

## **Data Driven Software Testing with Selenium Apache POI Tool**

Ecem İREN<sup>1\*</sup>, Gizem İREN<sup>2</sup> and Aylin KANTARCI<sup>3</sup>

<sup>1</sup>*Program of Computer Programming, İzmir Kavram Vocational School, İzmir, Turkey*

<sup>2</sup>*Department of Computer Engineering, Ege University, İzmir, Turkey*

<sup>3</sup>*Department of Computer Engineering, Ege University, İzmir, Turkey*

\*Corresponding author: *ecem.iren@kavram.edu.tr*

**Abstract** – The concept of quality in software means that the software meets needs and expectations of the customers and also is free from defects or deficiencies. Quality software saves time and money since it is unlikely to occur errors in testing and maintenance stages. Assuring the quality of the software is related with software testing processes. Testing is the process of controlling the system carefully from different aspects, either manually or automatically, to see if it coincides with customer requirements. There are many reasons to test the software such making sure the product quality before it is presented to the customer, reducing rework and development costs and detecting errors in the early stages of the development process and preventing them from spreading further stages. The aim of the study is that investigating and giving a brief information about both Apache POI library of Selenium test tool and data driven testing. Therefore, in the context of the study, a data driven test application is implemented with Apache POI library of Selenium in order to perform our test scenario. Then the test methods are executed and test results are examined. Finally, it is understood that data driven testing can be carried out easily with built in methods of Apache POI library and can be installed for free by users for testing purposes.

**Keywords** – *Software Testing, Data Driven Testing, Selenium, Apache POI, Software Quality*

### I. INTRODUCTION

Software quality has become increasingly an important topic with the increase in large-scale applications. Quality can be determined with software testing processes. Software testing is the activity of detecting and correcting errors before the product is released to the market or reaching the customer. In addition, it provides knowing whether the product has reached sufficient maturity. Software test results are constantly presented to the project team with reports, so that the reporting and testing process can be followed. All stakeholders have up-to-date information about the progress of the project and the actions to be taken. Software tests are important to ensure the following items such as demonstrating the software can meet the requirements, ensuring the software progresses as planned, improving the quality of the software, assuring the software is bug-free in an acceptable level, guarantee that the software is sustainable. Moreover, it increases the customer satisfaction and and creates a groundwork for future orders [1]. Different types of software test methods are applied according to the test scenario. Functional testing is a kind of test level where software is controlled based on its requirements. Functional test checks the results of a given set of data by comparing them with the expected result set. Therefore, it is operated as result oriented and simulates real system usage. Regression tests are performed after the necessary corrections are made in the application. The problems identified in the previous tests are resolved and that new errors will not occur. On the other hand, performance test is often used synonymously with "load test". It can also be defined as the functionality test (workability test) of a system under severe conditions like unexpectedly heavy loads, huge increase in requests, very intensive numerical operations, complex queries, etc. User acceptance tests are done by the users to determine to what extent the application

meets the requirements before accepting the application. The white box testing technique is code testing and the source code of the project is tested. For example; flow controls, conditions, etc. in software code. elements are analyzed in this stage. Hence, such tests depend on knowledge of the internal logic of the application code. Black box testing is a process of checking the the application through the interface. In this test technique, knowledge of coding is not required [2]. Furthermore, test automation is used to alleviate the burden of test processes since testing is the most time-consuming stage in the life cycle of software development. Automated test procedures does not only contribute to cost reduction, it also aims to increase the quality of the project. In this context, various test automation tools are developed to automate the test scenarios [3]. In this study, we investigate the usage of Apache POI library of Selenium tool to handle data driven testing. At the end of the study, it is seen that the library has useful built in methods to access and get data for testing purposes. While second part of the paper explains the Selenium ve its structure, third section gives information about Apache POI library. Fourth section shows the data driven test of Excel file with Apache POI library and Selenium. Finally, study is summarized and concluded in fifth section.

### II. GENERAL SCOPE AND STRUCTURE OF SELENIUM

Selenium was developed in 2004 by Jason Huggins working at Thought Works company The story behind this initiative is as follows: At that time, he was working on a web application requiring regular testing. During these studies, he realized that the manual test processes has increased causing inefficiency. First, he created a JavaScript program that could automatically control browser actions and named it JavaScript Test Runner.

Later, he completed this program as open-source and named it Selenium Core [4]. Selenium is very popular and widely used today because it can run on many different browsers and operating systems, can be controlled by different programming languages and contains many different test functions. One of the most important features of automation testing is to prevent errors that may occur due to human interaction (manual testing) during the testing process [5]. Selenium is an open-source and web-based testing tool used to automate test cases and improve performance [6]. Selenium has become an important tool due to the complexity of testing processes with the expansion of web applications in today's world. It includes a very broad set of testing functions for the process of testing almost any web application. It is possible to perform high level operations with a wide range of test functions on different browser platforms. Commands written in Selenium are called "Selenese". Action oriented commands play a role in testing actions such as "Click this link" or "Choose this option". If such actions fail or an error occurs, the test execution process is stopped. There are three versions of Selenium that can be used in full automation for web applications, each with a different role [5] [7].

### Selenium IDE

It can be defined as an integrated development environment for tests. Selenium IDE, originally designed by Shinya Kasatani, has a recording feature and is used as a Firefox extension. Editing of test cases and debugging take place in this environment.

### Selenium Core:

Selenium Core is a testing tool for web applications and can be described as a simpler form of Selenium. However, it can not perform actions such as switching between HTTP and HTTPS protocols and file processing.

### Selenium RC:

It's a good solution for tests that require more than just plain browser actions. It is suitable for relatively complex tests such as reading and writing files, querying the database, and sending test results by e-mail. It is a requirement for Selenium RC that the server should be started before the test scripts are run.

### Selenium Grid

Selenium Grid is a tool to execute Selenium tests in parallel on different machines. Selenium WebDriver is the core library to drive web browsers on a single machine.

### Selenium WebDriver

Selenium WebDriver, in other words Selenium 2.0, WebDriver directly communicates with the browser, so it can be said that it is much faster than Selenium RC [7]. Selenium WebDriver, which includes many language support for writing test files, has the ability to function through different browsers and

operating systems, but it does not have the ability to test desktop-based applications [8]. In addition, the ability to generate test results is not available. The general structure of Selenium is summarized as in Figure 1.

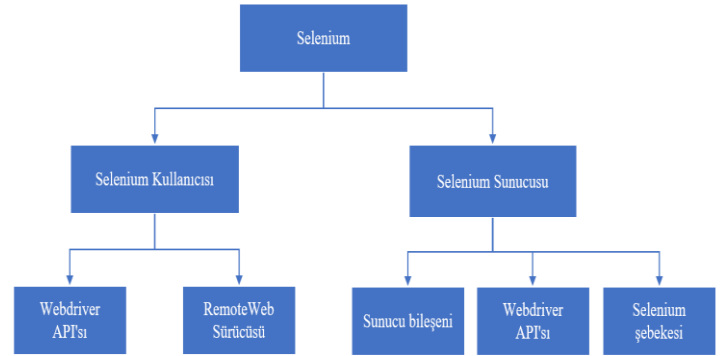


Fig. 1 General Structure of Selenium

As seen in the figure, Selenium user and server appear as two basic components. Both components include the WebDriver API. The WebDriver API is used to interact with web page and other application elements and to generate test scripts. Additionally, it includes a remote web driver class that communicates with a remote Selenium server. Selenium server consists of a server component used to accept requests from Selenium user's RemoteWeb driver, WebDriver API used to run web browser tests, and Selenium network used to run parallel tests via dissimilar machines and browsers (Devi et al., 2017). Workflow of Selenium can be explained as follows:

- The automation script is run.
- For each Selenium command, an HTTP request is created and sent to the browser driver.
- The browser driver uses an HTTP server to receive HTTP requests.
- The HTTP server determines the necessary steps to implement the Selenium command.
- The steps are executed on the browser.
- Application state is sent back to the HTTP server.
- The HTTP server sends the state back to the automation script [9].

Selenium WebDriver supports many different types of client libraries such as PHP, C# (Nunit), Java (TestNG, JUnit), Perl, Ruby (RSpec) and Python (Robot, PyUnit, Unittest). In addition, it also works in web browsers such as Chrome, FireFox, Safari and PhantomJS.

### III. APACHE POI LIBRARY OF SELENIUM

Apache POI is a popular API that allows programmers to create, modify and view MS Office files using Java programs. It is an open source library developed and distributed by the Apache Software Corporation. Apache POI contains different classes and methods to work with MS Office documents. The service is divided into different components such as:

- HSSF (Horrible Spreadsheet Format): It is used to read and write xls formats of MS Excel files.
- XSSF (XML Spreadsheet Format): It is used to read and write xlsx formats of MS Excel files.
- XWPF (XML Word Processor Format): It is used to read and write docx formats of MS Word files.
- HSLF (Horrible Slide Layout Format): It is preferred to read, update or create Powerpoint representations.
- HDGF (Horrible DiaGram Format): It contains methods and classes required for MS-Visio files.

There are also tools released by different companies such as Apache POI. These tools can be listed as Aspose cells for Java (Aspose), JXL (Common Libraries) and JExcel (Team Dev) [10].

#### IV. APACHE POI LIBRARY OF SELENIUM

A new project has been created in the Netbeans tool to perform data-based Excel tests. The relevant jar files should be added to the build directory of the project as seen in Figure 2. In addition, the TestNG tool is used for performing the tests and reporting of test results.

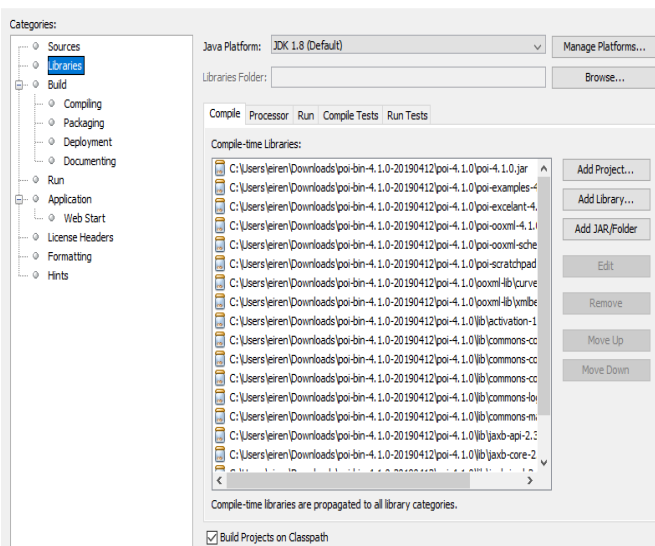


Fig. 2 Creating the Excel Transaction Class

An Excel list is created for testing. In this list, information such as user e-mail address and password are stored. In the project, e-mail addresses and passwords of the users are read respectively from the Excel file and directed to the Gmail login page. In order to extract the relevant data from the Excel file, a separate Excel operation class has been created as follows. In this class, some variables are created from the classes defined in the Apache POI library to access the target Excel file and the worksheet including the test data. The directory of the Excel file and the name of Excel worksheet are given as parameters to the constructor method of the class. Therefore, user should specify the appropriate parameter values while creating an object of this class. In this method, the FileInputStream object is defined with the directory name of Excel file parameter, and then the Excel workbook is accessed

with this object. Then, with the help of the Excel workbook object, the worksheet is accessed.

```
public class ApachePOIExcelOperations {
    public String path;
    public FileInputStream fis = null;
    public FileOutputStream fileOut = null;
    private XSSFWorkbook workbook = null;
    private XSSFSheet sheet = null;

    private Object[][] excelData;

    public ApachePOIExcelOperations(String path, String sheetName) {
        this.path = path;
        try {
            fis = new FileInputStream(path);
            workbook = new XSSFWorkbook(fis);
            sheet = workbook.getSheet(sheetName);
        }

        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Fig. 3 Extracting Excel Data

A separate test class has been created for data-based testing and it can be seen in Figure 4. Here, the ChromeDriver is defined and the web page to which the application will be automatically redirected is specified. If there is a method to be run before test automation in TestNG, this is done by adding the @BeforeTest notation to the relevant method.

```
public class DataDrivenTest {
    WebDriver driver;

    @BeforeMethod
    public void init()
    {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\eiren\\Downloads\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.get("http://gmail.com");
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    }
}
```

Fig. 4 Data-Based Test Class

A data provider is needed for data-based testing, and this dataset is created with the array filled with Excel data illustrated in Figure 5 by giving a special name in the parameter.

```
@DataProvider(name="ExcelDataForTest")
public Object[][] getExcelData() throws IOException
{
    ApachePOIExcelOperations operation = new ApachePOIExcelOperations("Credentials.xlsx", "Credentials");
    return operation.getExcelData();
}
```

Fig. 5 Creating Data Provider

While testing name of the dataset is described with data provider before the test method shown in Figure 6. Then, it is necessary to determine the appropriate fields on the web page in order to control the login with both the user name and password in the data set. At this point, built in methods of Selenium can be also benefited. After the fields are decided,

the data fields are sent and the login process of the web page is done automatically. The success of the test is determined by the URL information of the web page. The current URL is compared with the expected URL and the test result is considered as successful if both are equal. Otherwise, the test will fail. When the test class is run, it is seen that the test passed successfully as in the Figure 6 and it is understood that the Excel data is valid and correct.

```
@Test(dataProvider="ExcelDataForTest")
public void authenticateLogin(String userName, String userPassword) throws InterruptedException
{
    String expectedURL = "https://mail.google.com/mail/u/0/#inbox";
    WebElement email_phone = driver.findElement(By.xpath("//*[@id='identifierId']"));
    email_phone.sendKeys(userName);
    driver.findElement(By.id("identifierNext")).click();
    WebElement password = driver.findElement(By.xpath("//*[@name='password']"));
    Thread.sleep(1000);
    password.sendKeys(userPassword);
    driver.findElement(By.id("passwordNext")).click();
    Thread.sleep(5000);
    String actualURL = driver.getCurrentUrl();
    assertEquals(actualURL, expectedURL);
}
```

Fig. 6 Testing the data with Selenium Commands

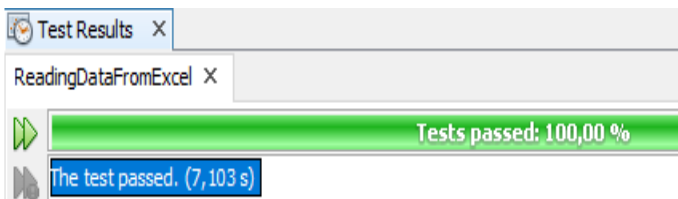


Fig. 7 Successful Test Result

## V. CONCLUSION

Software tests are of great importance in software projects in terms of cost, time and customer satisfaction. Time should be devoted to software testing to ensure the maximum quality of the software before it is presented to the customer. For this reason, tests play an important role in order to see that the products function correctly and meet customer expectations. On the other hand, application defects return to companies as a cost since correction of defects leads to costs. Tests need to be accomplished properly and comprehensively to minimize these costs. Thus, the errors determined by the testers are corrected in the early stages and are not spread to the later stages. This provides benefits to the institution in terms of time and cost. As a result, test experts should be included in the software projects developed and the software should be tested from different aspects before it is released to the user. Within the scope of the study, a data driven test application is written with Apache POI library of Selenium test tool in order to perform our test scenario. Then the test methods are executed and the results are examined. Selenium is free of charge and can be easily installed and used by users for testing purposes. Our aim is to give information about both Selenium tool and data driven testing for the people who are interested in testing operations.

## REFERENCES

- [1] B. Uzun and K. Koruyan, "An overview and approaches to status reporting in the software test process", *Journal of Management Information Systems*, vol. 5, no.1, pp. 52–63, Dec. 2019.
- [2] ITU Bilgi İşlem Daire Başkanlığı. (2013) Yazılım Testi ve Test Süreçleri. [Online]. Available: <https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/yaz%C4%B1%C4%B1m-testi-ve-test-s%C3%BCre%C3%A7leri>
- [3] H. A. Yener, F. Baştürk and B. Meçik, "General Mobile Productivity Increase with Software Development and Test Automation: General Mobile", *Karamanoglu Mehmetbey University Journal of Engineering and Natural Sciences*, vol. 1, no. 1, pp. 172-196, 2019.
- [4] P. Yadav and A. Kumar, "An Automation Testing Using Selenium Tool", *International Journal of Emerging Trends and Technology in Computer Science (IJETCS)*, vol. 4, no. 5(2), pp. 68-71, 2015.
- [5] J. S. Raju and V. Vaidhehi, "Design and Implementation of Hybrid Test Automation Framework for Web Based Application", *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 4, no. 3, pp. 108-112, 2017.
- [6] Chandrababha, Kumar, A. and Saxena, S., Data Driven Testing Framework using Selenium WebDriver, *International Journal of Computer Applications*, 118:18-23, 2015,
- [7] Gojare, S., Joshi, R. and Gaigaware, D., 2015, Analysis and Design of Selenium Webdriver Automation Testing Framework, *Procedia Computer Science – 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)*, 50: 341-346.
- [8] Devi, J. ve diğ., 2017, A Study on Functioning of Selenium Automation Testing Structure, *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(5): 855-862.
- [9] Quan, D. M., 2018, Implementing test automation with Selenium WebDriver, Lathi University of Applied Sciences, Degree Programme in Business Information Technology, Bachelor's Thesis.
- [10] ITU Bilgi İşlem Daire Başkanlığı. (2013) Yazılım Testi ve Test Süreçleri. [Online]. Available: <https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/yaz%C4%B1%C4%B1m-testi-ve-test-s%C3%BCre%C3%A7leri>