Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 88 (2015) 36476-36478

ISSN: 2229-712X

# Data Driven Automation Testing Framework Using Selenium Web driver

Vina. M. Lomte, Rishikesh Chandra, Ayush Gondhali, Ashish Shinde and Sanket Pimple
RMD Sinhgad School of Engineering, Pune-58, Pune University, Maharashtra, India.

**ARTICLE INFO**

**Article history:**
Received: 02 April 2015;
Received in revised form:
15 November 2015;
Accepted: 20 November 2015;

**Keywords**
Selenium Web driver,
Automation, Data Driven
Driver Script, Test Case.

**ABSTRACT**
The growing importance of quality software systems has brought about increasing demand for efficient software testing. Traditionally, there is a typical approach for testing the software manually, which consumes a lot of time and effort. Hence there is a need to decrease the amount of resources needed and to increase the efficiency and throughput. One attractive solution to this problem is test automation, i.e. allocating certain testing tasks to computers. This thesis opens up the discussion on test automation framework made by us. Here we have used Selenium Web driver, Java, HTML and other necessary languages and packages for implementing the framework. The framework uses the Driver Script and allows to automatically fetch data from the spreadsheets and uses them in testing the web application automatically and also generating report automatically for the same.



## Introduction

The abrupt increase in the growth of Software Industry has brought numerous web applications in the market and these applications have an enormous amount of users. Hence, the correctness of these applications is very much necessary.

Testing of 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 Automation in testing can bring a revolutionary change and can help in saving a lot of time and resources.

Our Automated Testing Framework will automatically take the test cases and accordingly select the testing functions/scripts and simultaneously generate a Log report, HTML report, and screenshots of each instance of testing and finally generating a complete HTML report on tested application/software, after comparison of the actual result with the expected result.

## About The Framework

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 Spread sheet and automatically start the testing process.

The Framework can be used from an executable file or as a library imported in a Java test case.

We can stop or pause the testing anytime. The testing process will stop automatically after finishing the test which also includes generation of success/failure screenshots, HTML reports, Log report and writing Passed/Failed/Skipped status in the reports and the spreadsheet.

An integral part of the test execution is the verification of the test results. The framework uses a logic based on tester's decision of success of a test step, to decide whether a test has passed or failed.

## Tools Used

Selenium Web-driver is an open source testing tool for web browser automation. It supports the web browsers such as Internet Explorer, Mozilla Firefox, and Google Chrome etc. It uses various Libraries such as Apache POI for reading and writing data to and from excel files, Java Logger for logging services. Selenium Web-driver has been used for building the whole framework in eclipse IDE. The log file generator has been developed using Java Logger customized to general needs. For screenshots a simple Robot method has been used.

## Working of Framework

The basic data which will be needed while the execution of the test is provided in the excel sheets. There exists one spreadsheet called Environmental file along with the framework. It contains Global variables like Release version of application to be tested and various parameters such as application link, application username, application password database source name, database username and database password. The only fixed parameter is the release version which has to be set by Automation Tester, and other parameters may vary according to application needs and can be extended as needed.

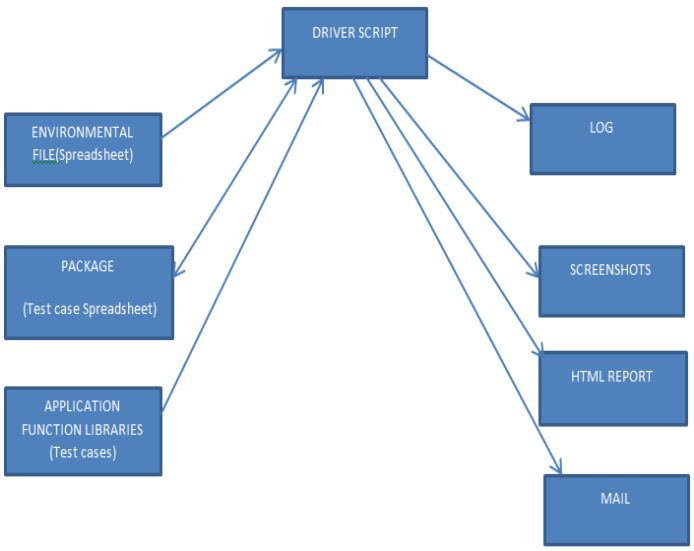



**Figure 1. Work Flow Diagram**

The above diagram shows the working and organization of different components involved in the framework.

The architecture as shown above in Figure 2 shows the structure of the framework involving the Selenium Web driver as the main block for testing since it automates the browser and its components as needed to test the Web App.

Tele:
E-mail addresses: rishirebel@gmail.com

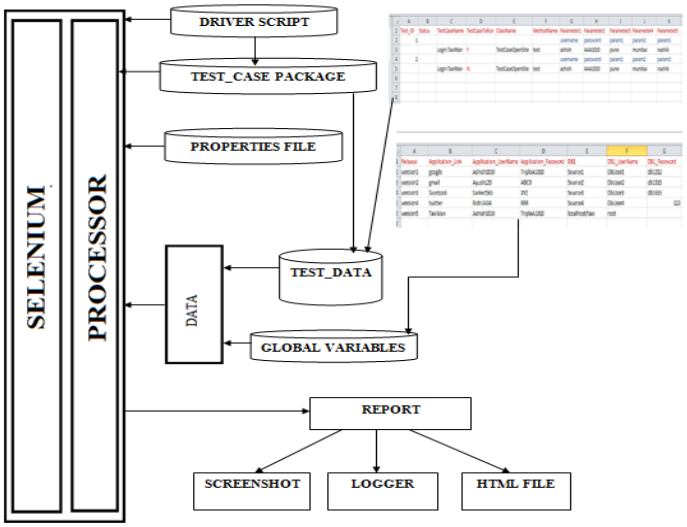
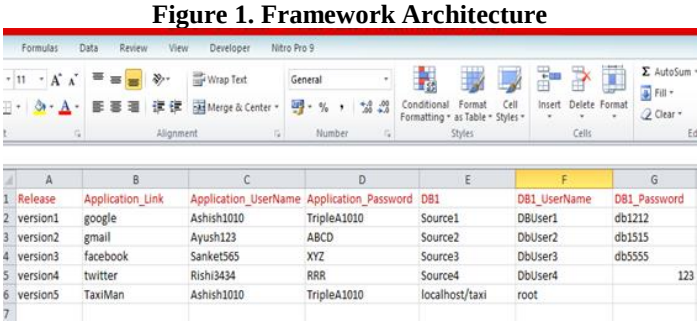


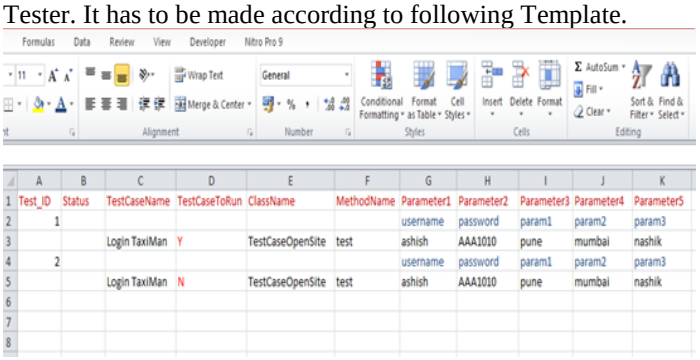

**Figure 1. Framework Architecture**



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Release | Application_Link | Application_UserName | Application_Password | DB1 | DB1_UserName | DB1_Password |
| 2 | version1 | google | Ashish1010 | TripleA1010 | Source1 | DBUser1 | db1212 |
| 3 | version2 | gmail | Ayush123 | ABCD | Source2 | DbUser2 | db1515 |
| 4 | version3 | facebook | Sanket565 | XYZ | Source3 | DbUser3 | db5555 |
| 5 | version4 | twitter | Rishi3434 | RRR | Source4 | DbUser4 | 123 |
| 6 | version5 | TaxiMan | Ashish1010 | TripleA1010 | localhost/taxi | root | |
| 7 | | | | | | | |

**Figure 2. Environmental File**

The tester has to create another spreadsheet in a fixed directory which contains the names of all test cases and methods to be invoked along with parameters required for same. These methods are the test cases which have to be prewritten by the Tester. It has to be made according to following Template.



| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Test_ID | Status | TestCaseName | TestCaseToRun | ClassName | MethodName | Parameter1 | Parameter2 | Parameter3 | Parameter4 | Parameter5 |
| 2 | 1 | | | | | | username | password | param1 | param2 | param3 |
| 3 | | | Login TaxiMan | Y | TestCaseOpenSite | test | ashish | AAA1010 | pune | mumbai | nashik |
| 4 | 2 | | | | | | username | password | param1 | param2 | param3 |
| 5 | | | Login TaxiMan | N | TestCaseOpenSite | test | ashish | AAA1010 | pune | mumbai | nashik |
| 6 | | | | | | | | | | | |
| 7 | | | | | | | | | | | |
| 8 | | | | | | | | | | | |

**Figure 3. Test data Spread sheet**

The tester must set the values of release version to be tested and file name of above spreadsheet in a properties file given in framework.

**The Framework works in following steps:**

1. Driver script fetches rows of Environmental file and compares its release version with that given in properties file.
2. Driver script generates Environment hash table for that row.
3. Now it reads spreadsheet whose name is given in properties file.
4. Now it generates Test Data hash table for a row and invokes the given test case.

(The Tester may include the Framework's methods to generate HTML report, create Logs and store screenshots according to his will.)

5. The Logs, Screenshots and HTML reports are generated in respective directory.

(Steps 4 and 5 repeat for every test case given in the spreadsheet.)

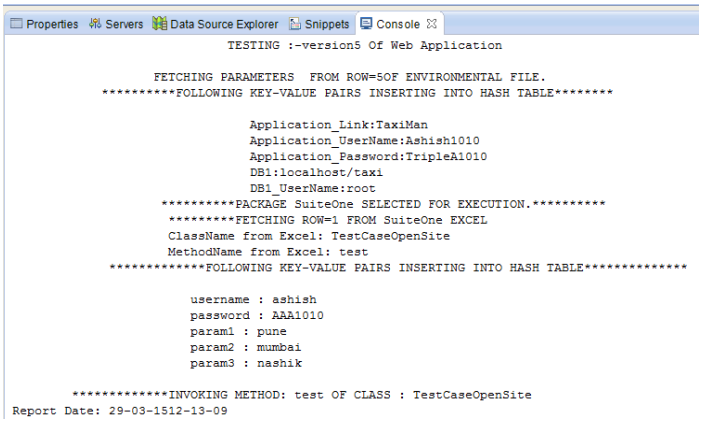



**Figure 4. Driver Script Running in Eclipse**

Following is the Screenshot of the Web app as an example, under test using the Framework.

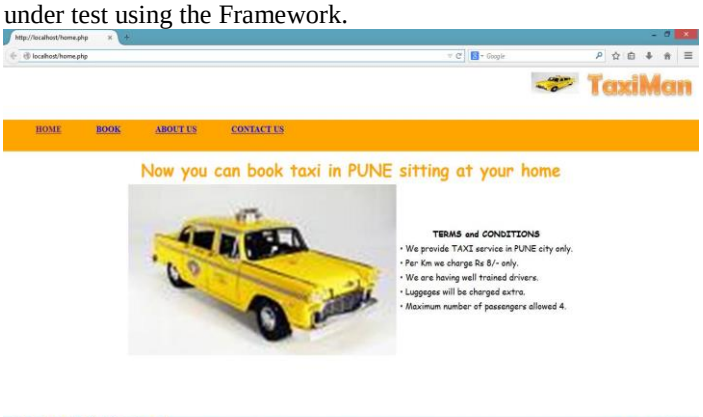



**Figure 5. Automatic opening of webpage**

As specified by tester in test case, selenium conducts automation of browser for the web app. The test data required in test case is fetched using Framework methods. Also, after this execution some of the required data is written on the excel sheet for example in the Status (pass/fail) column of the excel sheet.

**Following are the screenshots of the output of a test case**

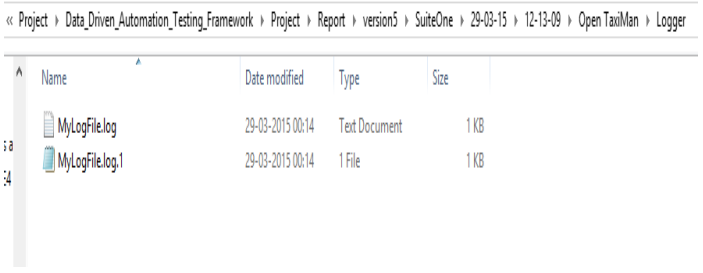



**Figure 6 Directory of Log files**

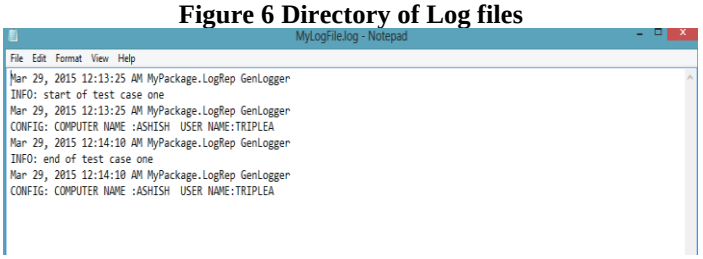



**Figure 7. Log report generated**

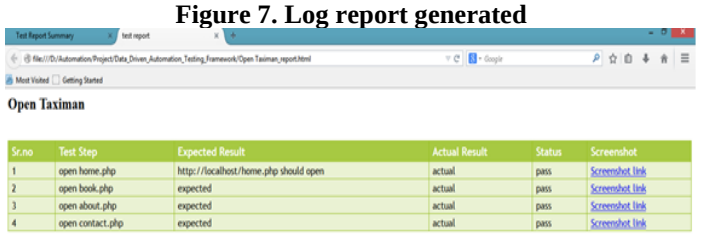



**Figure 8. The HTML Report**

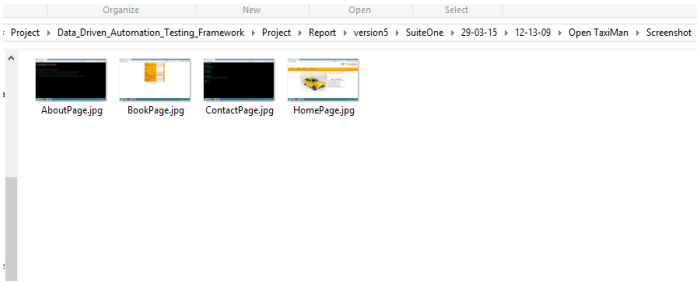



**Figure 9. Screenshot Directory**

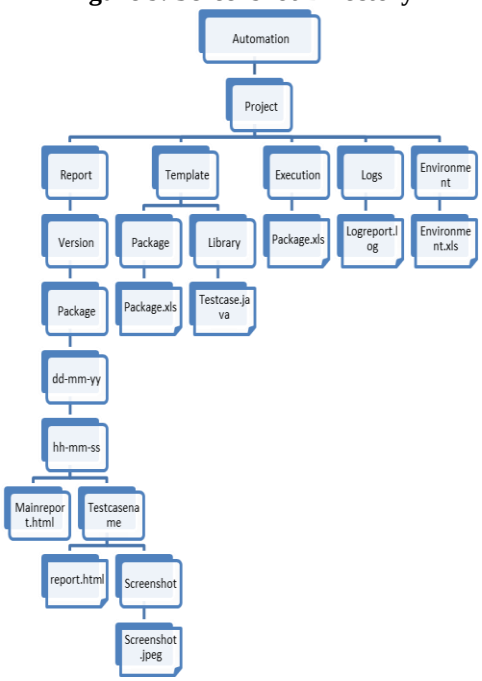



**Figure 10. Directory Hierarchy of the Framework**

The files created and used by the framework are organized in the above shown manner. The well-organized hierarchy allows the tester to manage a plethora of reports generated as a result of testing large projects. The report folders are generated dynamically as needed and the logs, screenshots and the HTML reports are stored in the respective folders.

**Acknowledgment**

We are extremely thankful to Prof. Deepak Rewadkar, Head of Department, Division of Computer Engineering, RMDSSOE for permitting us to undertake this work.

We express our heartfelt gratitude to our respected Project guide Prof. Vina M. Lomte for her kind and inspiring advice which helped us to understand the subject and its semantic significance. She enriched us with valuable suggestions regarding our topic and presentation issues.

**Conclusion**

A test automation framework can be built around model based testing that begins with automating the test design exercise and the development of manual test procedures. It further uses automation frameworks to attain modularity in automated test script design, and integrates with test execution tools for automated test execution.

**Such a framework will help test organizations to:**

Greatly scale when test designers and test automation developers are in short supply by training regular testers in model-based testing and test automation Deliver test automation quickly so that regression cycles can be automated right from the first cycle, and save effort and cost that would otherwise have been expended on manual testing.

**References**

[1].Wikipedia: Test Automation http://en.wikipedia.org/wiki/Test_automation
[2]. The History of Software Testing by Joris Meerts http://www.testingreferences.com/testinghistory.php
[3]. Busy Developers' Guide to HSSF and XSSF Features https://poi.apache.org/spreadsheet/quick-guide.html
[4].Java Logging Tutorial: http://www.vogella.com/tutorials/Logging/article.html
[5].Java I/O Manual https://docs.oracle.com/javase/tutorial/essential/io/
[6].Robot for Screenshot http://stackoverflow.com/questions/4490454/how-to-take-a-screenshot-in-java
[7].Java Robot Class http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html
[8]. Data Driven and Keyword Driven Test Automation Frameworks by Pekka Laukkanen http://eliga.fi/Thesis-Pekka-Laukkanen.pdf