Selenium Based Test Automation Framework For Web Application

1st Soham Prasad Kulkarni
Dept of Computer Science

University of Massachusetts Lowell
Lowell, Ma, USA

SohamPrasad_Kulkarni@student.uml.edu

Abstract— **This paper discusses issues in the automation testing of GUI-based Web applications. A industry based method of automation testing framework is proposed, which is based on the concept of page object model (POM), dynamic locators and certainly new technologies such as BrowserStack, Selenium and CircleCi. Its design and implementation are both provided. The framework's use demonstrates that it makes testing more convenient and efficient, with fewer resources and time costs but greater testing coverage. Both the ability to maintain and the stability are enhanced.**

Keywords—Selenium Webdriver, Web-application, Testing, BrowserStack, Locators, CirrcleCI

I. Introduction

With the advancement of internet technology, the Web application becomes more complicated and the scale of it grows larger and larger. Its quality and dependability receive much more attention. Web application testing, particularly regression testing, is far more difficult than traditional methods. And all the while, as the software development cycle shortens, web application testing, particularly functional regression testing, is becoming more difficult as the applications are becoming too complicated and became multi-page. Traditional methods of testing can no longer meet the needs of software development testing, so automation testing becomes the only option. Automation scripts generated by any online based technology are too hard to maintain as a commonly used method. As a result, developing customized great web testing tools and frameworks makes the tasks efficient for your product.

II. Background

Automation testing tools or programming language-based automation programs are used to control the testing classes and objects and complete the test job. The goal of automation testing is to automate testing without human intervention in order to reduce costs while also making testing more reliable and efficient. Many test tools have been developed and deployed.[6]

With the aid of cutting-edge technologies, such as Java, Python, JavaScript, and many more languages supported by Selenium, the notion of page of model, BrowserStack, CircleCi, and version controls such as GIT and Github, a customized testing framework can be created.

Different testing tools focus on different aspects of the testing process. The application can select the best tool based on the needs of the business. However, some aspects still require improvement. When a webpage changes, many automation testing tools fail to execute correctly. The tested script's second time input is required. The structure of the changed pages must be captured once more. Even minor page changes will result in the unexpected error. As a result, testing will take much longer. The testing scripts' maintenance degrades dramatically. As a result, it is necessary to conduct research, design, and develop a more flexible and adaptable automation framework.

III. Technical Introductions

Selenium: A portable testing framework with a focus on web-based applications is Selenium. Jason Huggins created it in its initial form back in 2004. To make the use of Selenium easier, many other tools can be integrated with it. a Selenium Grid driver. One element of the Selenium toolkit is the Selenium Web Driver. For testing dynamic web page elements like links, downloads, and pop-ups, the Web driver offers better support. among all the Selenium toolkit's other elements [1].

BrowserStack: is a cloud web and mobile testing platform that allows developers to test their websites and mobile applications across on-demand browsers, operating systems, and real mobile devices [2].

Git: is a distributed version control system that tracks changes in any set of files and is commonly used for coordinating work among programmers who collaborate on source code development during software development. Its objectives include data integrity, speed, and support for distributed, non-linear workflows [3].

Github: As a cloud similar to Microsoft OneDrive, GitHub also enables programmers to share, search for, download, and like programs. Similar to what we do on Facebook, but with different content. [4]

CircleCI: is a continuous integration and delivery platform that enables development teams to release code quickly while automating the build, test, and deploy processes. CircleCI can be configured to run extremely complex pipelines with caching, docker layer caching, resource classes, and many other features. After a GitHub or Bitbucket repository is authorized and added as a project to circleci.com, every code triggers CircleCI to run jobs. After the tests are completed, CircleCI sends an email indicating success or failure. [5]

IV. Design and Implementation

A The frame will essentially have a directory structure where we can put everything under the e2e-test directory and then create subdirectories for configs, pages, and actions that we need to test the UI. Starting with the config directory, it will contain all of your configuration files for running the framework, initializing variables, and sending various types of environments to test the page.

The actions directory will contain customized actions that are used on a regular basis. In other words, it will contain customized functions that are required to test the various elements of the UI on a regular basis.

For example, a click is a functionality that the tester uses frequently, so using the selenium You can create your own click functionality and use it however you want.

The Pages directory will contain several files for each page of your website, and each file will contain the locators and functions needed to create a testcase scenario, and the object of this file's page is exported and used in running the testcase. If a user wants to use the page as a load before another page, they can do so by using the page object rather than writing the entire script for the scenario. This is known as the page object model technique.

Apart from that, in the page's files, each element that is involved in testing is used with the help of locators.

To get the locators, you can use a variety of methods, one of which is XPATH. You can inspect on the element and then go to copy options, which will show you to copy XPATH.

You can also use CSS, Class, and ID for locators. The locators we are using are dynamic locators which can be constructed on the basis of which element you are using.

A tests directory is made, and the included files will have usage cases built into them. Each end-to-end file will be able to use the configuration from the config files to run the particular scenarios. Speaking of testing locally using Chrome and Firefox, test cases can also be executed using a third-party program called BrowserStack. The configurations offered by BrowserStack must be used to integrate it, and their website also provides guidance. As humans, we are unable to execute all test cases continuously or verify the web application at regular intervals. This is where continuous integration and deployment come into play. We may use CircleCi to perform the regression or sanity tests at predetermined intervals which can be done by creating a ".yaml" script that will execute after your code has been submitted to Github. According to your configuration, the Yaml file will generate a job and execute specific test suite jobs on the web application. In this certain manner the Testing framework can be designed and implemented

.

Conclusion

The use of page object models, dynamic locators, and efficient integration of BrowserStack and circleci will produce better results in performing tests at the industry level because it will save time and increase the productivity of web application testing.

References

[1] P. Ramya, V. Sindhura and P. V. Sagar, "Testing using selenium web driver," 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2017, pp. 1-7, doi: 10.1109/ICECCT.2017.8117878.

[2] https://medium.com/chaya-thilakumara/browserstack-b4ea85599a23.

[3] .https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

[4] https://medium.com/@george.seif94/a-full-tutorial-on-how-to-use-github-88466bac7d42

[5] L. Gota, D. Gota and L. Miclea, "Continuous Integration in Automation Testing," 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2020, pp. 1-6, doi: 10.1109/AQTR49680.2020.9129990.

[6] K. Sawant, R. Tiwari, S. Vyas, P. Sharma, A. Anand and S. Soni, "Implementation of Selenium Automation & Report Generation Using Selenium Web Driver & ATF," 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2021, pp. 1-6, doi: 10.1109/ICAECT49130.2021.9392455.

[7] Z. Wandan, J. Ningkang and Z. Xubo, "Design and Implementation of a Web Application Automation Testing Framework," 2009 Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 316-318, doi: 10.1109/HIS.2009.175.