**Selenium WebDriver**

Selenium WebDriver is one of the most popular tools for Web UI Automation. And no better than Python can complement it to automate a broad range of web applications.

Web UI Automation means the automatic execution of the actions performed in a web browser window like navigating to a website, filling forms that include dealing with text boxes, radio buttons and drop downs, submitting the forms, browsing through web pages, handling pop-ups and so on. Selenium WebDriver is the one that can automate all these tasks. It can interact with all types of Web browsers available till date like Firefox, Internet Explorer, Safari, and Chrome, etc.

**Selenium Setup**

Steps:

* Open command prompt (cmd)
* Cd <python installation path script folder>
* Run below command

For new installation:

pip install selenium

For update:

pip install -U selenium

**Write your first selenium web driver automation program:**

There are 2 ways to run Selenium python tests in Google Chrome. I'm considering Windows (Windows 10 in my case):

**Prerequisite:** Download the latest Chrome Driver from: https://sites.google.com/a/chromium.org/chromedriver/downloads

***Way 1:***

i) Extract the downloaded zip file in a directory/location of your choice  
ii) Set the executable path in your code as below:

self.driver = webdriver.Chrome(executable\_path='D:\Selenium\_RiponAlWasim\Drivers\chromedriver\_win32\chromedriver.exe')

***Way 2:***

i) Simply paste the chromedriver.exe under /Python/Scripts/ (In my case the folder was: C:\Python36\Scripts)  
ii) Now write the simple code as below:

self.driver = webdriver.Chrome()

**Example:**

We will show you a sample script that opens “www.google.com” enters a search text in the Google search text box. Test script then verifies the Google search page on which has the searched text displayed.

#### Selenium Webdriver Python Script For Firefox

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# create a new Firefox session

driver = webdriver.Firefox()

driver.implicitly\_wait(30)

driver.maximize\_window()

# Navigate to the application home page

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

# get the search textbox

search\_field = driver.find\_element\_by\_id("lst-ib")

search\_field.clear()

# enter search keyword and submit

search\_field.send\_keys("Selenium WebDriver Interview questions")

search\_field.submit()

# get the list of elements which are displayed after the search

# currently on result page using find\_elements\_by\_class\_name method

lists= driver.find\_elements\_by\_class\_name("\_Rm")

# get the number of elements found

print ("Found " + str(len(lists)) + " searches:")

# iterate through each element and print the text that is

# name of the search

i=0

for listitem in lists:

print (listitem.get\_attribute("innerHTML"))

i=i+1

if(i>10):

break

# close the browser window

driver.quit()

**How to Create Test Scripts in Selenium for Chrome**

import os

from selenium import webdriver

chrome\_path="/home/rahul/Documents/SeleniumPy/chromedriver"

driver=webdriver.Chrome(chrome\_path)

driver.maximize\_window()

driver.implicitly\_wait(30)

driver.get('http://www.google.com')

driver.quit()

**EXAMPLE 1**

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

user = ""

pwd = ""

driver = webdriver.Firefox()

driver.get("http://www.facebook.com")

assert "Facebook" in driver.title

elem = driver.find\_element\_by\_id("email")

elem.send\_keys(user)

elem = driver.find\_element\_by\_id("pass")

elem.send\_keys(pwd)

elem.send\_keys(Keys.RETURN)

driver.close()

**Explanation of the code**

* **Code line 1**: From selenium module import webdriver
* **Code line 2**: From selenium module import Keys
* **Code line 3**: User is a blank variable which will be we used to store values of username.
* **Code line 4**: pwd is also a blank (here it is empty, but the user can provide values in it) variable. This will be used to store values of the password.
* **Code line 5**: In this line, we are initializing "FireFox" by making an object of it.
* **Code line 6**: The "driver.get method" will explore to a page given by the URL.WebDriver will hold up until the page has completely been loaded (that is, the "onload" occasion has let go), before returning control to your test or script.
* **Code line 7**: "Asserts" keyword is used to verify the conditions. In this line, we are confirming whether the title is correct or not. For that, we will compare the title with the string which is given.
* **Code line 8**: In this line, we are finding the element of the textbox where the "email" has to be written.
* **Code line 9**: Now we are sending the values to the email section
* **Code line 10**: Same for the password
* **Code line 11**: Sending values to the password section
* **Code line 12**: Elem.send\_keys(Keys.RETURN) is used to press enter after the values are inserted
* **Code line 13**: Close

**EXAMPLE**

In this example,

* We will open a login page.
* Fill the required field"username" and "password".
* Then validate if the login was successful or not.

from selenium import webdriver

from selenium.common.exceptions import TimeoutException

browser = webdriver.Firefox()

browser.get("http://www.facebook.com")

username = browser.find\_element\_by\_id("email")

password = browser.find\_element\_by\_id("password")

submit = browser.find\_element\_by\_id("submit")

username.send\_keys("me")

password.send\_keys("mykewlpass")

submit.click()

wait = WebDriverWait( browser, 5 )

try:

page\_loaded = wait.until\_not(

lambda browser: browser.current\_url == login\_page

)

except TimeoutException:

self.fail( "Loading timeout expired" )

self.assertEqual(

browser.current\_url,

correct\_page,

msg = "Successful Login"

)

**Explanation of the code:**

* **Code line 1-2**: Import selenium package
* **Code line 4**: Initialize Firefox by creating an object
* **Code line 5**: Get login page (Facebook)
* **Code line 7-9**: Fetch username, password input boxes and submit button.
* **Code line 11-12**: Input text in username and password input boxes
* **Code line 15**: Click on the "Submit" button
* **Code line 18**: Create wait object with a timeout of 5 sec.
* **Code line 20 -30:** Test that login was successful by checking if the URL in the browser changed. Assert that the URL is now the correct post-login page