is 2.x and 3.x)

Note :

Selenium WebDriver version 3.x is no longer capable of running Selenium RC directly, rather it does through emulation and via an interface called WebDriverBackedSelenium.

But, **it does support Selenium Grid directly.**

Selenium Grid :

1. It is one of the component of selenium that is used to run automation scripts on multiple system simultaneously.
2. It is used to carry out compatibility testing on multiple browsers and platforms.

---

4. What are the key/Important topics of Selenium ?

- **Automation Framework -** guidelines and rules to write selenium code
- **GitHub** - Central Repository to store code
- **Maven** - build dependency tool for auto update of selenium version
- **Selenium Grid** - to test on multiple OS and browsers
- **Jenkins** - Continuous Integration
- T**estNG** - framework for generation of Test Reports and running multiple test scripts in one go

---

5. What are the Softwares required for Selenium ?

1. **Eclipse IDE** - Oxygen (Stable version)
2. **JDK 1.8**
3. **Selenium Server-Standalone**-3.7.1 (Stable version)

   *(Download it from the given url : **http://www.seleniumhq.org/download**)*

4. Driver Executables

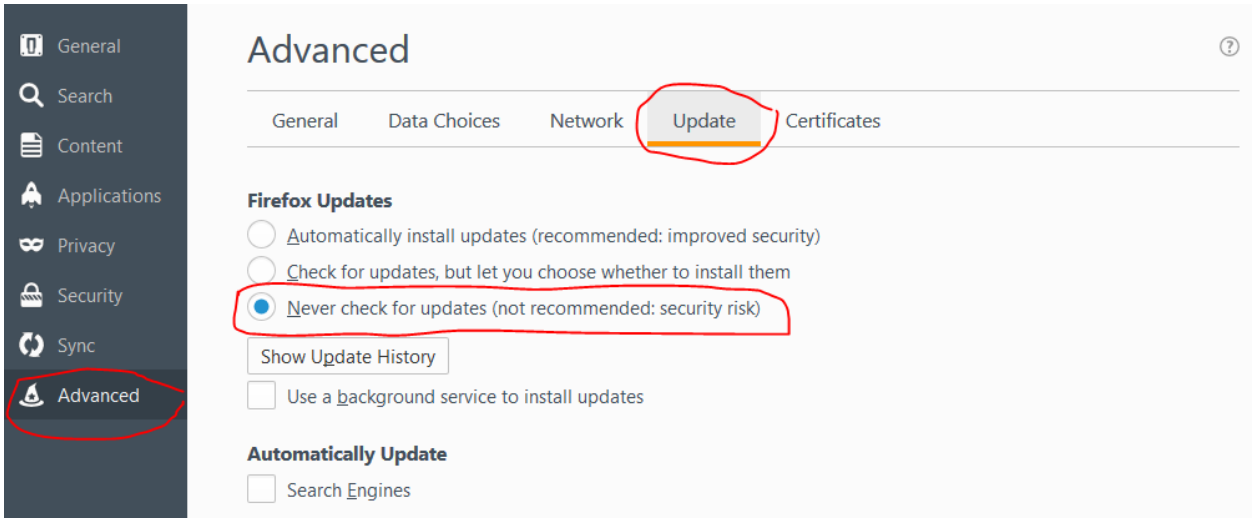
- For Firefox Browser
  - the name of the driver executable is : **geckodriver.exe**
  - Url to download : https://github.com/mozilla/geckodriver/releases
  - Version **0.19** is recommended for firefox browser with version 56.0 (selenium jar - 3.7.1)
- For Chrome browser
  - the name of the driver executable is : **chromedriver.exe**
  - Url to download : https://chromedriver.storage.googleapis.com/index.html?path=2.31/
  - Stable version of chrome version is 62.0 (Use chromedriver.exe with version 2.33)

5. Browsers:

Firefox (Version 57.0)

Chrome (Version 62.0)

Note : To stop auto update of firefox browser version, Make sure to disconnect the internet connection and then install 54.0 version, now go to Setting/Option in firefox browser and check the below checkbox - Never check for updates.



6. Application Under Test (AUT)

Application Name : actiTIME

Online url : https://demo.actitime.com/login.do

Offline url : https://localhost:8080/login.do

To download actiTIME application ,

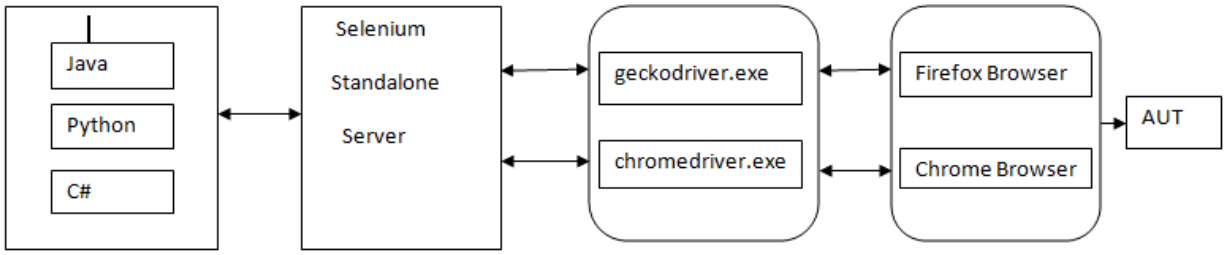
https://www.actitime.com/download.php

---

6. Selenium Architecture - High Level ?

OR

How selenium performs automation testing on browser ?



Client Binding **Driver Executables** **Browser**

OR **.jar file** **uses**

Language Binding **JSON Wire Protocol**

1. **Since selenium supports multiple languages such as Java, Python, C# etc, we can develop automation scripts in all the supported languages. This is know as language binding or Client binding.**
2. **When we execute the selenium code, request goes to the Selenium Standalone Server (also known as Selenium WebDriver Server), which further process the request based on the input received from the client binding and perform specific actions on the respective browsers using the browser specific driver executables,**

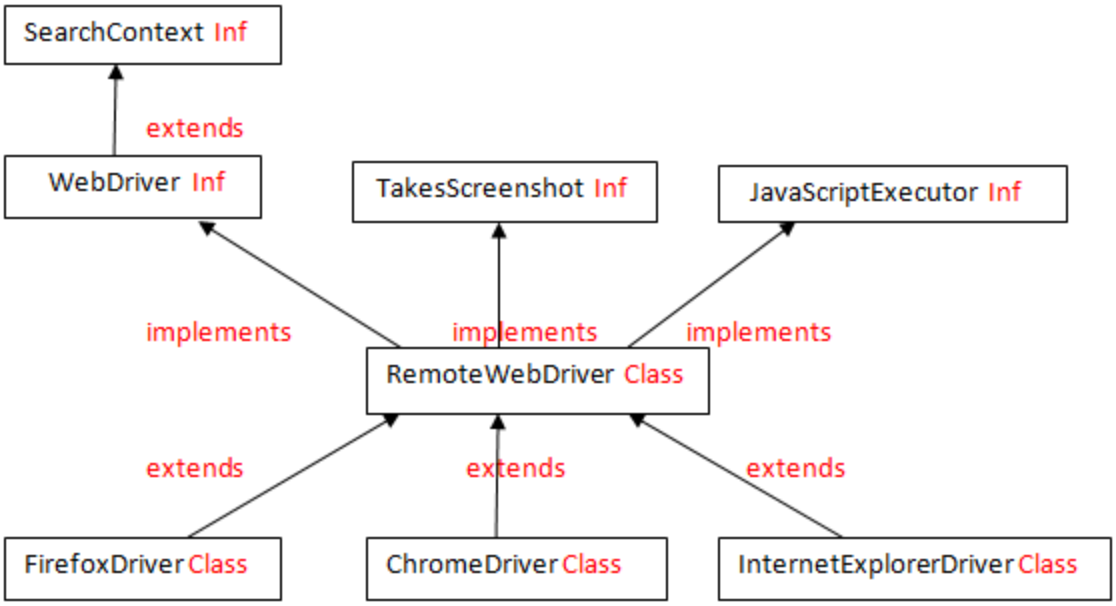
   Eg : geckodriver.exe for firefox browser and

   chromedriver.exe for chrome browser and so on…

3. **Driver executables uses a protocol called JSON Wire protocol to communicate with related browsers. (JSON stands for Java Script Object Notation)**

---

7. Selenium Java Architecture - Detailed Level



1. **SearchContext is the supermost interface present in selenium webdriver.**
2. **An interface called WebDriver extends SearchContext interface.**
3. **A total of 13 interfaces are available in selenium, which is implemented by a super most class called RemoteWebDriver**
4. **RemoteWebDriver is again extended by few browser specific child classes such as,**
    - **FirefoxDriver class to automate on firefox browser.**
    - **ChromeDriver class to automate on Chrome browser,**
    - **InternetExplorerDriver class to automate on IE and so on…….**

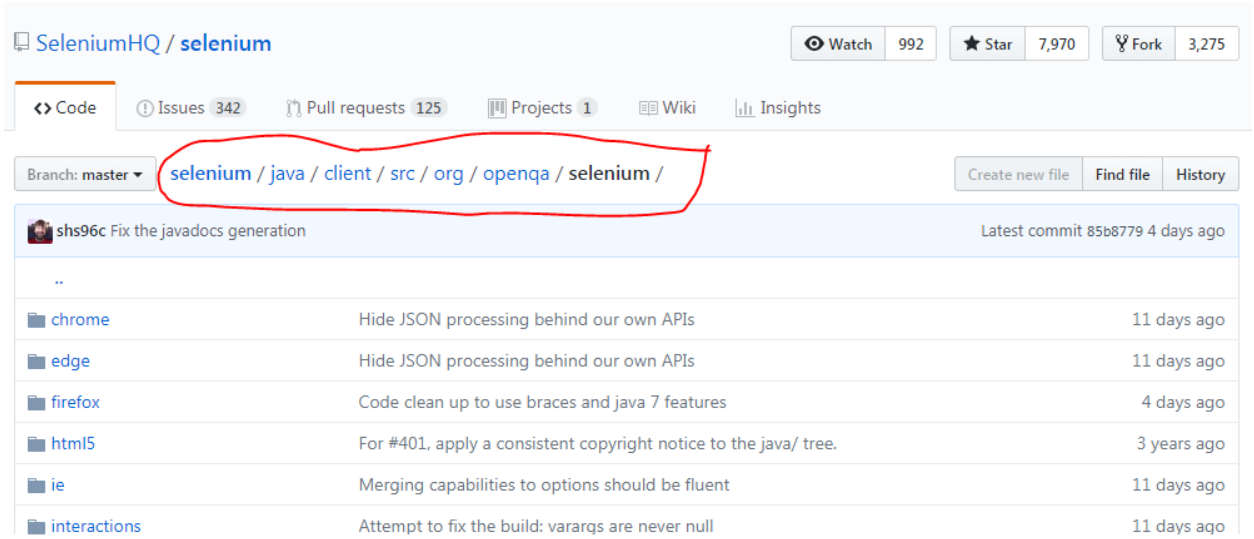
NOTE :

All the above mentioned **interfaces and classes** are present in a package called **"org.openqa.selenium".**

To view any information about Selenium interfaces, classes and methods, navigate to the below page.

https://github.com/SeleniumHQ/selenium/tree/master/java/client/src/org/openqa/selenium

Highlighted below in red is the navigation path.



---

8. *List down all the methods present in below interfaces of Selenium WebDriver.*

Methods of SearchContext interface :

1. **findElement()**
2. **findElements()**

Methods of WebDriver interface :

1. **close()**
2. **get()**
3. **getTitle()**
4. **getPageSource()**
5. **getCurrentUrl()**
6. **getWindowHandle()**
7. **getWindowHandles()**
8. **manage()**
9. **navigate()**
10. **quit()**
11. **switchTo()**

Methods of TakesScreenshot interface :

1. **getScreenshotAs(args)**

Methods of JavascriptExecutor interface :

1. **executeScript()**
2. **executeAsyncScript() - we don't use this for automation**

Methods of WebElement interface :

1. **clear()**
2. **click()**
3. **getAttribute()**
4. **getCssValue()**
5. **getLocation()**
6. **getRect()**
7. **getSize()**
8. **getTagName()**
9. **getText()**
10. **isDisplayed()**
11. **isEnabled()**
12. **isSelected()**
13. **sendKeys()**
14. **submit()**

9. *Why we upcast the browser related child class to WebDriver, and not RemoteWebDriver class (RemoteWebDriver being the super most class in selenium) ?*

Upcasting Example :

WebDriver driver = new FirefoxDriver();

- Converting a child class object to super type is called Upcasting.
- In selenium, we use upcasting so that we can execute the same script on any browser.
- In selenium, we can upcast browser object to RemoteWebDriver, WebDriver, TakesScreenshot , JavascriptExecutor etc, but a standard practice is to upcast to WebDriver interface.
- This is as per the Selenium coding standard set by the Selenium community. As a testimonial, navigate to the below selenium community site and check for the text as mentioned in the image below.

Url - http://www.seleniumhq.org/projects/webdriver/

WebDriver is the name of the key interface against which tests should be written in Java, the implementing classes one should use are listed as below:

ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver, InternetExplorerDriver, PhantomJSDriver, RemoteWebDriver, SafariDriver

---

10. Where did you use Upcasting in Selenium ?

WebDriver driver = new FirefoxDriver();

Explain the above statement..

1. WebDriver is an interface in Selenium that extends the supermost interface called SearchContext.
2. driver is the upcasted object or WebDriver interface reference variable.
3. " = " is an assignment operator.
4. new is a keyword using which object of the FirefoxDriver class is created.
5. FirefoxDriver() is the constructor of FirefoxDriver class which initialises the object and it will also launch the firefox browser.

---

11. Steps to install/integrate selenium server to the java project

1. Launch eclipse and go to package explorer [navigation path :- Window menu → Show View → Package Explorer]
2. Create a java project [File → New→ Java Project]
3. Right click on Java Project and add a new folder with name "driver" [File → New→ Folder]
4. copy **geckodriver.exe** file from your system and paste it into this driver folder
5. Similarly, create another folder with name **"jar"**and copy **Selenium Standalone Server.jar** file into this jar folder.
6. Expand the jar folder and right click on **Selenium Standalone Server.jar** file → select **Build Path** → select **Add to Build Path**
7. As soon as you add any .jar files to build path, a new folder will be available called **"Reference Libraries"** under the package explorer section and you can see the .jar file is added to this **"Reference Libraries"**
8. To remove the .jar file from the java build path, go to the Reference Libraries → select the .jar file → right click → select build path → Remove from build path.
9. Other way of adding .jar file to java build path is : right click on the project → build path → configure build path → Libraries tab → Add External jars → select the .jar file → Apply → ok

12. *This program demonstrates Upcasting concept (FirefoxDriver class object to WebDriver interface) and accessing various methods of WebDriver interface*

```
package qspiders;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;


public class UpcastingToWebDriver_LaunchBrowser {

   public static void main(String[] args) throws InterruptedException  {

            //setting the path of the gecko driver executable

            System.setProperty("webdriver.gecko.driver", ".\\driver\\geckodriver.exe");

            //Launch the firefox browser

            WebDriver driver = new FirefoxDriver();

            //Enter the url

            driver.get("http://www.google.com");

            //Get the title of the google page and print it on the console

            String title = driver.getTitle();

            System.out.println("the title of the page is :"+ title);

            //Get the URL of the google page and print it on the console

            String currentUrl = driver.getCurrentUrl();

            System.out.println("the URL of the page is :"+ currentUrl);

            //Get the source code  of the google page and print it on the console

            String pageSource = driver.getPageSource();

            System.out.println("the source code  of the page is :"+ pageSource);

            //Halt the program execution for 2 seconds

            Thread.sleep(2000);

            // Close the browser

            driver.close();

             }
```