

---

# Designing Framework for Testing Web Applications

Amit Singh<sup>[1]</sup>, Kirti Bhatia<sup>[2]</sup><sup>[1]</sup>Deptt. of Computer Sc., SKITM, Bahadurgarh, Haryana<sup>[2]</sup>HOD, Deptt. of Computer Sc., SKITM, Bahadurgarh, Haryana  
amitdahiya.NITK@gmail.com, bhatia.kirti.it@gmail.com

---

**Abstract:** Test automation offers a possibility to perform regression and manual testing which can be run quickly and repeatedly. It can be a cost-effective method for regression testing of software products. Automation testing is developing and executing tests that can run unattended, comparing the actual to expected results. There are many tools in the market for automation but I have used Selenium. Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Selenium is a set of tools for cross platform automated testing of web applications.

---

## Introduction

Selenium is a set of different software tools each with a different approach to supporting test automation. Most Selenium users focus on the one or two tools that most meet the needs of their project, however learning all the tools will give you many different options for approaching different test automation problems. Selenium supports:

- Run in many browsers:
- IE, Firefox, Safari, Opera and other browser
- Support many operating system
- Windows, OS X, Linux, Solaris and other OS's
- Can be controlled by many programming language
- #, Java, Perl, Python, Ruby and other languages

## 1 Components of Selenium

*Selenium IDE**Selenium Remote Control**Selenium Web Driver**Selenium Grid*

### 1.1 Selenium IDE

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases. It's an easy-to-use Firefox plug-in and is generally the most efficient way to develop test cases. It is just a record and play tool.

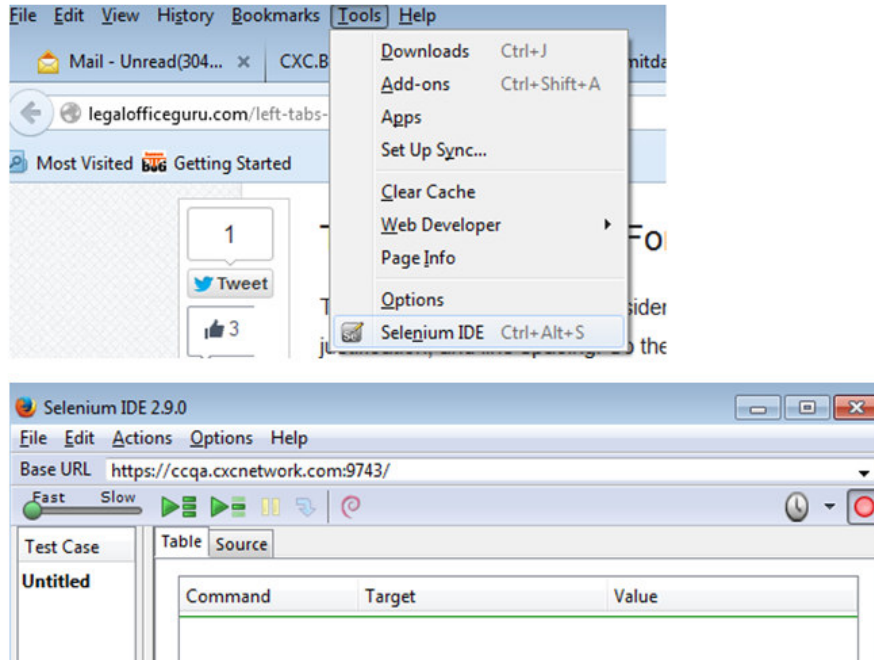


Figure1 Selenium IDE

### *Selenium Remote Control*

Selenium RC was the main Selenium project for a long time, before the WebDriver/Selenium merge brought up Selenium 2, the newest and more powerful tool. This version is kind of obsolete means it is used rarely only in legacy projects.

### *Selenium Web Driver*

The primary new feature in Selenium 2.0 is the integration of the WebDriver API. WebDriver is designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium-RC API. Selenium-WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded. WebDriver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems.

### Locating UI Elements (WebElements)

Locating elements in WebDriver can be done on the WebDriver instance itself or on a WebElement. Each of the language bindings exposes a "Find Element" and "Find Elements" method. The first returns a WebElement object otherwise it throws an exception. The latter returns a list of WebElements, it can return an empty list if no DOM elements match the query.

1. *Locate Element By ID-*  
`WebElement element=driver.findElement(By.id("Savebutton"));`
2. *Locate Element By Name-*  
`WebElementcheese=driver.findElement(By.name("cheese"));`
3. *Locate Element By link Text-*  
`WebElementcheese=driver.findElement(By.linkText("cheese"));`
4. *Locate Element By partial link text-*  
`WebElementcheese=driver.findElement(By.partialLinkText("cheese"));`
5. *Locate Element By CSS-*  
`WebElementcheese=driver.findElement(By.cssSelector("#food.dairy.aged"));`
6. *Locate Element By xpath-*  
`driver.findElements(By.xpath("//input[@type='button'][@value='submit']"));`

### Implementation

Steps that needs to do when you want to work with Selenium Web Driver

1. Install Eclipse from:---<https://www.eclipse.org/downloads/>
2. Create a Project  
Go to File →New→Project
3. Create a new Package under the above created project  
Go to File→New→Package
4. Create a new Class under the above created Package  
Go to File→New→Class
5. Add external Jar into the newly created project ( we need to add Selenium JAR, Test NG Jar and Sikuli JAR)
6. Right click on Project folder →Properties→Java Build Path→ Add External JARs

Now you are ready to right the code ----

### ACKNOWLEDGMENTS

I would like to thank my guide, Kirti Bhatia, HOD SKITM who escortme throughout this project..

### REFERENCES

1. <http://www.seleniumhq.org/>
2. J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
3. S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.
4. M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.
5. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.