

Implementation of Test Automation with Selenium Webdriver

Aneesha Gudavalli¹ and G.JayaLakshmi²

¹M.Tech Scholar, Department of Information Technology, V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

Email: gudavallianeesha@gmail.com

²Assistant Professor, Department of Information Technology, V R Siddhartha Engineering CollegeVijayawada, Andhra Pradesh, India

Email: jaya1123@vrsiddahrtha.ac.in

Abstract—As a result, a wide variety of methods are used to collect data that counts. All the data collection methods, web scraping is the most reliable and efficient one. Popular test automation framework Selenium can be used to scrape web pages in a variety of ways. In this paper, we demonstrate web page scraping using Python, a popular backend programming language. Test automation frameworks such as Selenium also support it. Information can also be scraped from static web pages. As a result, this paper examines the major advantages and challenges of web scraping in building web applications. The framework is fully functional and can be used by both novice and advanced users to automate the testing of web sites.

Index Terms— HTML, Python ,JavaScript,webdriver.

I. INTRODUCTION

Anywhere on the internet, information is one of the most valuable resources. To function properly, several web services and applications necessitate large amounts of data. Web search engines (Google, Duckduckgo), product value and feature similarity sites (versus.com, pricealarm.net) or inner tools used by several companies for market and entrant analysis are examples of such applications that are available. Two applications will be created to better understand this process.

One of the applications uses Web Scraping as a back-end service and compares three popular online electronics stores in real-time. Web Scraping provides the data for the subsequent application, which is a simplified edition of Google's web search engine based on daily Web Scraping. The principle of web scraping is to mine a large extent of data and hoard it to a local setting [1]. In the untimely days, the only way to mine data from websites was by copy-pasting what one axiom on the website. Web scraping is a popular technique as it allows innovative startups to quickly attain large amounts of facts. Most emblematic examples of web scraping are value comparison as well as assessment websites. Big firms also use this technique to grade and file web pages.

In order to scrape a website, you must crawl and parse it[1]. In accordance with the developers implementation of crawlers, crawlers visit the URL, download the page content, and follow links within the content. Data is extracted by parsing downloaded content. You must know the basic components of web applications as well as the website rendering process in order to implement parsers and crawlers.

A. Web application components

A website's components and structure can vary. HTML, CSS, JavaScript and multimedia resources are the four main components of a web page. In addition, Document Object Model, CSS selector and Xpath give developers the ability to control the content of the web page.

B. Web application interpretation process

However, it is important to understand how the browsers provide pages and interact with whichever backend servers. Difficulty of scraping varies based on when the beleaguered data are rendered or presented via API call. In some cases, a real web browser is used to scrape websites [2]. It is necessary to perform a discovery process before extracting data from websites. Crawlers visit sites and perform other tasks. Web Scrapers need parsers in sort to mine meaningful data as of scraped data.

In most cases, they are executed by programmers to arrange and mine specific details from the data, such as a CV parser that can mine a person's name and his information from an electronic message.



Fig. 1 A sample of web scraping

The critical benefit of utilizing the Selenium WebDriver for Selenium test robotization is that the correspondence with the basic program is performed through a program explicit driver. Every program will has its relating WebDriver which is utilized for speaking with the program on which tests are performed. Subsequently, any update in the program form/highlights would just include support for those new highlights in the WebDriver introduced on the test machine.

Principal Building Blocks:

1. Selenium User Libraries
2. JSON Wire-Protocol
3. Browser-Drivers
4. Browsers

Selenium Grid depends on the customer worker design, where there is just a single worker (additionally called Hub) and different customers that speak with the worker (likewise called Nodes). Worker is additionally alluded to as Hub, and the Client is alluded to as Node. By utilizing the Selenium matrix arrangement, you can accomplish parallelism in your testing movement. At the point when the client starts a test demand, the solicitation is first passed to the Hub, which then, at that point surrenders the test execution solicitation to the Node machine which coordinates with the necessary abilities referenced in the test

II. BACKGROUND

Jason Huggins and Paul Hammant, while functioning at ThoughtWorks, released Selenium as open-source in 2004. They chosen the name "Selenium" as a descant to Hewlett-existing Packard's Mercury testing outline. Huggins believes that Selenium should be the testing outline that overcomes Mercury's limits.

Selenium is a type mineral that protects the organization from Mercury toxicity, and that's the idea behind its name. Selenium core was the first version of selenium, which was designed to simulate the actions of a human user on web applications, e.g. by opening URLs, clicking links, entering text into forms, and so onward in an programmed manner. "Selenium Core" is a JavaScript documentation that translates commands called Selenese operations. "Remote Control" was created by Huggins and Hammant to provide a scripting level for Selenium Core (RC). Figure 1a shows the client-server architecture of Selenium RC. Java, Perl, Python, Ruby, C# or PHP-based Selenium scripts perform as clients, transfer Selenese commands to an HTTP substitute called Selenium RC server. There were a number of actions that could not be performed because JavaScript was the support technology to carry programmed web connections. As a result, the final performance of the test is negatively affected by Selenium's RC Server.

This study is in the area of test automation. We preserve locate a significant amount of existing contributions in this field in the prose. Rafi et al., for example, provide a thorough review of the benefits and drawbacks of programmed software testing [3].

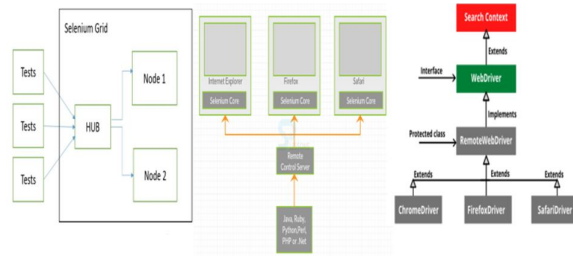


Figure. 2 Selenium Grid , Remote Control (RC) and WebDriver architectures

TABLE I. WEB DRIVER TESTING FRAMEWORK USING SELENIUM

Author	Year	Paper Title	Overview
Kasurinen, J.; Taipale, O.; Smolander, K.	2010	Software test automation in practice: Empirical observations	Revision to testing with realistic difficulty in usability are conversed
Tsung-Hsiang Chang et al	2010	GUI Testing Using Computer Vision	Novel method to GUI testing to interrelate and response to be observed
Mesbah, Ali & Prasad, Mukul.	2011	Automated cross-browser compatibility testing	Presented an automated solution cross-browser compatibility testing of recent web applications as a 'functional reliability check of web application behaviour athwart dissimilar web browsers
G. Mohan Doss Gandhi and Anitha S. Pillai	2014	Challenges in GUI Test Automation	Analysed the a variety of methods of recognizing UI controls in GUI Test Automation.
Miroslav Bures, M. Filipisky	2016	SmartDriver: Extension of Selenium WebDriver to Create More Efficient Automated Tests	Created tests which can be effortlessly read by testers or business analysts devoid of preceding extensive programming understanding.
M., Niranjanamurthy & Navale, Sushanth & Jagannatha, S & Chakraborty, Sudesha.	2018	Functional Software Testing for Web Applications in the Context of Industry.	Proposed functional testing test case is a set of circumstances or variable/s under which a functional tester will decide whether a system under test satisfy requirements or facility correctly.
Jussi Kasurinen,Ossi Taipale,and Kari Smolander	2020	Software Test Automation in Practice:Empirical Observations	Analyzed existing situation and upgrading needs in software investigation automation.

A large volume of data can be extracted from a website using web scraping. This allows us to identify brand attribute associations, which are crucial for any business. Test automation framework Selenium allows us to write down tests in many programming languages including Java, C#, and Groovy and Perl. Testing scenarios against different browsers is possible with WebDriver, a web automation framework. In order to automate a browser, Selenium-WebDriver makes direct calls to the browser using the local automation support of each browser. WebDriver can be used in any programming language, which makes it easier to design tests. Basically, Selenium web driver locates a web element on screen based on its position.

To develop a Selenium hybrid framework, which is a combination of page object model, pytest, and data-driven techniques in python language for testing web applications and automating web pages against different browsers? It seeks to integrate the benefits of multiple structures to make a better, more efficient framework. Web pages are first tested and then automated using the data that is acquired from a bot or web pages, which is then stored in a database.

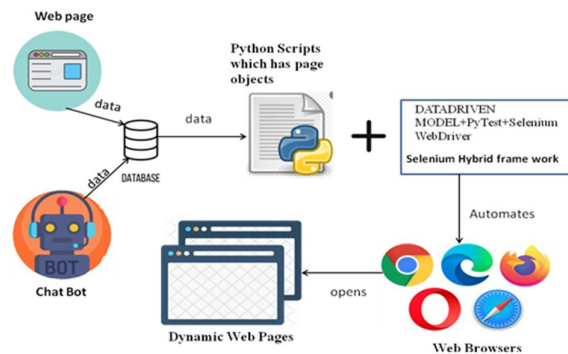


Figure. 3 Selenium Hybrid Framework

SeleniumBase robotically handles familiar WebDriver actions such as rotating up web browsers and cutback screenshots throughout test failures.

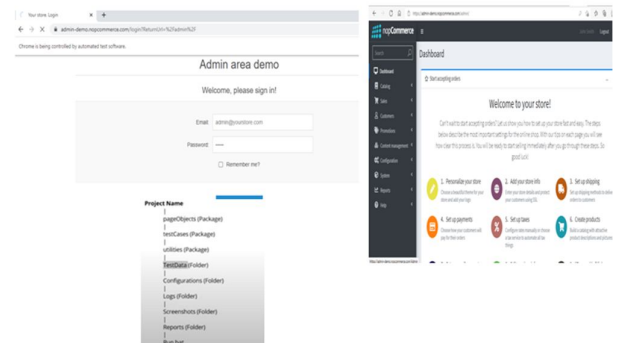


Fig. 3 (a) Results and observations

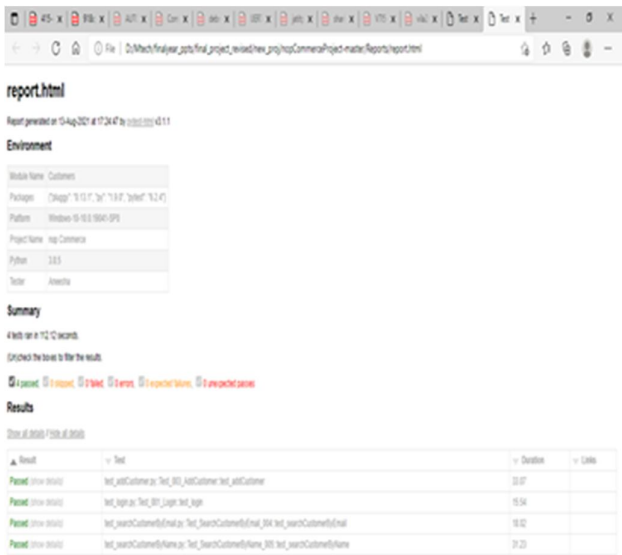


Figure. 3 (b) Results and observations

```
13/08/2021 07:24:53 AM: INFO: ***** Test_001_login *****
13/08/2021 07:24:53 AM: INFO: ****Started Home page title test ****
13/08/2021 07:24:53 AM: INFO: ****Opening URL****
13/08/2021 07:24:58 AM: ERROR: *** Home page title test failed***
13/08/2021 07:25:04 AM: INFO: ****Started Login Test****
13/08/2021 07:25:16 AM: INFO: ****Login test passed ****
13/08/2021 07:32:10 AM: INFO: ***** Test_001_login *****
13/08/2021 07:32:10 AM: INFO: ****Started Home page title test ****
13/08/2021 07:32:10 AM: INFO: ****Opening URL****
13/08/2021 07:32:14 AM: INFO: **** Home page title test passed ****
13/08/2021 07:32:20 AM: INFO: ****Started Login Test****
13/08/2021 07:32:29 AM: INFO: ****Login test passed ****
12/08/2021 07:33:18 AM: INFO: ***** Test_001_login *****
12/08/2021 07:33:18 AM: INFO: ****Started Home page title test ****
12/08/2021 07:33:18 AM: INFO: ****Opening URL****
12/08/2021 07:33:22 AM: INFO: **** Home page title test passed ****
12/08/2021 07:33:33 AM: INFO: ****Started Login Test****
12/08/2021 07:33:44 AM: INFO: ****Login test passed ****
12/08/2021 07:34:59 AM: INFO: ****Started Login Test****
1/08/2021 07:35:02 PM: INFO: ***** Test_001_login *****
1/08/2021 07:35:02 PM: INFO: ****Started Home page title test ****
1/08/2021 07:35:02 PM: INFO: ****Opening URL****
1/08/2021 07:35:06 PM: INFO: **** Home page title test passed ****
1/08/2021 07:35:14 PM: INFO: ****Login test passed ****
1/08/2021 07:35:31 PM: INFO: ****Started Login Test****
1/08/2021 07:35:31 PM: INFO: ***** Test_001_login *****
1/08/2021 07:35:31 PM: INFO: ****Started Home page title test ****
1/08/2021 07:35:31 PM: INFO: ****Opening URL****
1/08/2021 07:35:33 PM: INFO: **** Home page title test passed ****
1/08/2021 07:35:42 PM: INFO: ****Login test passed ****
1/08/2021 07:39:21 PM: INFO: ****Started Login Test****
1/08/2021 07:39:21 PM: INFO: ****Started Test_002_DOT_login ****
1/08/2021 07:39:21 PM: INFO: ****Started Test_002_DOT_login ****
1/08/2021 07:39:21 PM: INFO: ****Started Test_002_DOT_login ****
1/08/2021 07:39:21 PM: INFO: ****Started Test_002_DOT_login ****
1/08/2021 07:39:21 PM: INFO: **** Passed ****
```

Figure. 3 (c) Results and observations

Features of framework of test automation using selenium as well as Python:

- Page object model Framework.
- Allure report.
- Reading locators as of JSON file.
- Reading test data as of JSON file.
- Incorporated with TestRail to renew test cases category after every run.
- Framework can incorporate by CodeShip.

Create new test case:

In order to construct a new test case by means of the Framework

1. In locators part, create a latest locator for the element to use:

```
[{
    "pageName": "HomePage",
    "name": "link_login",
    "locateUsing": "xpath",
    "locator": "//a[contains(text(),'Log In')]"
}]
```

2. In test data part, the test data is added which is needed for test case:

```
{
    "environment": "https://learn.letscodeit.com/",
    "browser": "firefox",
    "email": "test@email.com",
    "password": "abcabc"
}
```

3. In order to group the integration with TestRail, revise testrail.cfg with testrail domain as well as credentials

```
[API]
url = https://[your testrail domain].testrail.io
email = [testrail email]
password = [testrail password]

[TESTRUN]
project_id = [project id]
```

bs4 and requests are replaced by a single scraping library. When running Selenium in headless mode, Chrome browser runs on screen. If we have taken Wikipedia for scrapping, there are better places to get information, such as the Python Wikipedia library and Wikipedia's data dumps.

- test search functionality of <https://wikipedia.com/> web site.
- Search for “selenium” text word
- Validate that “selenium” text word is in look for input field
- Verify that there is a link with “Selenium Web Browser Automation Python” text word

Each time we click on one of our links, we collect data and append it to our dataframe. Because we need to access a large number of pages; this can take a little time. Using the dataframes apply method, we can clean up the data a bit. To make all of our description text lower case, we used it. The stopwords were also removed from the list. They are all words in a text that are necessary for grammar but don't necessarily convey meaning or convey a message. Some of these words include: at, a, to, etc. A word like with or is has been removed. Punctuation has also been removed. Our text clean function will be applied to all cells in the description column of the dataframe using this method. Last but not least, we can save our work to CSV format.

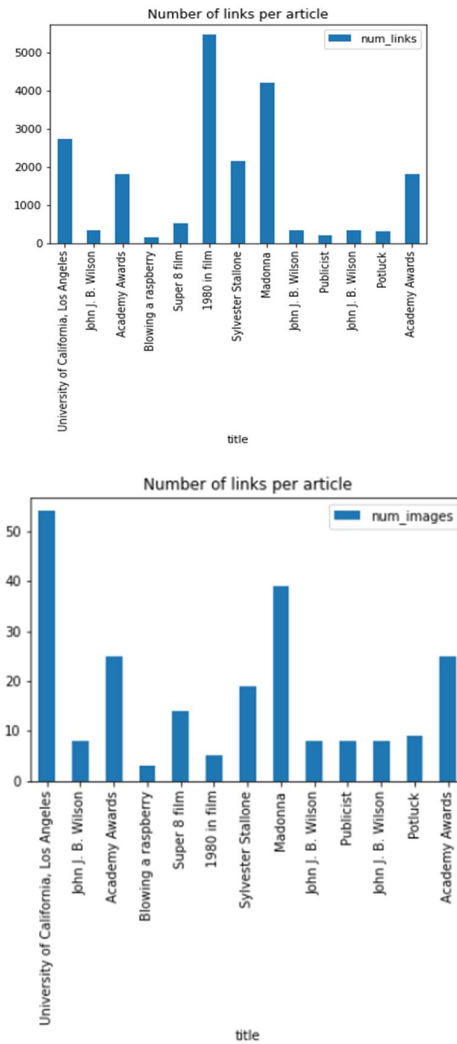


Figure. 4 Scraped data to dataframe

df.to_csv("wiki.csv")

An Alert is a pop-up window and is triggered by action performed by the user or by design due to system settings. Their intention is to give information to the user or obtain permission from the user or to obtain input from the user. When an Alert acquires triggered, and a pop-up emerges on the web page. The control vestiges with the parent web page merely. So the initial task for the Selenium Webdriver is to control the focal point from the parent page to the Alert pop-up. An effortless Alert has a note on it as well as an 'OK' button. While it pops up, a user clicks on the 'OK' button to admit it.

The access time for creating test case is given by Implicit Wait and Explicit Wait which inculcates the webdriver to pass the time for a definite amount of time before elements stack on the webpage.

III. CONCLUSIONS

The authors can conclude on the topic discussed and proposed. Future enhancement can also be briefed here. Software development, and in particular Agile software development with Scrum methodology, is the focus of this research. Users should not be required to have any programming or test automation experience. The test automation framework should introduce users to the process. These pages do not contain information about test automation or testing. To achieve the main goal, a concrete framework is presented in this paper, along with the

main features that were implemented. Python, popular Python libraries, Page Object pattern and Selenium WebDriver were used to develop the framework.

In this paper discuss about selenium framework. The main benefit of using automated tools is to avoid manual effort. Selenium is a web based automation framework which uses different platform and programming languages. These features are record and playback and run in parallel tests. It can reduce the time and provide free software and easier for programmers for programming as well as for scraping data. . In future, we can create an application by integrating Robotic Process Automation with Artificial Intelligence for better approach. few research work on these integrated fields are in progress.

REFERENCES

- [1] Mahto D, Singh L. A Dive into Web Scraper World. Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on March 2016 : 689-693
- [2] Upadhyay S, Pant V, Bhasin S, Pattanshetti M. Articulating the Construction of a Web Scraper for Massive Data Extraction. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) in February 2017
- [3] Rafi, D.M.; Moses, K.R.K.; Petersen, K.; Mäntylä, M.V. Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In Proceedings of the 2012 7th International Workshop on Automation of Software Test (AST), Zurich, Switzerland, 2–3 June 2012; pp. 36–42.
- [4] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [5] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
- [6] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.
- [7] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, “High resolution fiber distributed measurements with coherent OFDR,” in Proc. ECOC’00, 2000, paper 11.3.4, p. 109.
- [8] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.
- [9] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [10] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-hive/macros/latex/contrib/supported/IEEEtran/FLEXChip> Signal Processor (MC68175/D), Motorola, 1996.
- [11] “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.
- [12] A. Karnik, “Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP,” M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [13] J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [14] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.