

IMPLEMENTATION OF DATA DRIVEN AUTOMATION TESTING OF WEB APPLICATIONS USING SELENIUM

Janani. V¹, Krishnamoorthy .K², Revathi. K³

¹Assistant Professor, ²Professor & Head, ³PG Scholar ,Dept of CSE,

Adhiyamaan College of Engineering, Hosur, (India)

ABSTRACT

Software testing is a process of executing a program or application with the intent of finding the software bugs. Testing Automation would allow us to reduce the cost and will help us in increasing the quality of Software. Automation Framework follows Data Driven Approach which passes data stored in Spreadsheet as input to the number of test scripts written for executing test cases. Data-Driven Framework is where test input and output values are from data files and are loaded into variables in captured or manually coded scripts. The framework should include navigation through the program, reading of the data files, and logging of test status and information. All the processes should be in the test scripts and each process finds the errors in the software and generates a list of errors. Those errors are properly listed in a Spreadsheet and at the same time also contain the code area and line where it appeared, type of error and its possible solutions. This paper includes integration of selenium and JMeter for load testing.

Keywords: *Automation Testing, Data Driven Approach, Driver Script, Jmeter, Selenium Webdriver Test Case.*

I INTRODUCTION

Software testing services and solutions would be used in IT sectors for large applications. The software testing landscape is increasingly moving towards becoming a joint effort across various levels of the organization through a more service-driven test management process, leaving behind the traditional testing center of excellence model. A test automation framework is a set of assumptions, concepts and tools that provide support for automated software testing. The main advantage of such a framework is the low cost for maintenance. Software testing automation has long been recognized for its potential to improve testing capacity by reducing redundant, manual operations while maximizing test accuracy. Testing all these application is mandatory before launching them in the market. But at the same time testing all these applications will require a lot of time and human effort and can also have some probability of human errors. Therefore incorporating Automation in testing is another step demanded by today's need of technological advancement. Data-Driven testing means executing a set of steps with multiple sets of data. The Framework runs a driver script which reads the test cases from a list in a spreadsheet and accordingly selects the testing functions/scripts and concurrently generates a logger report,

result report and screenshots of each test case execution and finally generating a complete report on tested application/software, after comparison of the actual result with the expected result.

II FRAMEWORK DETAIL

2.1. Process

- Initiation of the Testing Process

Automation Framework will start executing tests with a push of a button and run tests on its own. It can set up the test environment and preferably also check that all preconditions are met. It will read the Test ID from the Spreadsheet and automatically start the testing process.

- Test Execution

It is possible for us to start the test execution manually. It can be a convenient method because tests can be started automatically at a specified time or after a certain event. Easiest way to start execution at a certain time is making it possible to start test execution manually from command line and using operating system's features for scheduling. The testing process will stop automatically after generating the Test Report, which will be a combination of Logger Report, Result Report and Screenshots. The link of Test Report will be automatically sent to the Tester and in the Spreadsheet

- Verifying Test Result

An integral part of test execution is verifying test results. Framework will automatically verify test result as one or more comparisons between actual outcome of a test and predefined expected outcome. The comparison will be done by referring the Spreadsheet which contains a column containing the data for Expected Result.

- Assigning Test Status

After a test is executed and its results are verified, it is expected from the testing framework that it generates a status for the same. If the test was executed without any problems and all comparisons between actual and expected outcomes match the test gets a pass status. In every other case the status is fail. Besides the status every test case should also get a short but descriptive status message. For passed tests this message is normally not that important but with failed tests it can give details about the cause of the problem.

2.2. Tool Used

Selenium is an open source automation testing tool for web based applications. It runs directly on browser and supports almost all available browsers such as Firefox, chrome, IE, Opera, Safari etc. It runs on all platforms such as Windows, Linux and Macintosh. It's a very useful tool for System functional testing and browser compatibility testing.

Selenium is a browser automation tool, commonly used for writing end-to-end tests of web applications. A browser automation tool does exactly what you would expect: automate the control of a browser so that repetitive tasks can be automated. At a very high level, 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 Web Driver, 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. Within the project, they are referred to as “IDE”, “Web Driver” and “Grid”. Selenium is a browser automation tool which lets you automated operations like: type, click, and selection from a drop down of a web page.

III RELATED WORK

3.1 A Test Automaton Framework Based on WEB

In the year 2012, Fei Wang and Wencaai Du [1] explained about the design the new test automation framework. The test automation framework that was integrated by the two kinds of test automation tool Selenium and Jmeter. Jmeter developed by Apache organization is an open source pressure test tool based on Java. It can be used on the server, the network or other object to simulate huge loading, in different pressure under a variety of testing their strength and analyze the overall performance.

Step 1: The information about the same type of web applications abstracts way from the web-based test automation framework and form into a single configuration file.

Step 2: Testing the framework

The new testing automation framework integrated by Selenium and Jmeter can share the test steps and test data among different testing, such as UI testing, backend testing, loading test and so on. It is convenient to switching in various types of testing for web applications. It supports multiple browsers and a variety of operating system. It can be widely used in web application test automation.

3.2 Design and Implementation in Selenium IDE with Web Driver

In the year 2012, NidhikaUppal, Vinay Chopra [2] analysis of selenium IDE with selenium Web Driver and finally, enhance the Selenium IDE with using Web Driver and compatible with others browser. Now days the Web applications being developed to be compatible with all browsers but in Selenium IDE tool is limited to Firefox browser only and Selenium IDE cannot be tested on all browsers .If users wants to run recorded tests in different browsers like IE or Chrome then Selenium tools web driver has features to support with others browser .Integrating Selenium IDE and web driver in one single package so that recorded tests on IDE can be run as web driver tests from single UI [2].

Step 1: Integrating Selenium IDE and Web Driver in one single package so that recorded tests on IDE can be run as web driver tests from single UI.

Step 2: Selenium Web Driver and Changes done to achieve running test cases in IE or Google Chrome from Selenium IDE.

- Multi-browser testing including improved functionality for browsers not well-supported by Selenium IDE.
- Handling multiple frames, multiple browser windows.
- And the code that calls Batch file to execute scripts:

Step 3: Implementation: Clicking on selenium IDE window.

Running and recording testing scripts in Selenium IDE with others browser like IE, Chrome and it can only possible with web driver and it also improved functionality of browsers.

3.3 Implementation of Selenium with JUNIT and Test-Ng

In the year 2012, Deepti Gaur, Dr. Rajender Singh Chhillar [3] implements selenium with different frameworks that is, junit and testng. To create and run integration tests with Selenium, you must complete the following steps:

- Use the Selenium IDE to record and play tests.
- Export tests created with the Selenium IDE as JUnit tests.
- Add the JUnit tests to your Java project in your IDE and run the tests.

A good test case is one that has high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error. Test Cases in Selenium are nothing but recording the Web Application and testing that again using the Selenium tool. The IDE allows many options for running your test case. You can run a test case all at once, stop and start it, run it one line at a time, run a single command you are currently developing, and you can do a batch run of an entire test suite. Execution of test case is very flexible in the IDE. JUnit 4 and TestNG are both very popular unit test framework in Java. Both frameworks look very similar in functionality.

3.4 Data Driven Automation Testing of Web Application Using Selenium

In the year 2011, Navaraj Javvaji, Anand Sathiyaseelan, Uma Maheswari Selvan, [4] the goal is to set forth the approach for “Automating the Web Applications Using SELENIUM RC. The Data driven testing is creation of test scripts to run together with their related data sets in the framework. The main advantage of the automated tests is the reusability and also the maintenance of these tests is easy. This requires the preparation of the data sheets which is completely independent of the test automation tool. Data driven framework contains some main components, they are as follows.

- a) **ANT Script:** A Script that is used to execute a set of instruction using build.xml. It is used to run single test or batch test. Single test could be used to execute a single test suite while batch test can be used to run multiple test suites.
- b) **Reusable Library:** It consists of the common functions that can be reused. It is a collection of utilities that deliver the capabilities to create detailed drill down reports in different formats, help log events and save snapshots of tests that fail
- c) **Library Files:** Library files consist of the major supporting libraries files which are necessary for the automation. Ex: selenium-server.jar
- d) **Logger:** Logger is mainly designed to help the user to view the step by step execution of the test scripts along with the time stamp, execution status whether it passed or failed.
- e) **Test Data:** Test Data contains the excel workbooks with test data sheets holding the input values to the application under test.
- f) **Test Report:** The test report is generated by ANT at the end of the test execution. The test report displays the overall execution status.

IV EXISTING SYSTEM

4.1 Data Driven Testing

The Data driven testing is creation of test scripts to run together with their related data sets in the framework. The framework contains some main components, they are ANT script, Reusable Library, Library Files, Logger, Test Data, Test Report, Test Screens, Test Scripts, Test Suite.

4.2 Selenium Webdriver Workflow



Step 1: Read the spreadsheet and select the first row.

Step 2: Store all the information about the test suite in the temporary variables.

Step 3: Choose function/test script from the function column and search it in the libraries.

- If not found generate error.
- Else, pass the parameters from same row to the function.

Step 4: Execute Function.

- Take screen shots and store in reports.
- Store Reports in HTML format
- Create Log files.
- Store report in excel/ sheet/spreadsheet(pass/fail/status)

Step 5: Repeat 2 to 4 for all rows in excel sheet.

Step 6: Generate final report and store it.

Step 7: Generate necessary report links in the spreadsheets.

V IMPLEMENTATION & RESULT

System Specifications

Selenium webdriver, java, TestNG, JUnit, JMeter;

Selenium WebDriver makes direct calls to the browser using each browser's native support for automation. There are so many browsers and many programming languages there is need for common specification provided by WebDriver API.

TestNG is a testing framework developed in the lines of JUnit and NUnit, however it introduces some new functionalities that make it more powerful and easier to use. TestNG is an open source automated testing framework. TestNG is similar to JUnit (especially JUnit 4), but it is not a JUnit extension. It is inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes. TestNG can generate reports based on our Selenium test results.

Result - Test Report

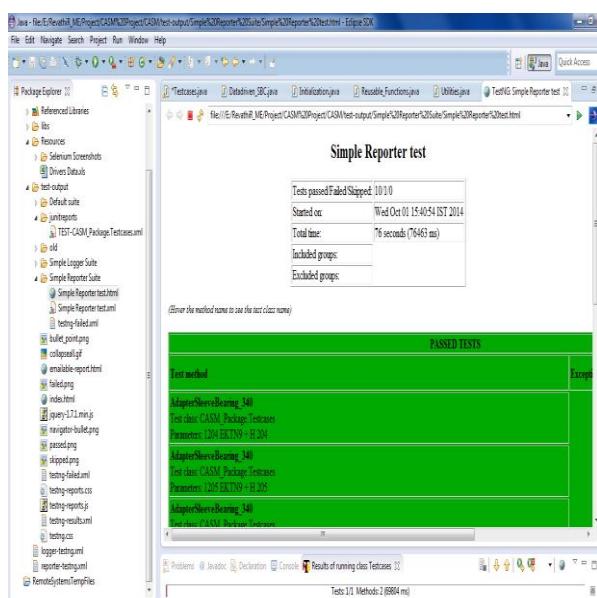


Fig: Test Report

VI PROPOSED WORK -- INTEGRATION OF SELENIUM WITH JMETER

In proposed work ,integrated by Selenium and Jmeter. It can share the test steps and test data among different testing, such as UI testing, loading test and so on. It is convenient to switching in various types of testing for web applications. It supports multiple browsers and a variety of operating system. It can be widely used in web application test automation.

Jmeter:

Jmeter is an Apache jmeter that can be used as a load testing tool for analyzing and measuring the performance of a variety of services, with a focus on web applications. It can be used to simulate a heavy load on server, network or object to test its strength or to analyze overall performance under different load types.

Key Features of JMeter

1. Cross-platform. JMeter can be run on any operating system with Java.

2. Scalable.
3. Multi-protocol support. Multiple implementations of pre and post processors around sampler. This provides advanced setup, teardown parametrization and correlation capabilities
4. Various assertions to define criteria
5. Multiple built-in and external listeners to visualize and analyze performance test results

Graph Explanation

Throughput is calculated as requests/unit of time. The time is calculated from the start of the first sample to the end of the last sample. This includes any intervals between samples, as it is supposed to represent the load on the server. The formula is: Throughput = (number of requests) / (total time).

Graph Result

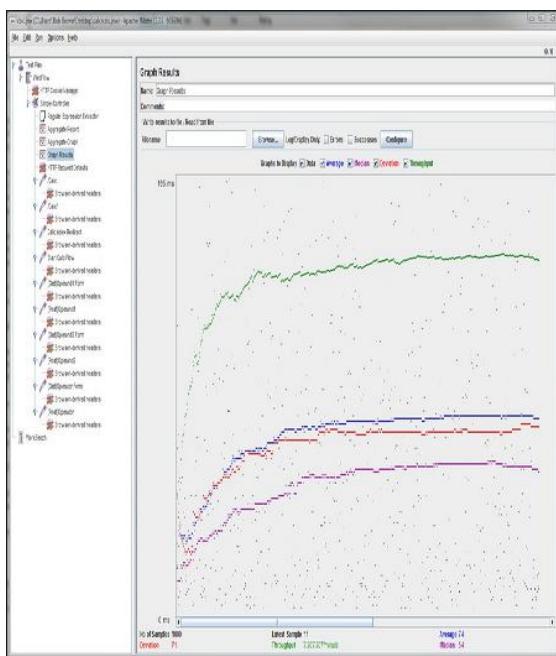


Fig: JMeter

VII. CONCLUSION

Selenium is open source software and also a web application testing tool. In a very short span of time, Selenium automation testing framework is gaining wide acceptance as a popular and successful mode of website automated testing. Has great flexibility and extensibility. Along with its tight integration with the browser unmatched by available proprietary tools. Today everything is web based thus it is becoming more and more complex. For this, huge information platform, quick release cycle as well as quick regeneration is required. This requires the web application to be comprehensive, expansibility and efficiency. Load testing was completed after integrating selenium webdriver and JMeter.

REFERENCES

- [1] Fei Wang and Wencaai Du, "A Test Automation Framework Based on WEB" proc. IEEE 11th International Conference on Computer and Information (ACIS 12),IEEE Press, 2012, pp. 683-687, doi:10.1109/ICIS.2012.21
- [2] 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.
- [3] Deepti Gaur, Dr. Rajender Singh Chhillar, " Implementation of Selenium with JUNIT and Test-Ng", IJCSMS International Journal of Computer Science and Management Studies, Vol. 12, Issue 03, Sept 2012.
- [4] Navaraj Javvaji, Anand Sathiyaseelan, Uma Maheswari Selvan, "Data Driven Automation Testing of Web Application Using Selenium", Conference Proceedings, Step-Auto 2011.
- [5] Ms. Rigzin Angmo, Mrs. Monika harma, "Selenium Tool: A Web based Automation Testing Framework " International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), 2014.
- [6] Chandraprabha, Ajeet Kumar, Sajal Saxena, " SYSTEMATIC STUDY OF A WEB TESTING TOOL: SELENIUM" International Journal Of Advance Research In Science And Engineering ,IJARSE, Vol. No.2, Issue No.11, November 2013.
- [7] Rasul Niyazimbetov, "Web application testing solutions with selenium".
- [8] McMahon, C., History of a Large Test Automation Project Using Selenium 2009,8
- [9] Sherry singla, Harpreetkaur, "Selenium keyword automation testing framework", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4,2014.
- [10] Y.C. Kulkarni, Y.C. Kulkarni, "Automating the web applications using the selenium RC", ASM's International Journal of Ongoing Research in Management and IT e-ISSN-2320-0065, 2011.
- [11] Monika Sharma and Rigzin Angmo, "Web based Automation Testing and Tools", international journal of Computer Science And Information Technology (IJCST), Vol. 5(1), 2014, ISSN:0975-9646, pp. 908-912.
- [12] Anuja Jain, M.S. Prabu, Swarnalatha, "Web-Based Automation testing Framework", International Journal of Computer Applications ,Vol 45, pp1-5, 2012.
- [13] Gaurav Saini, Kestina Rai, " Software Testing Techniques for Test Cases Generation" , International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, 2013.
- [14] Gowri Shankar Palani, Summary of web application testing methodologies and tools <http://www.ibm.com/developerworks/web/library/wa-webapplicationtesting/> IBM 2011 (January 10, 2013)
- [15] http://seleniumhq.org/docs/05_selenium_rc.htm
- [16] Open QA: Selenium, <http://www.openqa.org>.
- [17] <http://www.methodsandtools.com/tools/tools.php?jmeter>
- [18] <http://jmeter.apache.org/>
- [19] <http://www.w3schools.com/xsl/>