

Analysis and implementation of selenium automation using selenium web driver and ATF

Reetu Tiwari¹, Khushboo Sawant²

MTech Scholar, Computer Science Engineering, LNCT, Indore, Madhya Pradesh, India¹

Assistant Professor, Computer Science Engineering, LNCT, Indore, Madhya Pradesh, India²

tiwarireetu1995@gmail.com¹, sawantkhushboo@gmail.com²

Abstract: *Number of software system has been implemented as web applications which are very complex and its quite hard to test these applications. Automation testing uses automation tools to reduce human intervention and repeatable tasks. Selenium is a web application testing framework also a freeware software that is easily available. This automation testing framework has gained wide acceptance as a popular and successful mode of automated testing in a very short time span. Proposed testing freeware will render us an efficient way of performance and other parameters to ascertain compatibility, accuracy, aspect and consumption of web applications which suite for new release of the application using this framework.*

Keywords: *Web applications; Automation testing; selenium web driver; Automation testing framework.*

1. INTRODUCTION

The main aim of software testing is to execute the application and find the fault in application as early as possible. Software testing consumes 30 to 60 percent of all life cycle cost, depending on product criticality and complexity [1]. With the development of latest technologies, web applications became more popular that in this era large number of software systems has been implemented as web applications. The quality of these web applications is one of the most focusing concern while deploying these web applications. So, to increase the quality of software, testing plays a vital role. SDLS becomes shorter and shorter; this makes the software testing more difficult. Manual testing is a time taking process and it require human intervention. So, to avoid these issues, automation testing came into page. To support these tasks there are various commercial and open source tools available, such as Watir, JMeter, Selenium, QTP and many other. For this paper we have proposed automation testing framework based on the selenium web driver and TestNG tool.

2. RELATED WORK

Selenium is an open source automated testing suite for web applications across different browsers and platforms

which supports multiple programming language. Selenium is a type of functional web testing tool and only a single tool which has four components: Selenium Grid, Selenium Remote control, Selenium IDE and Selenium Web Driver [2]. It is Integrated development environment which has Firefox plug-in, i.e plugins which can be used to design test cases, Selenium Remote control run tests inside every JavaScript compatible browser (which is now available on all web browsers), Web Driver was developed for better support for dynamic web pages where elements of a page may change without the page itself being reloaded and Selenium Grid allows you to run your tests on different machines against different browsers [1].

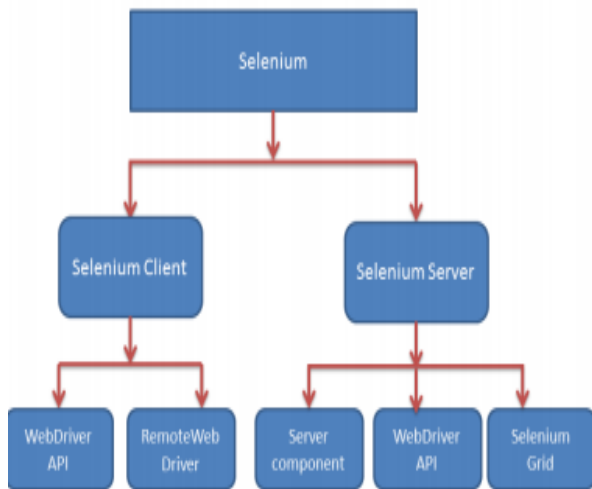
Automated testing is used by developer to save assets and time. Selenium will run directly in the web browser and supports almost all available browsers like Google Chrome, Mozilla Firefox, Microsoft's Internet Explorer, Opera, and Macintosh Safari as well as it will support all platforms like Linux, Windows and Mac. It is a very useful tool for System functional analysis and browser relationship or compatibility testing as well as compare to other available automation tools it is very flexible and simple to use. [3] For this vast information platform and quick release cycle quick regeneration of site is required. This requires the application to be all-inclusive, extensibility and efficiency. For this there exists some framework which includes automation tools. In

this paper we will discuss about the result analysis from the automation testing tool as we know sometimes there may be an issue arising to choose proper framework for automation testing. Selenium is a set of tools functioning with many Operating Systems, browsers, programming languages and other different testing framework, individually with other different approaches in support of automation test for testing web-based application [4].

3. AUTOMATION COMPONENTS

- ❖ Selenium IDE
- ❖ Selenium RC
- ❖ Selenium WebDriver
- ❖ Selenium Grid

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases. It's an easy-to-use Chrome and Firefox extension and is generally the most efficient way to develop test cases. It records the user's actions in the browser for you, using existing Selenium commands, with parameters defined by the context of that element. This is not only a time-saver, but also an excellent way of learning Selenium script syntax.



Selenium Remote Control (RC) is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website using any mainstream JavaScript-enabled browser.

Selenium RC comes in two parts.

1. A server which automatically launches and kills browsers and acts as a HTTP proxy for web requests from them.
2. Client libraries for your favorite computer language.

WebDriver is a web automation framework that allows you to execute your tests against different browsers, not just Firefox, Chrome. **WebDriver** also enables you to use a programming language in creating your test scripts (not possible in **Selenium IDE**)

Selenium Grid is a part of the Selenium Suite that specializes in running multiple tests across different browsers, operating systems, and machines in parallel.

Selenium Grid has 2 versions - the older Grid 1 and the newer Grid 2.



4. RESULT ANALYSIS

We have planned to design automation platform based on selenium web driver and TestNG as well as ReportNG tool (Fig. 1). The framework designed in this paper includes five components listed below.

- ❖ Object Components
- ❖ Intake File
- ❖ Utility Section
- ❖ Reports and test cases
- ❖ Revised Report

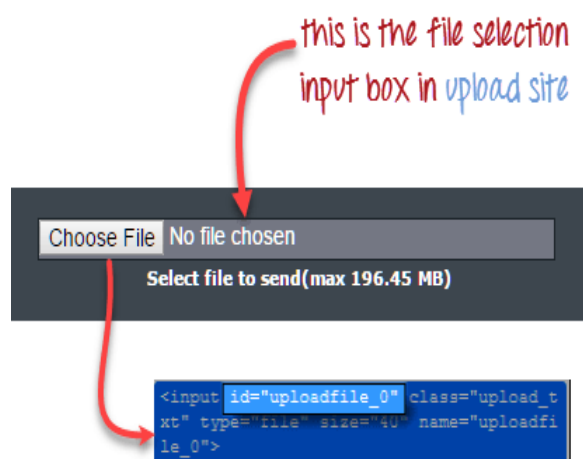
4.1 Object Components

Web driver of selenium supports various types of locator to locate the web page elements. They can be located by its id, link text, and xpath or css locators. Object repository stores all the locators of web page elements. This will simplify the task of writing the test cases. For e.g. previous version of web application contains 'Login' button. In next version of web application 'Login' button changed to 'Login Now', so it is required to change all the test cases which contains the 'Login' button [2]. To avoid such kind of problems, we have implemented object repository which contains the id's, xpath and link text for all web page elements. Whenever tester writes the test case, tester will use the information to

locate the web page element. This will reduce the maintenance cost of test cases. [5] Whenever change occurs in web application elements, tester needs to change only object repository.

4.2 Intake file

In any application user needs to enter some of the input like for hitting the RGPV website and move inside the portal we need to enter the username and password Such kind of inputs stored in intake file. Rather than entering same information in web application, tester may access these inputs from input file. In this file tester can store the input values required by web application.



The test cases using the TestNG as well as ReportNG the two-testing framework to generate the test report. TestNG generates the report in HTML format, which is tedious to understand. Some organization want specific format of report. So, there is need to customize the TestNG We get is a small and easy-to-use command-line program used to automate downloads. Basically, we will access We get from our WebDriver script to perform the download process.

4.3 Utility Section

Utility section contains two files which are described below.

❖ User Actions File:

Selenium web driver doesn't support the direct functions to perform certain operations like clicking a button, selecting checkbox etc. This section contains the common functions like click button, select checkbox, click link etc. This will diminish the redundancy of code in script. This file also contains the application specific functions for example if web

application contains table and you need to verify particular column is sorted or not.

❖ Utility file:

This file comprises the common functionality of web application like login and logout. In test suite tester need to login to web application to test internal functionality of application and log out after completion of suite it needs to logout. To avoid this kind of repetition, we have added login and logout functions in utility file. Screenshot Generation: Selenium web driver does not support the screenshot for failure test cases. [9] We will be going to try some new function that will take the screenshot for failure test case only. Using this function tester can easily capture the error occurred in web application. This will also addons to developer to analyses their failure. After execution of test suite, screenshots for failure test cases are stored in directory according to date wise folder. Steps to generate screenshot

1. Make a directory where we can store the screenshot for pass and failure test case.
2. Capture the result from TestNG and ReportNG
3. Checkout the test reports
4. Cross verify if the test cases are passed then store it in the directory according to the past cases that should be sequentially arranged in order of the test priority
5. Set the date and time for screenshot Image as title.
6. Store the image file in mentioned directory.

4. 4 Reports and test cases

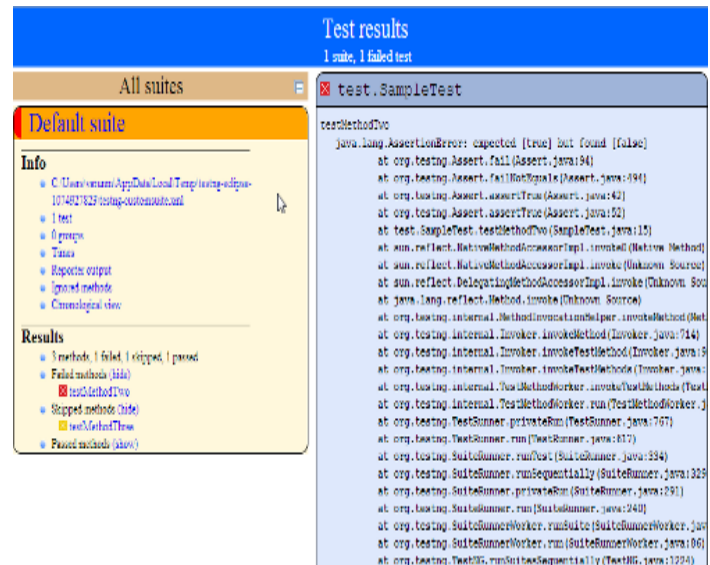
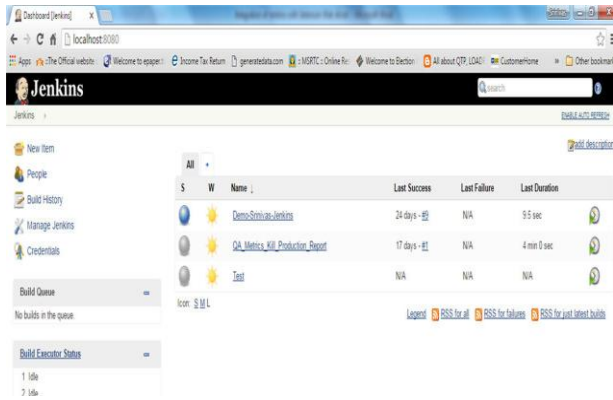
Selenium driver do not have any functionality regarding the storage of the test cases so for that we will be using some of the tolls for the generation of report according to organization. We have implemented one class in which we have captured the TestNG report. We have used ReportNG for generating the test report as well as thought of using some different tool for generating the report in the pdf format so that this report will have the table of the passes and failed test cases as well as the possibly the screenshots for the same also we tried of calculating the time for each of the test cases to pass [6]. So, one can easily verify the error page of web application.

4.4.1 Jenkins Implementation:

We thought of implementing all the test scenarios with the automated build creation tool that is Jenkins so that the test cases will be executed regularly by the triggering of Jenkins build. For that we will be performing the following steps: [8]

1. Creating a job in Jenkins
2. Completing the build configuration for that job.

3. Try to execute the test cases using the batch file
4. Configuring an automatic build creation in Jenkins



5. REVISED REPORTS

Once we are done with writing out the test scripts the most awaited of all is to get the generated test report and it is observed that after proper usage of both the test script as well as the testing framework the regression testing efficiency is highly improved. Tester can write the test cases twice than older approach of writing test cases. This minimizes the human resource required for testing out the applications. The maintenance cost of test cases also reduced due to centralized repository. This framework reduces the error rate of failing the test cases due to synchronization issues. Ultimately passing rate will increase which shows how much accurate proposed framework is over the traditional approach of testing. We will try to be executed test suite of more than 50 test cases on student any of the web application. After automation run we got the following results (Table 1.) in terms of overall pass rate, failure rate, execution time etc. [7] Proposed framework synchronizes the test cases properly, so failure rate is reduced than traditional approach.

5.1 Testing report template:

We can have an idea or a view that how actually the report will be generated using the TestNG framework for our test scripts:

6. CONCLUSION AND FURTHER WORK

In this paper we have proposed an idea of using the new automation testing framework to test the web-applications based on selenium web driver which will surely reduce the time required to write the test cases and increase the pass percentage of test cases. It will further reduce the hectic workload of tester. By using this framework, we can generate the customized reports and analyze the failures using screenshots. Tester can maintain all the data from a central place. They are also helpful for dynamically changing web applications. The automation test scripts are easy to understand using this framework.

REFERENCES

- [1] Chandraprabha, Ajeet Kumar, Sajal Saxena - "SYSTEMATIC STUDY OF A WEB TESTING TOOL: SELENIUM". IJARSE -ISSN-2319-8354(E), Vol. No.2, Issue No.11, Pages-113-120, November 2013
- [2] Richa Rattan - "COMPARATIVE STUDY OF AUTOMATION TESTING TOOLS: QUICK TEST PRO AND SELENIUM". VSRD International Journal of Computer Science & Information Technology, Vol. 3 No. 6 June 2013
- [3] Sherry Singla, Harpreet Kaur. "Selenium Keyword Driven Automation Testing Framework". IJARCSSE ISSN: 2277 128X, Volume
- [4] <http://www.seleniumhq.org/>
- [5] <http://www.tutorialspoint.com/selenium>
- [6] P.Yadav and A. Kumar, "AN AUTOMATION TESTING USING SELENIUM TOOL," International Journal of Emerging

Trends & Technology in Computer Science (IJETTCS), vol. 4, Issue 5(2), pp. 68-71,2015.

- [7] S.Gojarea,R. Joshi and D.Gaigaware, “Analysis and Design of Selenium WebDriver Automation TestingFramework,” Procedia Computer Science, vol. 50,pp.341 – 346, 2015.
- [8] <http://seleniummaster.com/sitecontent/index.php/introduction-to-selenium-automation/selenium-ide/85-what-are-the-main-features-of-selenium-ide>.
- [9] <http://easyonlineconverter.com/>.