

Data Driven Testing Framework using Selenium WebDriver

Chandraprabha
Research Scholar
SRMSCET , Bareilly

Ajeet Kumar
Assistant Professor
SRMSCET , Bareilly

Sajal Saxena
Technology Analyst
Pune , India

ABSTRACT

Software Development Life Cycle is having various phases, among them software testing is a phase which is an ongoing process just start after the requirement gathering phase. Software testing have to face various challenges. Even though manual testing is an easy task but it is very time consuming and labor intensive process. Automation Testing is the solution of all the problems which are faced during manual testing. The aim of this research paper is to perform automation testing using “Selenium WebDriver”. With Selenium WebDriver I have developed data driven framework which means separating data to code for reusability purpose. In this framework we abstracted the data which would be used in excel file and the program is written to access data from that excel files.

Keywords

Software Testing, Automation Testing, Selenium WebDriver, Framework, Data Driven Framework

1. INTRODUCTION

Software testing is the very important phase of software development life cycle process almost 30-40% of total project's effort is invested in testing. The American novelist and historian Edward Eggleston wrote that “persistent people begin their success where others end in failure”. In software testing persistence is essential. Software testing is done to discover the errors not to conceal them. As web applications are becoming more complex testing process is also becoming complex and important, a professor from university of California William How den wrote that “testing is the unavoidable part of any responsible effort to develop a software system”. Software testing can be done either

manually or using any tool. As manual testing is labor intensive and time consuming process hence using an automated tool for software testing is a good solution. Selenium is an open source testing tool which is used to automate the test cases and enhance the testing performance. Selenium is an automated testing tool for web application. As web is growing testing of such web applications are becoming too more complicated. Testers have to face various challenges while testing such complicated web applications like multiple browser support, multiple platform support, handling wait conditions, handling flash objects, multiple language support etc. Selenium WebDriver ensure the tester to handle all such challenges. Selenium WebDriver basically work in two ways first locate the element and then perform some action on them. Selenium WebDriver locate element by using-id, name, Xpath, CSS, link text, partial link text. Selenium provides a rich set of functions which is used to testing of a web application.

2. PROPOSED WORK

In this paper we will explain about how we can design and use data driven automation framework in Selenium WebDriver using Java. Here with selenium WebDriver we will use TestNG unit testing framework. TestNG provides rich set of annotations and also able to generate reports. While testing an application several times it is required to test the same functionality with the different set of input data. In such scenario test data should not be embedded with test script for reusability purpose. Data is stored in some external files. External file may be xml, excel, database and csv. The test script is first connected with the external data source and then extract the data from that source. The benefit of this framework is that when we have to do the functionality testing with different dataset then we don't need to change our code. We have just to give the path of that data source which reduces our time and effort both.

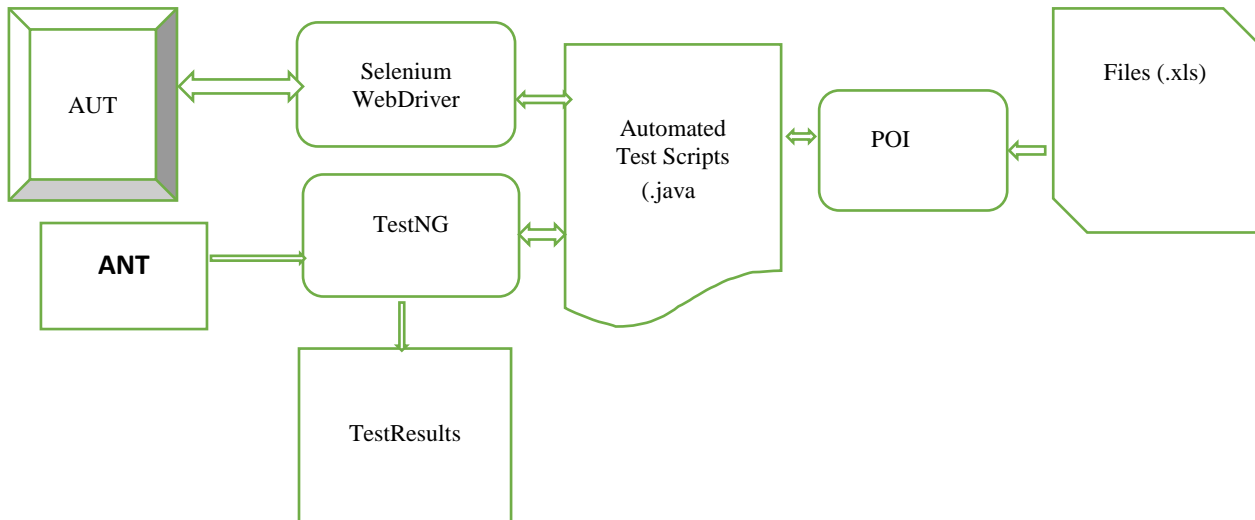


Fig1 Selenium Data Driven Architecture

In the above architecture (Fig 1) WebDriver interact with the AUT (application under test) which we have to test and first locate the elements which we have to test and then perform the operations. Here TestNG unit testing framework is used and also the benefit of TestNG with ANT is that it is able to generate HTML reports which could not be generated using WebDriver. Ant is a build tool which is able to generate HTML reports. Automated test scripts are written in java. Data is accessed from the excel file. To access the data from excel file Apache POI is used.

Apache POI is a project by Apache foundation which provides pure java libraries for reading and writing files in Microsoft Office formats. POI contains several sub

components but here we are using only HSSF and XSSF. HSSF stands for Horrible Spreadsheet Format which reads and writes Microsoft Excel format files. XSSF stands for XML spreadsheet format which is used to read and write Office Open XML format files

3. EXPERIMENTAL RESULTS

Data Driven framework works in two steps-

Step 1: The first step of data driven framework is to create an external file which stores the test data. Test data could be stored in Excel file, Xml file, Data base and CSV. In our project we are storing data in Excel file for user login.

User ID	Password
chandrprabhasaxena@gmail.com	128aspm
shikha123@gmail.com	piffi12344
abhiabc@gmail.com	1mhjagdjh23

Fig 2: Login Data

Step 2: Second step of data driven framework is to populate the data into automation test script. This step could be understood by following sub steps: First we have to import all the required packages, second sub step is to declare a class

third and most important sub step is to create a method which will read the data from excel and last sub step is make a user define function which will convert cell data into string and after this we can use this data for login purpose.

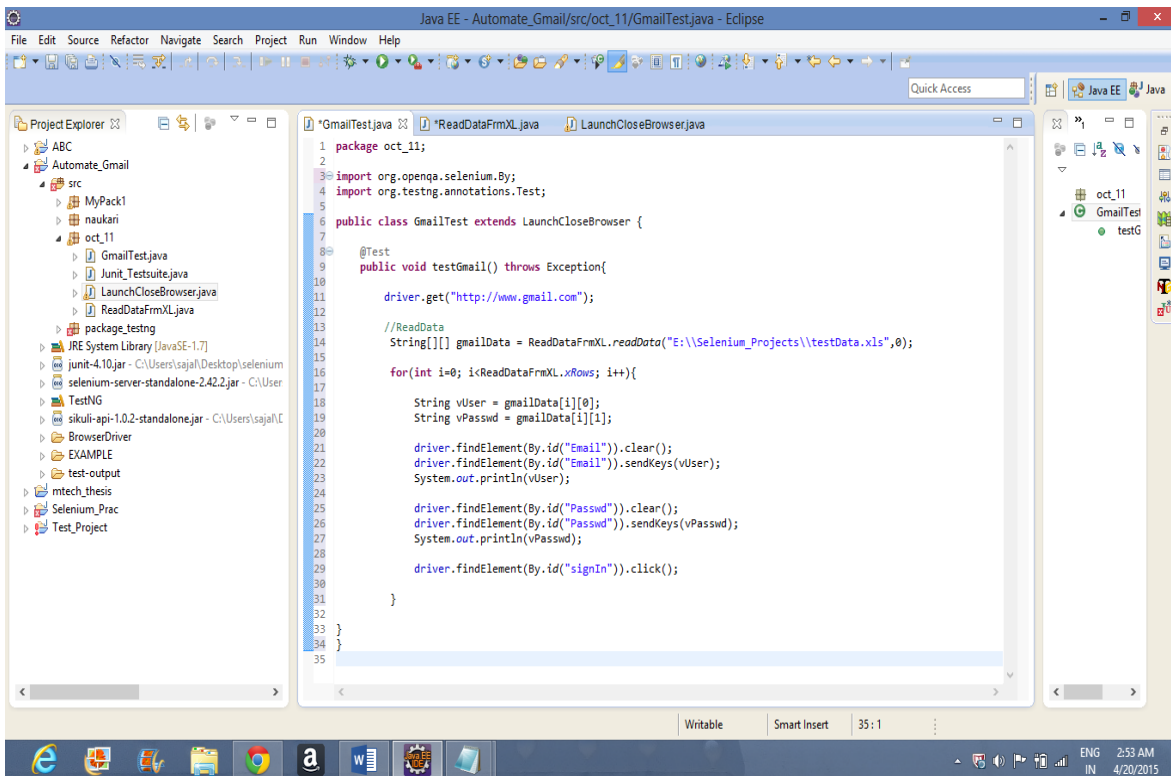


Fig 3: Logging into Gmail account Script

Here in fig 2 we are accessing Email id and password from the excel file which is stored in our machine at location E:\\Selenium_Projects\\testData.xls. Here we are using String [][] gmailData = ReadDataFrmXL.readData("E:\\Selenium_Projects\\testData.xls",0);

Here we are using two dimensional array of string type for storing userid and password. We are using readData() method

of ReadFmXML file by giving the location of our testdatafilepath and sheet no.

We have created a class ReadDataFrmXML which is reading the data from excel file and we are extending another class LaunchCloseBrowser where the basic method of Selenium WebDriver is written which include opening the browser maximizing it and path of browserdriver.

```
public class ReadDataFrmXML {  
  
    static intxRows;  
  
    static intxCols;  
  
    public static String[][] readData(String fPath,  
    intsheetNum)  
    throws Exception{  
  
        File file = new File(fPath);  
  
        FileInputStreamfIP = new FileInputStream(file);  
  
        HSSFWorkbookwb = new HSSFWorkbook(fIP);  
  
        HSSFSheet sheet = wb.getSheetAt(sheetNum);  
  
        xRows = sheet.getLastRowNum()+1;  
  
        System.out.println("Rows are :"+ xRows);  
  
        xCols = sheet.getRow(0).getLastCellNum();  
    }  
}
```

```
System.out.println("Columns are : " + xCols);

String[][] xData = new String[xRows][xCols];
for(int i=0; i<xRows; i++){
HSSFRow row = sheet.getRow(i);
for(int j=0; j<xCols; j++){
HSSFCell cell = row.getCell(j);
String value= CellToString(cell);
xData[i][j] = value;
System.out.println(value);
//System.out.print("##");
                                }
                                //System.out.println("@");
                                }
                                return xData;
                                }
}
```

Here apache POI will return cell type object which we have to convert into string using following:

```
public static String CellToString(HSSFCell cell) {
int type = cell.getCellType();
Object result = null;
switch(type){
case HSSFCell.CELL_TYPE_BLANK:
result = "";
//System.out.println("Blank Value");
break;
case HSSFCell.CELL_TYPE_BOOLEAN:
result = cell.getBooleanCellValue();
//System.out.println(result);
break;
case HSSFCell.CELL_TYPE_ERROR:
//System.out.println("There is some error.");
throw new RuntimeException("Error");
case HSSFCell.CELL_TYPE_FORMULA:
throw new RuntimeException("Formula can not be eveluated.");
case HSSFCell.CELL_TYPE_NUMERIC:
result = cell.getNumericCellValue();
//System.out.println(result);
break;
case HSSFCell.CELL_TYPE_STRING:
System.out.println(result);
result = cell.getStringCellValue();
//System.out.println(result);
break;
default:
System.out.println("Out of world.");
}
```

```
throw new RuntimeException("Out of world.");  
    }  
return result.toString();  
    }  
}
```

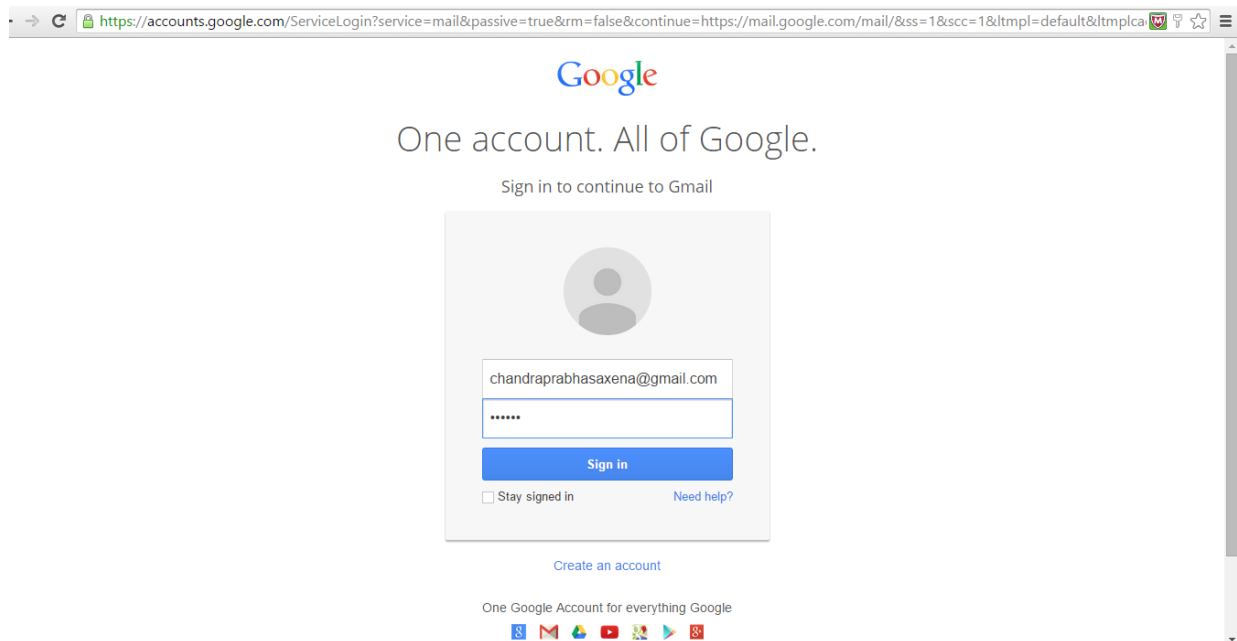


Fig 4: Logging into Gmail account

4. DATA DRIVEN FRAMEWORK ADVANTAGES

- The most important feature of this framework is that it provide the reusability and reduces the total number of scripts required to cover all the possible combinations of test scenarios. Hence we can cover the complete set of test scenarios in lesser amount of code.
- Second benefit of this framework is that if any change is done in test data matrix it will not affect the test script code.
- A single test scenario can be executed for different test data values.

5. CONCLUSION

Meaning of automation is not just to automate the test cases using the tool even it is much more. While automating a web application various challenges may be faced by the tester as now a days web application has become more complex and feature rich. Selection of a good framework is also a problem while automating an application. In this paper we have created data driven framework to perform Automation testing forweb application using Selenium WebDriver. Here we are using excel files to access the data. Here we will use TestNG unit testing framework to generate the reports. Data Driven framework is a good solution while we are having a big data set and it is time saving and effort saving solution.

Future work will use data driven framework with keyword driven framework to develop the domain specific -

requirement basedknowledge repository. Knowledge repository contains domain specific keywords that will help in faster execution of the keywords and save lot of time and project budget.

6. REFERENCES

- [1] Innovative approaches of automated tools in software testing and current technology as compared to manual testing, Global journal of enterprise of information system, January 2009-june 2009.
- [2] Alex Cervantes, “Exploring the Use of a Test Automation Framework”, IEEEAC paper #1477, version 2, up dated January 9, 2009.
- [3] Khaled Mustafa, Rafa E. Al-Qutaish, Mohammad I. Muhairat, “Cassification of Software testing Tools Based on the Software Testing Methods”, 2009 second International Conference on Computer and Electrical Engineering, 978-0-7695-3925-6, 2009.
- [4] Harpreet Kaur, Dr.Gagan Gupta, “Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete”.
- [5] R.S.Pressman, “Software Engineering A Practitioner’s Approach”, Mcgraw-Hill International Edition, ISBN 007-124083-7.
- [6] Harpreet Kaur and Sherry Singla “Selenium Keyword Driven Automation Testing framework” in International Journal of Advanced Research in Computer Science and Software Engineering.

- [7] Rashmi, Neha Bajpai “a Keyword Driven Framework for Testing Web Application” in International Journal of Advanced Computer Science and Applications, Vol. 3, No. 3, 2012.
- [8] Razak R.A and Fahrurazi presentea a paper “Agile Testing with Selenium” IEEE explorer digital library December 2011.
- [9] Nidhika Uppal and Vinay Chopra “Design and Implementation in Selenium IDE with WebDriver” in International Journal of Computer Applications May 2012.
- [10] Nidhika Uppal and Vinay Chopra “Enhancement and Elimination of Roadblocks in Automation Testing Tool Selenium RC” in September 2012.
- [11] Laukkanen, Pekka, “Data-Driven and Keyword-Driven Test Automation Frameworks”, Master’s Thesis, Software Business and Engineering Institute, Department of Computer Science and Engineering, Helsinki University of Technology, 2006.
- [12] Pettichord, “Seven steps to test automation success” in Proceedings of the Software Testing, Analysis & Review Conference (STAR), 1999.
- [13] G. A. Di Lucca, A. Fasolino, F. Faralli, and U. D. Carlini. “Testing web applications” in International Conference on Software Maintenance, 2002.
- [14] B. Posey and Mosley, “Just Enough Software Test Automation”, Prentice Hall PTR, 2002.