ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

A Literature Study on Selenium Webdriver 2.0

Ms.R. Anitha
Assistant Professor, Valliammai Engineering College,Anna University

***Abstract* - Selenium is a open source automated testing suite for web applications across different web browsers and platforms. It is similar to HP Quick Test Pro but the Selenium focuses on automating web-based applications. The testing is done by using selenium is called selenium testing. This paper studies on selenium Webdriver 2.0 which uses mozilla firefox browser and how it is implemented by using Eclipse IDE. WebDriver calls the Web browser directly and all the test script is executed similarly to make the automation.**

***Index Terms:*— Software Automation Testing; Selenium Webdiver2.0 ;Automation Framework;**

I. INTRODUCTION

Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Executing the test cases manually without any tool support is known as manual testing. Taking tool support and executing the test cases by using automation tool is known as automation testing. Automation testing will be useful to execute the set of test cases frequently. Automation runs test cases significantly faster than human resources. Selenium is a browser automation tool, commonly used for writing end-to-end tests of web applications. Selenium is a suite of three tools.

The first of these tools, Selenium IDE, is an extension for Firefox that allows users to record and playback tests. The record/playback paradigm can be limiting and isn't suitable for many users, so the second tool in the suite, Selenium WebDriver, provides APIs in a variety of languages to allow for more control and the application of standard software development practices. The final tool, Selenium Grid, makes it possible to use the Selenium APIs to control browser instances distributed over a grid of machines, allowing more tests to run in parallel. Selenium WebDriver supports the following programming languages:Java,C#,PHP,Pearl,Ruby and Python.In this paper Selenium-Java is used in Eclipse IDE environment.

II. PROPOSED WORK

Installing Selenium-Prerequisites
Step 1: Install Java on your computer.
Step 2: Install Eclipses IDE
Step 3: Download the Selenium Java Client Driver
Step 4: Create a New project in Eclipse

(Implementation of Selenium – java Scripts)

To run the web-based testing Mozilla firefox browser is used. The geckodriver for Mozilla firefox browser supports the web based testing.

1.Installing Java On the Computer

Download and install the latest version of Java Software Development Kit (JDK).

2.Configuring Eclipse IDE

Download and install "Eclipse IDE for Java Developers". Make sure to choose correct link for downloading eclipses which corresponds to the OS i.e. for Windows 32 Bit and 64 Bit version. You should be able to download a ZIP file named "eclipse-java-kepler-SR1-win32-x86_64.zip".Extract it in C Drive.

3. Download the Selenium Java Client Driver

Download the Selenium Java Client Driver from http://docs.seleniumhq.org/download/
After Downloading jar files from the above link save it in particular location.

4. Create a new Java Project in Eclipse

Go to C drive and open eclipse folder and launch eclipse.exe Select workspace and create a new java project. Go to libraries tab and add all the external jar files from the Selenium-java client driver which we have downloaded in step 3.Our version of selenium java client driver is selenium-java-3.0.1.(latest version).
The library settings after adding of jar files will be as follows:
Adding of Jar files will take the following steps and include all the Jar files in the Selenium-java-3.0.1(latest version)

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

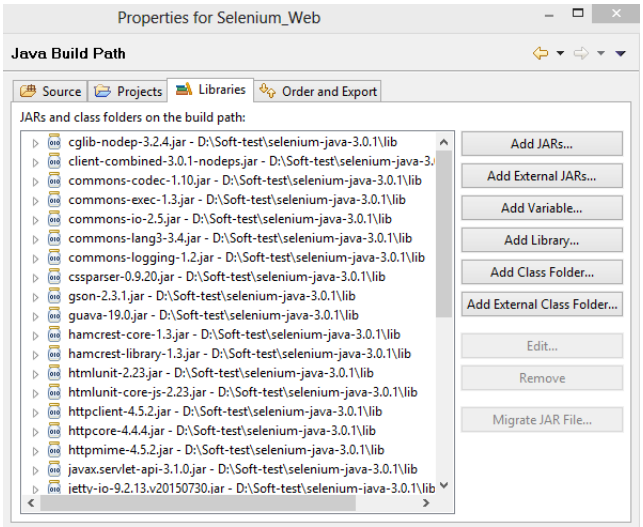


Figure 1: Java Build Path-Adding Jar Files

After adding of jar files, the package explorer of eclipse will be looked like the following:

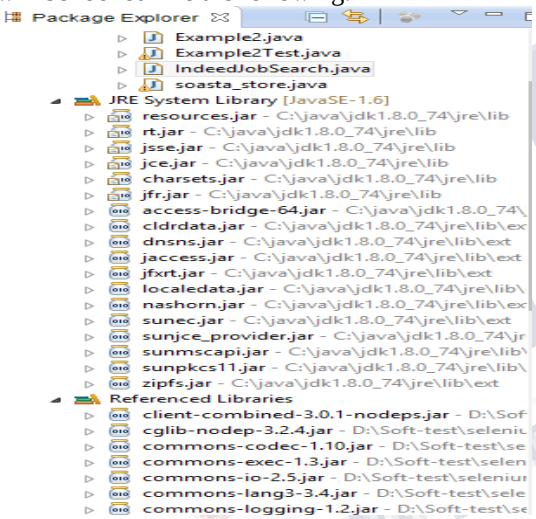


Figure2 : Package Explorer

III. WEB TESTING CODE

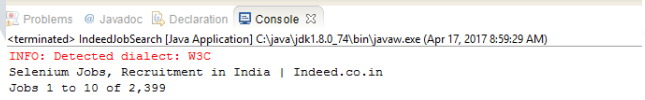
To test a website go to the site to be tested. For our example, " https://www.indeed.co.in/" is used. The web browser used in the testing is Mozilla Firefox. The geckodriver is used for the mozilla firefox browser for driving the browser.The coding starts with the inclusion of required selenium . The coding starts with the inclusion of required selenium packages.

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

public class IndeedJobSearch {
/**
     * @param args
     * @throws InterruptedException
     */
    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub
        WebDriver driver=new FirefoxDriver();
        driver.get("https://www.indeed.co.in/");
        Thread.sleep(2000);
        driver.findElement(By.id("what")).sendKeys("Selenium");
        driver.findElement(By.id("where")).sendKeys("India");
        driver.findElement(By.id("fj")).click();
        Thread.sleep(2000);
        System.out.println(driver.getTitle());
        System.out.println( driver.findElement(By.id("searchCount")).getText());
        driver.close();
    }

}
```

Fig(3)Selenium Webdriver Code for Testing Website

The above code explains the following:

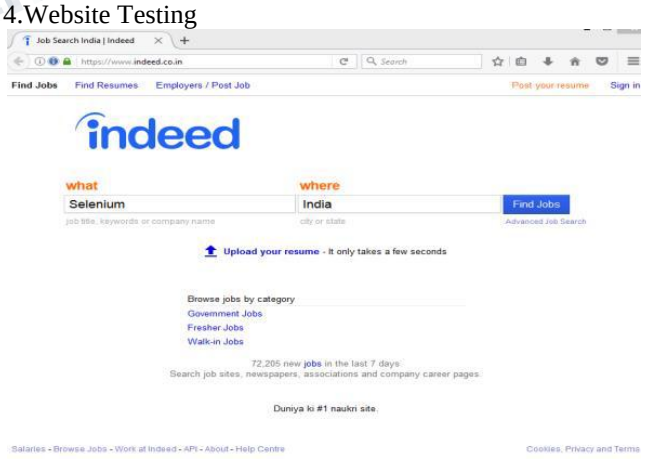
1. Import the selenium webdriver and firefox webdriver library files.
2. When a class implements an interface it provides the methods defined in that interface,with that we can use same method names for different web browsers.
3. Selenium sees everything on the page like textbox, button, link, dropbox etc as web elements.



Fig(4) (i)Console Output

1. The above code in fig (3) opens up the website specified in the code and it finds the textbox by using the id ,name,class name,tagname etc.
2. It will enter the selenium keyword in "what-find element by id" textbox and India in "where-find element by id" textbox and search

4.Website Testing



Fig(5)(a)Website Testing

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

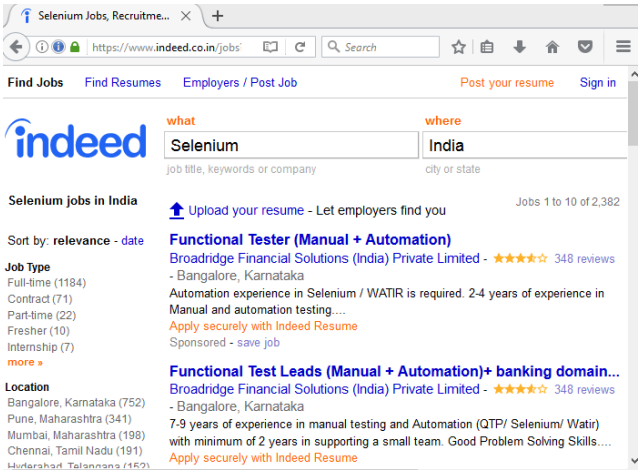


Fig 5(b)Webite Testing

The fig5(a) & (b) shows after the source code opened the website and searched the keyword accordingly.

IV. CONCLUSION

Selenium Webdriver directly executes commands on different browsers with out any proxy servers.Selenium Webdriver executes test scripts very fast because there is no another server to wait for language bindings commands to executes and it will take very less time to identify the web elements on different browsers.

REFERENCES

[1] Andreza M. F. V. de Castro, Gisele A. Macedo, Eliane F. Collins and Arilo C. Dias-Neto, Extension of Selenium RC Tool to Perform Automated Testing with Databases in Web Applications.AST 2013, San Francis -co, CA, USA.

[2] M.Archana Reddy K. Rajasekhar Reddy Selenium Webdriver Tool To Perform Auotmation Testing In Web Applications Volume No: 1(2014), Issue No: 10 (October) ISSN No: 2348-4845

[3] Insha Altaf 1 Jawad Ahmad Dar 2 Firdous ul Rashid 3 Mohd. Rafiq 4 SURVEY ON SELENIUM TOOL IN SOFTWARE TESTING 20 15 IEEE

[4] Nidhika Uppal Vinay Chopra Design and Implementation in Selenium IDE with Web Driver International Journal of Computer Applications (0975 – 8887) Volume 46– No.12, May 2012

[5] Ms. Rigzin Angmo Mrs. Monika Sharma Performance Evaluation of Web Based Automation Testing Tools 2014 IEEE

[6] Sanjeev Gupta Sunil Kumar Chirag Saxena Review Paper on Comparison of AutomationTesting Tools Selenium and QTP MIT International Journal of Computer Science and Information Technology, Vol. 5, No. 2, August 2015, pp. 55-57 ISSN 2230-7621©MIT Publications

[7] Satish Gojarea,*,Rahul Joshib,Dhanashree Gaigawarec Analysis and Design of Selenium WebDriver Automation Testing Framework 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)