

# Examining Application of Selenium IDE For Software Testing and It Automation

Damini Khadakkar<sup>1</sup>, Vinay Kumar Sharma<sup>2</sup>, Steffi Zachariah<sup>3</sup>

<sup>1,2,3</sup>Department of Information Technology

<sup>1,2,3</sup>AISSMS IOIT

**Abstract-** In today's world web is the need of hour. Most of the crucial transactions and tasks are being done online in the form of web applications. Testing these web applications is an important task. Testing a web application manually is tedious, so there is a need to go for automation testing. In automation testing there is utilization of a product device to run repeatable tests against the application to be tried. There are various focal points of automation testing. There are various open source and business devices accessible for test mechanization. Selenium is one of the broadly utilized open source device for test computerization. Test automation enhances the effectiveness of programming testing procedures. Test automation gives quick criticism to engineers. It additionally discovers the imperfections when one may miss in the manual testing. In test automation we can perform boundless emphases for testing the same example of code ceaselessly commonly. The application of this tool can be seen in the automation of monthly allowance calculation of the employees of an organization. Manually this tasks usually takes more than a week's span but the use of Selenium IDE can reduce it up to a few hours.

**Keywords-** Selenium , automation, selenium grid, selenium IDE, selenium web driver, RC

## I. INTRODUCTION

Web testing is an important factor for program testing that focuses on web applications. Complete testing of an electronic application is essential with a specific end goal to recognize and adjust the blunders, before it is uncovered to the general population. There are various web application execution devices available. We utilize them to test the web applications and web related interfaces. Regardless of testing web applications, we can test web servers, sites and other web interfaces by making utilization of automation web testing. All these available techniques have their pros and cons and hence there is a need to find the best possible technique amongst all these to reduce the efforts required, minimize the time required and provide maximum efficiency to the task performed. Selenium data integration tool thus provides all the above features to enable automatic software testing.

## II. TECHNIQUE

### 1. Manual Testing:

This is the most traditional way of web based application testing. The web applications are tested step by step and sequentially. This process should be continuously monitored and checked for the possible errors. Manual testing requires comparatively more time. The test cases should be error free and should be able to identify possible bugs in the web applications. The main drawback of this type of testing method is that for the repetitive test cases, the same test has to be implemented again and again. This is a very time consuming and a tedious process. Repetitive testing is an integral part of regression testing and hence it cannot be avoided. But it can be simplified with the use of proper automated testing techniques.

### 2. Automated Testing:

Automated testing tools are capable of executing tests, reporting outcomes and comparing results with earlier tests. Test carried out with these tools can be run repeated at any time of the day. The method or process being used to implement automation is called test automation framework. Unlike manual testing method automated testing is an efficient way of web application testing. It has a proper interface for writing the test cases, interconnection with different web drivers as well as a vast language support for writing the test cases.

Selenium is one such automated testing tool from amongst a number of software testing tools available in the market. Selenium was made and engineered by Jason Huggins in 2004. He started his research when he was trying an interior application physically at thought works and he realized that he can enhance utilization of his time instead of manually going through the same steps. Initially he began with java script and made a library in java scripting that can interface with the site page and can run the same test automatically against a number of programs. That java script library then got to be Selenium Core, which contains every one of the properties of Selenium Remote Control (RC)<sup>[4]</sup> and Selenium IDE<sup>[5]</sup>. Probably

Selenium was an huge device, yet it had its pros<sup>[2]</sup>. The security of Selenium was not all that precise because of which a portion of the things were difficult to do in Selenium<sup>[3]</sup>.

### III. INTRODUCTION TO SELENIUM-IDE

The Selenium-IDE (Integrated Development Environment) is the instrument used to build up your Selenium experiments. It's a easy to use Firefox module and is the most effective approach to create experiments. It also contains a setting menu that allows you to chose UI component. This is a period saver, as well as a path breaking method for learning Selenium script syntax<sup>[5]</sup>.

#### IDE FEATURES

##### Menu Bar:

The File menu has alternatives for Test Case and Test. Utilizing these you can include another Test Case, open a Test Case, spare a Test Case, and fare Test Case. Every one of these alternatives are accessible for Test Suite<sup>[7]</sup>.

The Edit menu permits duplicate, glue, erase, fix, and select all operations for altering the orders. The Options menu enables the changing of settings. You can set the esteem for specific orders, add client characterized client expansions to the base arrangement of Selenium summons, and determine the organization (dialect) utilized when sparing your experiments. The Help menu is the standard Firefox Help menu.

##### Toolbar:

The toolbar contains catches for controlling the execution of the experiments, including a stage highlight for investigating the test cases[10].

##### Test Case Pane:

The script is shown in the experiment sheet. It has two tabs, one for showing the charge and their parameters in a table like organization.

The other tab - Source shows the experiment in the local organization in which the record will be stored. This is HTML in spite of the fact that it can be changed to a programming dialect. The Source see permits one to alter the experiment in its crude structure, including duplicate, cut and glue operations<sup>[11]</sup>.

The Command, Target, and Value passage fields show the present chose order along with its parameters. These are passage fields where you can adjust the chose order. The primary parameter determined for a charge in the Reference tab of the base sheet dependably goes in the Target field. In the event that a second parameter is indicated by the Reference tab, it generally goes in the Value field<sup>[12]</sup>.

##### Roll up pane

The base sheet is utilized for four different functions- Log, Reference, UI-Element, and Rollup

##### Log

When you run the experiment, error messages and data messages showing the advancement are shown in this sheet naturally, irrespective of the possibility that you don't first choose the Log tab. These messages are frequently valuable for experiment troubleshooting. Also see the Info catch is a drop-down permitting determination of different levels of data to log<sup>[13]</sup>.

##### Element and Roll-up:

Point by point data on these two sheets (which cover propelled components) can be found in the VI-Element Documentation on the Help menu of Selenium-IDE.

### IV. SELENIUM RC

Selenium RC was the main Selenium project for a long time, before the Web Driver/Selenium merge brought up Selenium 2, the newest and more powerful tool<sup>[7]</sup>.

Selenium I is still actively supported and provides some features that may not be available in Selenium 2 for a while, including support for several languages (Java, JavaScript, Ruby, PHP, Python, Perl and C#) and support for almost every browser out there.

### V. HOW SELENIUM RC WORKS

First, we will describe how the components of Selenium RC operate and the role each plays in running the test scripts.

#### RC COMPONENTS

Selenium RC segments are:

- The Selenium Server which dispatches and murders programs, translates and runs the Selene charges

released through from the test program, and goes about as a HTTP proxy, catching and checking HTTP messages went between the program and the AVT.

- Client libraries which give the interface between every programming dialect and the Selenium RC Server.

Selenium I is still actively supported and provides some features that may not be available in Selenium 2 for a while, including support for several languages (Java, JavaScript, Ruby, PHP, Python, Perl and C#) and support for almost every browser out there.

## VI. SELENIUM SERVER

Selenium Server gets Selenium orders from your test project, translates them, and reports back to your system the effects of running those tests.

The RC server packs Selenium Core and naturally infuses it into the program. This happens when your test system opens the program. Selenium-Core is a JavaScript program, an arrangement of JavaScript capacities which translates and executes Selenese orders utilizing the program's fabricated as a part of JavaScript interpreter<sup>[11]</sup>. The Server gets the Selenese summons from your test project utilizing basic HTTP GET/POST asks. This implies you can utilize any programming dialect that can send HTTP solicitations to mechanize Selenium tests on the program.

## VII. WEB DRIVER AND THE SELENIUM SERVER

You may, or may not, require the Selenium Server, contingent upon how you expect to utilize Selenium-Web Driver. On the off chance that you will be just utilizing the Web Driver API you needn't bother with the Selenium-Server. On the off chance that your program and tests will all keep running on the same machine, and your tests just utilize the Web Driver API, then you don't have to run the Selenium-Server; Web Driver will run the program straightforwardly.

There are a few reasons however to utilize the Selenium-Server with Selenium-Web Driver.

- You are utilizing Selenium-Grid to dispatch your tests over numerous machines or virtual machines (VMs).
- You need to associate with a nearby machine that has a specific program form that is not on your present machine.

## VIII. SELENIUM GRID

Selenium-Grid permits you run your tests on different machines against different programs in parallel. That is, running various tests in the meantime against diverse machines running distinctive programs and working frameworks. Basically, Selenium-Grid<sup>[8]</sup> bolster disseminated test execution. It takes into account running your tests in an suitable test execution environment<sup>[12]</sup>.

### WHEN TO USE IT

For the most part talking, there are two reasons why you might need to utilize Selenium-Grid.

- To run your tests against a number of programs, various renditions of program, and programs running on different working frameworks.
- To lessen the time it takes for the test suite to finish a test pass.

Selenium-Grid is utilized to speed up the execution of a test go by utilizing different machines to run tests in parallel. Some test suites can take hours to run. Another motivation to support the time spent running the suite is to shorten the turnaround time for test results after designers registration code for the AUT. Progressively programming groups honing lively programming improvement need test input as quickly as could be expected under the conditions instead of sit tight overnight for an overnight test pass<sup>[13]</sup>.

Selenium-Grid is additionally used to boost running tests against a number runtime situations, chiefly, against distinct programs in the meantime. When the test suite is run, Selenium-Grid gets every test-program mix and assigns every test to keep running against it's required browser<sup>[11]</sup>.

What's more, one can have a matrix of all the same program, sort and form. At the point when the suite runs, every test is gone to Selenium-Grid which doles out the test to the following available Firefox example. In this way one gets test finish where possibly 12 tests are all running in the meantime in parallel, altogether decreasing the time required to finish a test pass<sup>[17]</sup>.

Selenium-Grid is tremendously flexible. These two illustrations can be consolidated to allow a variety of occurrences of every program sort and form. A design, for example, this would give both, parallel execution for quick test pass consummation and backing for various program sorts and forms simultaneously<sup>[17]</sup>.

## IX. CHALLENGES

As you know Selenium is a free ware open source testing tool. There are many challenges with Selenium. One of the biggest drawbacks with user interface testing tools is that they're slow for various reasons. One way to bring the test run times down is to run them in parallel on a grid of servers, Engineers would be more likely to run automated browser UI tests if they could run 1000 tests in 1 minute total time on 1000 machines instead of 1000 tests in 1000 minutes on 1 machine. But, most projects allocate only one machine, maybe two, to browser testing. Selenium supports only web based applications. It doesn't sustain any non-web based (Like Win 32, Java Applet, Java Swing, .Net Client Server etc.) applications. When you contrast selenium with QTP, Silk Test, Test Partner and RFT, there are many challenges in terms of maintainability of the test cases. Since Selenium is a freeware tool, there is no direct support if one is in trouble with the support of applications. There are many challenges if one has to interact with Win 32 windows even when you are working with Web based applications. Bitmap comparison is not supported by Selenium. Any reporting related capabilities, you need to depend on third party tools. Selenium sometimes can't even automate on web applications under some circumstances:

- 1) For example when the application is made in flash.
- 2) Or the application has custom objects or objects that can be inspected or that code can't be viewed for example silver light applications<sup>[15]</sup>.

## X. CONCLUSION

Improvement on Selenium is continuing at an fuming speed, and new elements are included day by day. The Selenium designers are greatly open to every one of the inquiries tended to the selenium-clients and selenium-level mailing records. Selenium computerizes programs. It is intended for mechanizing web applications for testing purposes, yet is surely not restricted to simply that. Exhausting electronic organization odd jobs can likewise be computerized too. Selenium has the backing of a portion of the biggest program merchants who have taken strides to make Selenium a local piece of their program. It is additionally the center innovation in countless other program robotization instruments, APIs and systems. Selenium collaborates with the UI components of a site/application utilizing JavaScript infusion. In this manner a custom intermediary server is utilized to inculcate the JavaScript code into the html source. At the point when utilizing the Selenium IDE this charm is done consequently. It not, Selenium RC is utilized to control

the html source to be robotize capable. The Selenium IDE is the slightest demanding and speediest route to make tests and replay them a short time later inside the program. On the off chance that you need to make fast bug multiplication scripts and to help in robotization supported exploratory testing, Then you need to use Selenium IDE; a Firefox add-on that will do simple record-and- playback of communications with the program. On the off chance that you need to make influential, program based relapse computerization and scale and distribute scripts crosswise over many situations, then you need to utilize Selenium Web Driver; an accumulation of dialect particular ties to drive a program - the way it is planned to be driven. Selenium Web Driver is the successor of Selenium Remote Control which has been confidently censured. The Selenium Server likewise incorporates assembled in lattice abilities.

## REFERENCES

- [1] Martin Fowler, Kent Beck, John Brant, Refactoring: Improving the Design of Existing Code, U.S., 1999, pp. 11-21.
- [2] Martin Fowler, Kent Beck, John Brant, Refactoring: Improving the Design of Existing Code, U.S., 1999, pp. 2 - 20.
- [3] Antawan Holmes and Marc Kellogg, Automating Functional Tests Using Selenium, Proceedings of AGILE 2006 Conference, 2006.
- [4] McMahon, C, History of a Large Test Automation Project Using Selenium, 2009.8.
- [5] Shauvik. R. Choudhary, Dan Zhao, WATER: Web Application Test Repair, ACM, 20 II.
- [6] Marian, Michal. Automated acceptance testing tools for Web application using Test-Driven Development. pzreglad Elektro Techniczny. 2010-9.
- [7] Tuomas Pajunen, Tommi Takala and Mika Katara. Model-Based Testing a General Purpose Keyword-Driven Test Automation Framework. International Conference on Software Testing. Verification and Validation Workshops. 2011.
- [8] Antawan Holmes and Marc Kellogg, Automating Functional Test Using Selenium. Proceeding of Agile Conference, 2006.
- [9] Chris McMahon. History of a Large Test Automation Project Using Selenium. Agile Conference. 2009.

- [10] Xinchun Wang, Peijie Xu, Build an Auto Testing Framework Based on Selenium and FitNesse. International Conference on Information Technology and Computer Science. 2009.
- [11] Zhenghua Feng, Ju Gao, Hongwei Zeng, Research of Web Application Test Automation, Computer Engineering and Design, 2010.
- [12] Web Site, <http://docs.seleniumhq.org/docs/>
- [13] Pan Liu and Huaikou Miao, A New Approach to Generating High Quality Test Cases, 19th IEEE Asian Test Symposium, 2010, pp.71-76.
- [14] F. Ricca and P. Tonella. Analysis and Testing of Web Applications. In: Proc. of the International Software Engineering Conference. Toronto Canada: IEEE Computer Society Press, May 2001, 25-34.
- [15] Andreas Bruns, Andreas Kornstadt, and Dennis Wichmann, Web Application Test With Selenium, IEEE Software, 2009, pp. 88-91.
- [16] C. A. R. Hoare. An axiomatic basis for computer programming. Communications of the ACM, 12(10):576-580,583 October 1969.
- [17] R. W. Floyd. Assigning meanings to programs. Proceedings of the American Mathematical Society Symposia on Applied Mathematics. Vo.