

A faster approach for accessing Snap Deal URL using Multi browser with Selenium Web Driver

¹Mahan Sunhare, ²Abhishek Tiwari

Student (M-tech), Guide
MIT, Ujjain, India

ABSTRACT: In current paper, we are presenting a new approach for accessing Snap Deal URL using multiple browser support with the help of Web Driver which is faster than the approach using Selenium RC because of No interaction with the server as it interacts in selenium RC. In Selenium Web Driver, elements of the web page are accessible through XPATH; it makes faster access of the element than RC. Browser compatibility and consistency are the major challenges for the Testers. To overcome this problem created an approach which support the fastest URL access with multiple browsers using Selenium WebDriver which faster than the Selenium Remote Control (RC). Web Driver communicates with the browsers directly while selenium Remote Control communicates through the server that's why our proposed approach is faster than Selenium Remote Control (RC). The experimental results have proven that the execution of the current approach is faster.

1. Introduction

Test automation used in:

- Recurrent regression testing
- Round trip testing
- Quick feedback to developers
- Support for Agile based methodologies.
- Customized defect reporting.
- Finding defects skipped by manual testing.

Selenium is a collection of tools which enables fast development of automation for web supported applications. It is licensed under Apache License 2.0.

The selenium suite package consists of the Selenium Remote Control, WebDriver and Selenium Grid.

I. SELENIUM REMOTE CONTROL (RC)

Selenium Remote Control is an automation test tool. Selenium RC has two parts.

- A server which plays a role as a HTTP proxy for web requests automatically launches and closes browsers.
- Client libraries for computer languages.



Architecture of Selenium RC's is more complicated than selenium WebDriver.

In case of RC, first we require to begin a separate application i.e. Selenium Remote Control Server before perform testing. The Remote Control Server plays a role as an "agent" between Selenium commands given by you and your browser. When you start testing, Selenium RC Server injects a JavaScript program i.e. Core into web-browser. After injection, Core will start getting the instructions given by the Remote Control Server from the test program. After getting the instructions, Selenium Core would execute them as JavaScript commands. The browser would follow the instructions of Core, and would relay its output to the

Remote Control Server. The Remote Control Server will get the response of the browser then display the results to your Server will get the next instruction from your test script and repeat the whole cycle. Selenium Remote Control is slower since it uses a JavaScript program i.e. called Selenium Core. Core directly controls the web-browser, not you.

Selenium Remote Control 's API is more matured but contains some redundancies and also having few confusing commands.

As most of the time, testers are in dilemma that whether to use type or type Keys; or whether to use click, mouse Down, or mouse Down at. Worse, different browsers interpret each of these commands in different ways too!

II. SELENIUM WEBDRIVER

WebDriver is an automation tool for web application testing, and in specific to verify that they work as expected.

Goal of WebDriver is to support a friendly API that's easy to understand and explore, easier to use than the Selenium-Remote Control (1.0) API, which will support to make your tests easier to maintain and read.

It's not bind to any particular web based application test framework, so it can be used similarly well in a unit testing or from a simple old "main" method.

Selenium WebDriver supports the below mentioned browsers:

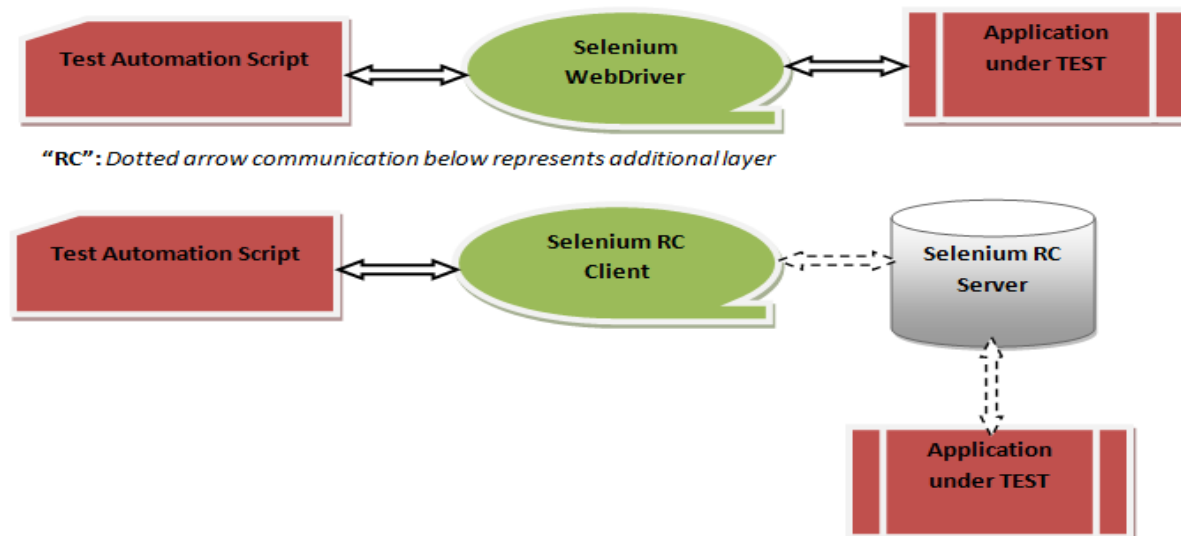
- Chrome,
- Internet Explorer (IE),
- Opera,
- Firefox
- Html Unit.
- Android enabled phones
- iOS enabled phones.

As a result, Selenium Web Driver is a most useful automated testing tool was developed called Selenium 2.0.

Web Driver is a purely object oriented testing framework.

In WebDriver, no server required to start.

WebDriver's architecture is much simpler than Selenium RC's.



2. Research Methodology:

Research work is used to calculate the performance of selenium web driver and also working of the selenium Web driver which is simpler than selenium RC.

In present work, we have evaluated performance of web driver using below mentioned code:

SnapDealController.java

```
package com.snapdeal;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
```

```

import org.openqa.selenium.support.ui.WebDriverWait;

public class SnapDealController {
    public static void main(String[] args) {

        SnapDealBrowsersLaunch browserLaunch = new SnapDealBrowsersLaunch();
        WebDriver driver = browserLaunch.LaunhBrowser("IE");
        WebElement serachBar = driver.findElement(By
            .xpath("//*[@id='inputValEnter']"));
        WebDriverWait wt = new WebDriverWait(driver, 60);
        wt.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//*[@id='inputValEnter']")));
        serachBar.clear();
        serachBar.sendKeys("Mobile Phones");
        WebElement serachButton = driver.findElement(By
            .xpath("//*[@id='sdHeader']/div[4]/div[2]/div[3]/button"));

        serachButton.click();
        WebDriverWait wtl = new WebDriverWait(driver, 60);
        wtl.until(ExpectedConditions.visibilityOfElementLocated(By
            .xpath("//*[@id='686534480658']/div[2]/a/img")));

        WebElement selectAnyMobile = driver.findElement(By
            .xpath("//*[@id='686534480658']/div[2]/a/img"));

        selectAnyMobile.click();
    }
}

```

SnapDealBrowsersLaunch.java:

```

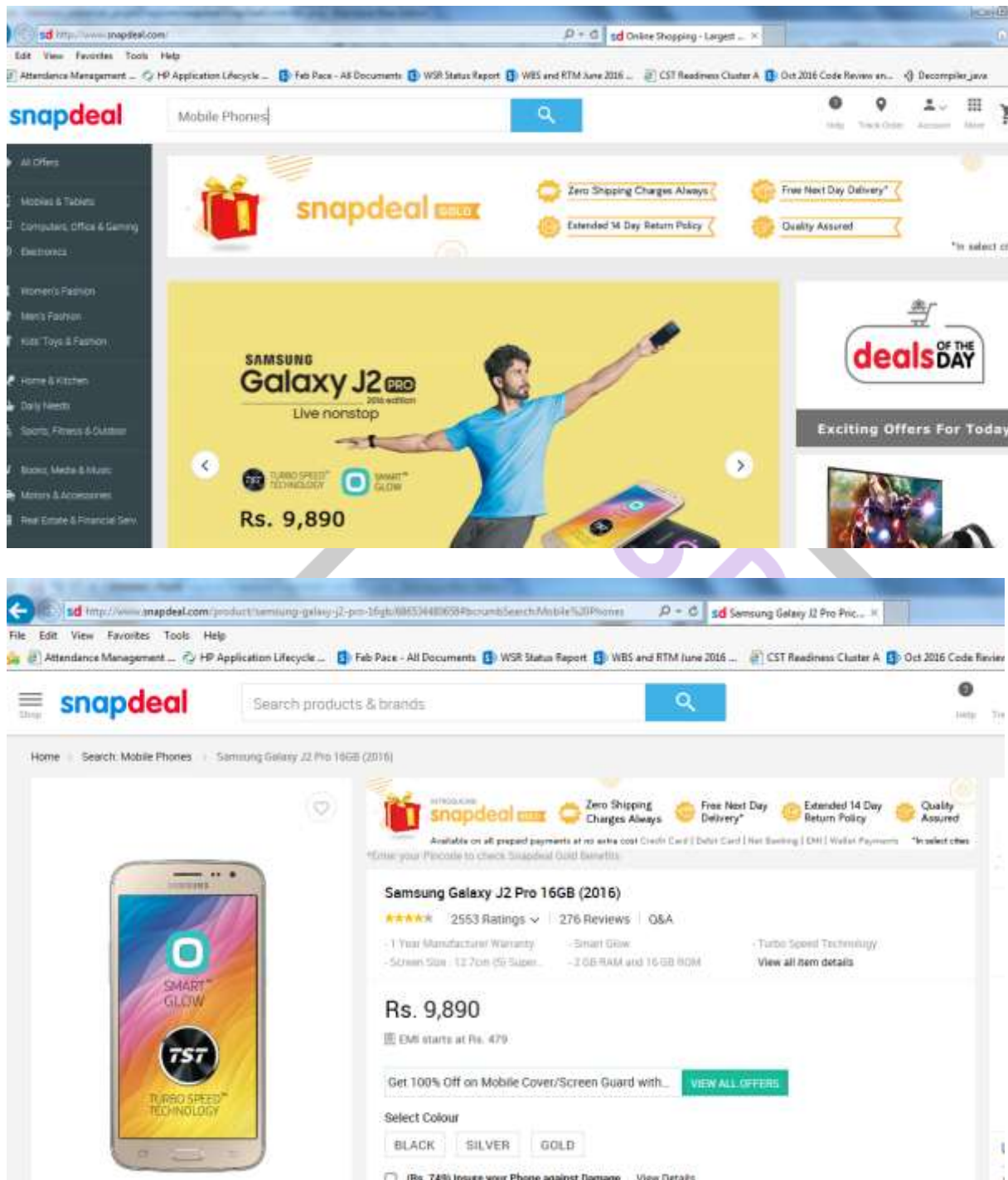
package com.snapdeal;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;

public class SnapDealBrowsersLaunch {
    WebDriver driver = null;
    String snapDealUrl = "http://www.snapdeal.com/";

    public WebDriver LaunhBrowser(String browsername) {
        if (browsername.equalsIgnoreCase("firefox")) {
            driver = new FirefoxDriver();
        } else if (browsername.equalsIgnoreCase("chrome")) {
            System.setProperty("webdriver.chrome.driver",
                "D:\\workspace\\Selenium_workspace\\MultiBrowser\\chromedriver.exe");
            driver = new ChromeDriver();
        } else if (browsername.equalsIgnoreCase("IE")) {
            System.setProperty("webdriver.ie.driver",
                "D:\\workspace\\Selenium_workspace\\MultiBrowser\\IEDriverServer.exe");
            driver = new InternetExplorerDriver();
        } else {
            System.out.println("Browser '{0}' not recognized. default Firefox
browser." + browsername);
        }
        driver.get(snapDealUrl);
        driver.manage().window().maximize();
        return driver;
    }
}

```

As output of the code as mentioned below in the screenshot which renders through the code automatically in a faster manner because of simple API and XPATH.



3.Related Work:

In SDLC, the use of test automation is to automate the execution of tests (test cases) and calculate the actual outcomes to expected outcomes. Automation Test tool can also be used to automate some constant tasks but add some extra testing that would be difficult to perform in manual testing [1]. With the era of internet technology, Web based application becomes more complex. The reliability and quality of the product is a matter of concern for the organizations. In Web based application testing, particularly the regression testing is more difficult task for manual testers [2], [3]. Software developers contribute to build "beautiful" pages but don't care about accessibility. Microsoft IE is used by the major companies so that the pages are viewed

only by this browser(IE), ignoring that in other browsers their pages look poorly, or (in some cases) some features are not visible [4]. Research is required to help focusing on Browser compatibility issue for consistent look and feel across the browsers. Therefore, it becomes important to carry out research by providing tools and mechanisms that would help towards Browser based testing in a faster way using Selenium Web Driver as compared to Selenium RC. Current paper intends to give a contribution in this direction by proposing browser based testing tool like Selenium Web Driver.

4. Conclusion:

In current paper, we are presenting a fast approach using web driver to access a snap deal URL using XPATH which faster than selenium RC because no extra layer or server is required to run a web testing. In current paper, we proposed an approach for accessing the Snap Deal URL using multiple browser support with XPATH which is faster than selenium RC because of direct communication with the browser instead of server interaction to communicate with the browsers. Current approach is simple and easy to use for an Automation Tester to validate web specific scenarios.

REFERENCES

- [1] A. Kolawa, D. Huizinga. (2007). Automated Defect Prevention: Best Practice in Software Management. *IEEE Computer Society Press*. ISBN 0-470-04212-5.
- [2] F. Cannizzo, G. Marcionetti, P. Moser,” Evolution of the Tools and Practices of a Large Distributed Agile Team”, in Proceedings of the Agile 2008 Conference, IEEE Computer Society, 2008, pp. 513-518
- [3] F. Coda, C. Ghezzi, G. Vigna *et al.* “A Software Engineering concept towards Web Based Development”. ‘*The 9th International workshop on ‘Software Design and Specification’*’, Ise-Shirna Japan:IEEE ‘Computer Society Press’, April 16-18, 1998, pp. 8-17.
- [4] J. T. Yang, J. L. Huang, F. J. Wang, and W. C. Chu, ‘An Object-Oriented Framework Supporting Web Based Testing’. IEEE twenty third Annual International Computer S/W and Application Conference, USA: IEEE Computer Society Press, October 2000, pp. 122-127.
- [5] M Kirchner, A standard approach for Web Based Testing for the Pages Accessibility. The 5th IEEE International Workshop on Web Site Evolution (WSE’03), 2003.