

# Case Study of Selenium Web Driver and Comparability Study of Automation Testing Tools

Shweta Dwiedi<sup>1</sup> Pradeep Kumar<sup>2</sup>

<sup>1</sup>M.Tech Student <sup>2</sup>Assistant professor

<sup>1,2</sup>NIET College of Engineering, Greater Noida India

**Abstract**— Software Testing is a practice to ensure the quality as well as the finding bugs. It is a time consuming and difficult process. Because now a days web application become more complex. Testing is often testing skipped by Test Engineer due to this reason. It became more complex to ensure the quality of software. So to overcome this problem we use automation testing tools. So Test automation can help to avoid this situation. Automation testing is use the automation tool to reduce the human efforts as well as Manual Testing. We can avoid the repetitive work or redundant works. The main Objective of this thesis is to use Automation Testing Tool “Selenium Web driver” for web applications. With this web testing tool, I have developed data driven framework. I have developed that means instead of writing multiple functions to automate website, we have abstracted those functions to excel files and in that excel file. By this way web application is automatically tested by taking the data from separate file using data driven framework.

**Key words:** Selenium Web driver, Automation Testing Tools

## I. INTRODUCTION

Automation Testing is a mythology by which we can test our application using tools. This testing is use of tools and reduces the need of manual or human interaction, redundant or repetitive work Selenium is a one of the automation tool designed to automate the web application using programming language. This tool designed to generate automated tests and enhance the testing performance. Software developer use automation testing to save time and resources .Selenium is an open source automation testing tool for web based applications. It runs directly on browser. It 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 is very useful tool functional testing and browser compatibility testing. It is really strong as compare to other automation tools .It is very flexible and simple to use. There are many languages supported in Selenium. It use independent language it means that your web application is developed in java language and if you want to automate this application then you can use any language. So working with selenium knowledge of any one programming language is needed. Selenium is a browser automation tool, commonly used for writing end-to-end tests of web applications. Selenium is a suite of three tools. The first of these tools is Selenium IDE, It is an extension for Firefox that allows users to record and playback tests. But it is limited we cannot use it for many users, so the second tool in the suite is Selenium Web Driver; It provides APIs with a multiple languages to allow for more control of application. It is Selenium Grid, which specializes in running multiple tests across multiple browsers and operating systems, and

machines in parallel. So they are called as "IDE", "Web Driver" and "Grid".

## II. PROPOSED WORK

In this section we will explain about automation tool and design of Test Cases using core java language for IRCTC web application.

Here we will use selenium web drive with TestNG unit testing framework. TestNG provide the rich set of annotations and also help in generate the reports.

In this paper we will explain about Selenium IDE, Web driver and Comparability study of Automation Testing Tools. On the basis of case study by taking an example of IRCTC Web application we will discuss about Web Driver Testing tools.

With the use of Selenium Web Driver, I can automate the following test scenarios for IRCTC web site.

On the basis of test cases we can automate IRCTC site by using selenium web driver.

Note:

- All the CAPTCHA Fields have to be filled Manually
- After Searching trains all fields and links are generated dynamically so you have to inspect the field every time.
- I have wrote this code to book ticket between NDLS(New Delhi) to SBC (Bangalore) Journey date May,15,2017
- You cannot inspect element by right clicking on element,

```
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import com.google.common.base.Predicate;
import com.thoughtworks.selenium.Wait;
public class Irctc { public static void main(String[] args)
throws InterruptedException {
WebDriver driver=new FirefoxDriver();
driver.navigate().to("https://www.irctc.co.in"); // IRCTC
Website
URLdriver.findElement(By.cssSelector("#usernameId")).se
ndKeys(""); //Enter USER NAME
```

```
driver.findElement(By.cssSelector("#loginFormId >
div.grid_16.alpha.g_box > div.grid_16.alpha > table:nth-
child(1) > tbody > tr:nth-child(2) > td:nth-child(2) >
input")).sendKeys(""); //Enter Password
Thread.sleep(10000); //Used to Enter CAPTCHA Value
manually (Can't be auto filled).
driver.findElement(By.name("submit")).click(); //Clicks on
SUBMIT Button Automatically.
```

- press F12 and do it manually.
- This code will only take you to payment gateway after that you have to handle it manually.

#### A. Implementatin of Code

```
webElements.sendKeys(""); //Enter "From" Station Code.
(Example "NDLS" for "New Delhi").
webElements.sendKeys(Keys.TAB);
Thread.sleep(1000);
WebElement webElements1=
driver.findElement(By.id("jpform:toStation"));
webElements1.sendKeys(""); //Enter "to" Station Code.
(Example "SBC" for "Bengaluru").
Thread.sleep(1000);
driver.findElement(By.id("jpform:journeyDateInputDate")).
sendKeys(""); //Enter Date Of Journey
Thread.sleep(3000);
driver.findElement(By.id("jpform:jpsubmit")).click(); //Click
s on SUBMIT Button Automatically.
Thread.sleep(3000);
driver.findElement(By.id("clink-12034-CC-
1")).click(); //Clicks on Class Automatically to show seat
availability.
Thread.sleep(3000);
WebDriverWait wait = new WebDriverWait(driver, 10);
WebElement element =
wait.until(ExpectedConditions.elementToBeClickable(By.cs
sSelector("#c1 > div:nth-child(1) > div:nth-child(2) >
table:nth-child(1) > tbody:nth-child(1) > tr:nth-child(2) >
td:nth-child(5) > a:nth-child(2)")); //Waiting for "Book
Now" link to be generate.
driver.findElement(By.cssSelector("#c1 > div:nth-child(1) >
div:nth-child(2) > table:nth-child(1) > tbody:nth-child(1) >
tr:nth-child(2) > td:nth-child(5) > a:nth-
child(2)")).click(); //Clicks on "Book Now" link.
Thread.sleep(3000);
driver.findElement(By.className("psgn-
name")).sendKeys(""); //Enter Passenger Name.
Thread.sleep(3000);
driver.findElement(By.cssSelector("#addPassengerForm\\:p
sdetail\\:0\\:psgnAge")).sendKeys(""); //Enter Passenger
Age.
Thread.sleep(3000);
```

```
WebElement
webelements5=driver.findElement(By.cssSelector("#addPas
sengerForm\\:psdetail\\:0\\:psgnGender"));
Thread.sleep(3000);
webelements5.sendKeys(""); //Enter Passenger Gender.
Thread.sleep(3000);
webelements5.sendKeys(Keys.TAB);
Thread.sleep(3000);
WebElement
webelements4=driver.findElement(By.cssSelector("#addPas
sengerForm\\:psdetail\\:0\\:berthChoice"));
Thread.sleep(3000);
webelements4.sendKeys(""); //Enter Seat of your choice.
Thread.sleep(3000);
webelements4.sendKeys(Keys.TAB);
Thread.sleep(3000);
WebElement
webelements3=driver.findElement(By.cssSelector("#addPas
sengerForm\\:psdetail\\:0\\:foodChoice));
Thread.sleep(3000);
webelements3.sendKeys(""); //Enter Meal Type.
Thread.sleep(3000);
webelements3.sendKeys(Keys.TAB);
Thread.sleep(3000); //Enter CAPTCHA Value Manually.
driver.findElement(By.id("addPassengerForm:mobileNo")).sendKe
ys(""); //Enter Passenger Mobile No.
Thread.sleep(3000);
driver.findElement(By.id("validate")).click(); //Clicks on "Make
Payment" button.
Thread.sleep(3000);
driver.findElement(By.id("CASH_CARD")).click(); //Select
Payment Option.
Thread.sleep(3000);
driver.findElement(By.cssSelector("td.CASH_CARD > table:nth-
child(1) > tbody:nth-child(1) > tr:nth-child(2) > td:nth-child(2) >
input:nth-child(1)")).click(); //Select Payment Gateway.
Thread.sleep(3000);
driver.findElement(By.cssSelector(".prefdBankOpt > input:nth-
child(2)")).click(); //Clicks on "Add as preferred bank" Option.
Thread.sleep(3000);
driver.findElement(By.id("validate")).click(); //Clicks on payment
button.
}
}
```

### III. AUTOMATED TESTING TOOLS

For the selection of right automated software testing tool, it is important to create a list of requirements to view when choosing a tool for evaluation. If we do not have a list of requirements, we may waste time for downloading,

installing and evaluating tools that only meet some of requirements, or may not meet any of them. This study evaluate four major tool vendors that are Selenium, Soap UI, HP Unified Functional Testing, and Test Complete on their test tool characteristics, test execution capability, test reporting capability, scripts reusability capability, play back capability etc.

Because of the more advantages of the automation testing over manual testing, various companies are engaged in developing various automated test tools for various applications on the basis of the license associated with testing tools we can categorized these automated testing tools as follows:

- 1) Open source test tools
  - 2) Commercial test tools
- 1) *Open Source Test Tools:*

These testing tools however may not require purchase of license and the code of the application is available to the user for further modifications to be performed. These test tools are free for the users to use. It can be downloaded from the internet or can be obtained by the vendor without any charges e.g. Selenium, SoapUI.

- 2) *Commercial Test Tools:*

It includes those testing tools which are closed source in nature and license has to be purchased so as to harness their functionalities to full extent. These tools are the commercial software for sale. User should pay for it to use the software. Costs may be as per the functionality of the test tool. Example under this category is HP Unified Functional Testing (UFT) and Test Complete.

#### A. Selenium

Selenium is a one of the efficient open-source automated testing tool which provide a nice testing framework for testing wide variety of applications exporting scripts in almost every language including java, .net, c#. The main feature of Selenium is multi browser supports for execution of test cases.

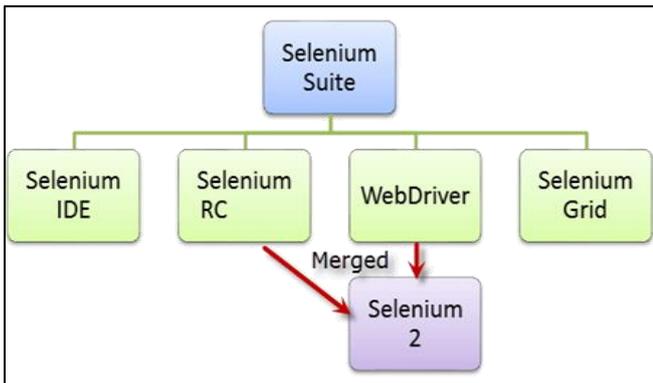


Fig. 1: Selenium Suite and Its Components

#### B. HP Quick Test Pro (QTP) or HP Unified Functional Testing (UFT)

HP UFT is a Functional testing tool which is best suited for regression testing of the applications. It is a licensed/commercial tool owned by HP, one of the most popular tools available in the market. It compares the actual and expected result and reports the results in the execution summary details.

It is an easy and extremely user-friendly tool that works well with Windows & Web based applications. It is a functional testing tool which has the feature for storing screenshot of each and every page navigated during the execution. So it can be used as a proof for completion of testing, and also you can refer the screenshots of previous executions if there is any need to refer them.

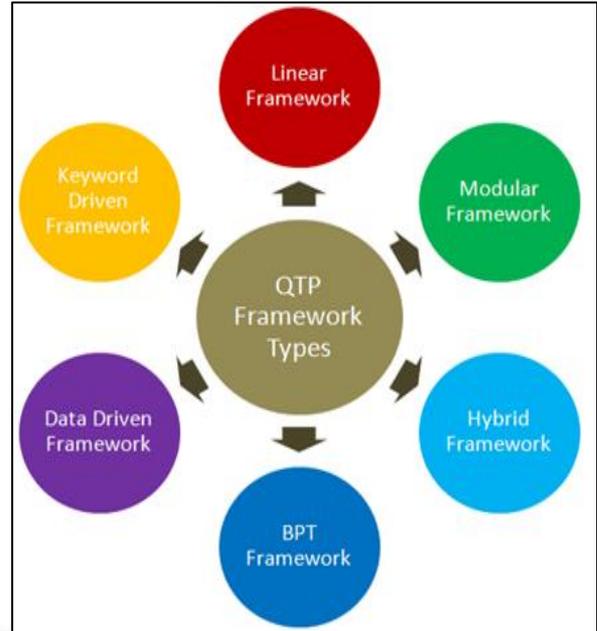


Fig. 2: QTP Framework

#### C. Test Complete

TestComplete is a functional automated testing tool developed by smart bear software that gives testers the ability to create automated tests for Microsoft Windows, Web, Android (operating system), and iOS applications.

TestComplete is used to create and automate many different software test types. Record and playback test creation records a tester performing a manual test and allows it to be played back and maintained over and over again as an automated test.



Fig. 3: Test Complete

#### D. SoapUI

SoapUI is an API testing tool that's both free, open source, and cross-platform. With an easy-to-use graphical interface, and enterprise-class features, it allows you to easily and rapidly create and execute automated functional, regression, compliance, and load tests. SoapUI is not just a functional API testing tool but also lets us perform non-functional testing such as performance and security test.

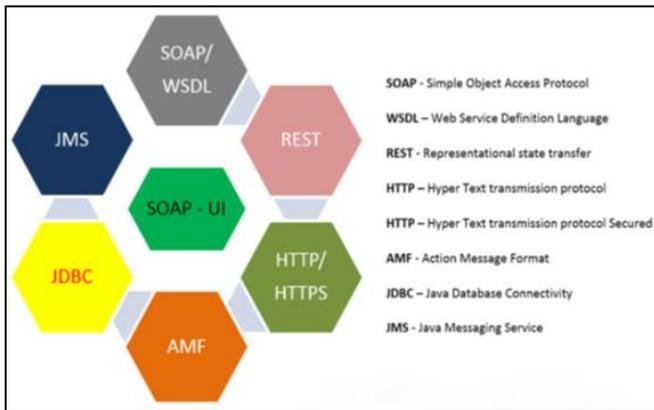


Fig 4: SOAP-UI

#### IV. EVALUATION STUDY

There are a number of open source and commercial web testing & window application tools available in the software market. Although the core functions of these tools are similar, but differ in functionality, features, usability. With above mentioned aspects, we have selected four functional testing tools for comparison which are Selenium, SoapUI, HP Quick Test professional/Unified Functional Testing and TestComplete.

For this comparative study we have used the current version of selenium that is 2.9.0, HP QTP/UFT 12.02, SoapUI 5.2.0 and current version TestComplete 11.0.

Comparison between these four tools is made on the basis of different parameters. These parameters can be Record-playback capability, Script generation capability, Script languages support, Application support, Technical support, Data-driven testing capability, Report generation capability, Debugging support, Easy to learn, License and Training cost etc. Table below list all evaluation parameters with the meaning of parameters.

#### V. CONCLUSION

One can select any one automation tool on the basis of type of application need to be tested, budget, and the efficiency required for testing the application. The requirement is fully filled by Selenium web driver then there is no need to go with QTP. If you don't want to spend money on testing tool then Selenium is a good automation tool. But on the basis of all things or as per conclusion QTP is the best automation tool.

#### ACKNOWLEDGMENT

I thank Mr. Pradeep Kumar, Prof. and Head, Dept. of M.tech, NIET College of Engineering, Greater Noida for his continuous support and encouragement for completing this research paper and also thanks to NIET management.

#### REFERENCES

- [1] Tarannam Bharti, Er. Vidhu Dutt, "Functionality Appraisal of Automated Testing Tools", IJCST Volume 3 Issue 1, Jan-Feb [2015]
- [2] Xinchun Wang, Peijie Xu, "Build an Auto Testing Framework Based on Selenium and FitNesse". International Conference on Information Technology

- and Computer Science, 2009. ITCS 2009. Page(s):436 - 439 ISBN: 978-0-7695-3688-0 IEEE 25-26 July 2009
- [3] Dubey, Neha, and Mrs Savita Shiwani, "Studying and Comparing Automated Testing Tools; Ranorex and TestComplete", IJECS 3.5 [2014]: 5916-23.
- [4] Harpreet Kaur and Dr.Gangan Gupta, "Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete", International Journal of Engineering Research and Applications, Issue 5, Sep-Oct [2013], pp.1739-1743.
- [5] RichaRattan and Shallu, "Performance Evaluation & Comparison of Software Testing Tool", International Journal of Information and Computation Technology. Volume 3, Number 7 [2013], ISSN 0974-2239, pp. 711-716