Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, No. 5, July : 2023

WEB APPLICATION TEST AUTOMATION WITH ROBOT FRAMEWORK

Harshitha.C, Dept. Of Information & Science, National Institute of Engineering,
Dr K.Raghuveer, Professor, Dept. Of IS, National Institute of Engineering.

Abstract
The demand for robust and extensible system software has significantly increased. However, relying on manual testing techniques can impede the accuracy and quality of the software product, especially during regression testing where errors are more likely. To address this issue, this paper introduces an automated system that streamlines the testing process and ensures consistency. By leveraging the keyword-driven testing approach and reusing existing keywords, the system validates the functionality of the system software, resulting in time savings and improved code standards.

Keywords: Regression testing, keyword-driven testing approach.

I. Introduction

To ensure seamless integration of a component into a larger system, three key attributes—compatibility, accuracy, and resilience—must undergo thorough testing. Compatibility refers to the suitability of a component's provided interface with the required interface specifications of the application. Accuracy pertains to the component's ability to produce correct outputs when provided with accurate inputs, while resilience focuses on preventing any behavior that could potentially compromise the overall system, particularly when incorrect inputs are encountered. Integration testing becomes increasingly complex as the number of components grows, making test automation a crucial aspect of improving the software development process. Over time, system-level test automation has evolved through multiple generations, with each generation introducing higher levels of abstraction in test design. The current state-of-the-art approach in test automation, known as keyword-driven testing, abstracts test implementation through high-level actions, or keywords. In GUI testing, keywords typically represent basic user interactions like key presses, typing, or text reading. Tests are constructed as sequences of keywords, which are then automatically translated into specific low-level scripts. Although numerous test automation tools exist, both commercial and open-source, only a few are well-suited for black-box testing. Keyword abstraction offers two primary advantages: reduced maintenance efforts associated with tests and improved test development and comprehension. By consolidating the detailed implementation of actions within a single script, maintenance work is simplified to that particular script. Furthermore, the creation and understanding of tests do not require extensive programming expertise, as keywords resemble familiar, high-level commands commonly used in everyday activities. Compared to traditional test scripts, keyword scripts are also significantly shorter. The Robot Framework, an open-source automation framework, is a versatile tool that excels in test automation. It fulfills all requirements with ease and is implemented in Python, making it compatible with various platforms. Unlike many other open-source tools, Robot Framework stands out for its exceptional support for multi-platform environments and regular maintenance. Additionally, it boasts a highly modular architecture.

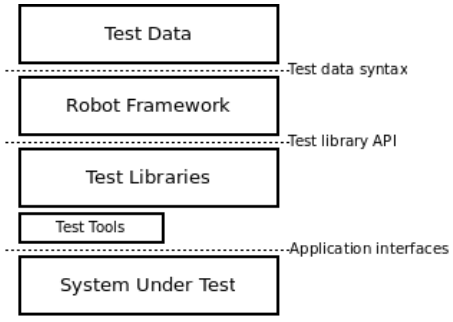

Fig: High level architecture

The test data is presented in a user-friendly tabular format, making it simple and easy to edit. Upon starting the framework, it proceeds to process the test data, execute test cases, and generate logs and reports. The central framework itself lacks knowledge about the specific target being tested or the interactions managed by the test libraries. These libraries have the option to interact with the application interface directly or utilize lower-level test tools as drivers. Test cases are stored in HTML files and leverage keywords implemented within the test libraries to steer the software under test. Meanwhile, test suites are constructed from files.

II. Literature

[1] "In the 2018 IEEE 11th International Conference on Software Testing Verification and Validation (ICST) paper titled 'Automating Web Application Testing from the Ground Up: Experiences and Lessons Learned in an Industrial Setting,' the authors discuss the significance of automating web application testing due to the rapid growth in software development and deployment. Testing web applications presents various challenges, such as ensuring functionality across different web browsers, including multiple browser versions, and assessing compatibility on various devices, such as mobile and desktop, to accommodate diverse user views. Numerous open-source automation tools are available online to facilitate these testing procedures, including Selenium IDE, Casper JS, Progress Test Studio by Telerik, and Browser Stack. Although these tools were found to present certain obstacles during the automation testing of the software examined in the paper, it does not imply that they lack overall effectiveness for automation testing. Following a thorough analysis of the encountered difficulties, the researchers developed a customized framework specifically tailored to the testing requirements of the system under evaluation. This bespoke framework proved highly effective for the application being tested; however, it necessitated higher development time and cost."

[2] In their paper titled "Extension based on robot framework and application on Linux Server" presented at the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS) in Beijing, Na and D. Huaichang highlight the benefits of utilizing the Robot Framework for automating functional test regression. By employing an automated test suite, the entire product can be thoroughly explored on a daily basis, whereas manual testing would require significantly more time to revisit all aspects. Automation facilitates the rapid detection of bugs, enabling faster debugging and ultimately increasing the value of automation. Additionally, automated tests can be executed in different orders each day, with the entire process in the robot framework taking only a few hours and a single command to run the test cases. The study conducted in this paper demonstrates that automated testing using the robot framework significantly reduces testing time compared to manual testing.

[3] Qiu Na and Du Huaichang conducted a study titled "Enhancement of the Robot Framework and Its Application on Linux Servers." The research focuses on the significance of the Robot Framework, an open-source automation tool utilized for testing, and explores methods to expand its library. In order to fulfill the requirements of the specific application being tested, the Robot Framework can be enhanced through the integration of additional test libraries. The decision to employ the Robot Framework as the automation testing tool in this project was motivated by its excellent scalability and support for keyword- and data-driven techniques. Furthermore, it possesses the capability to test various types of clients or interfaces and can be utilized for distributed testing. The testing capabilities of the Robot Framework can be expanded by implementing test libraries in either Python or Java.

Conclusion

The Robot framework offers a keyword-driven approach for writing test cases, simplifying installation and facilitating test case creation and execution. It enables a seamless flow of test cases, including pre-conditions, actions, and verification, with the added benefit of easy-to-build keywords and a wide range of existing libraries. Additionally, developers can effortlessly compose their own test cases. By automating tasks, the Robot framework minimizes manual work and ensures accurate results.

References

[1] V. Debroy, L. Brimble, M. Yost and A. Erry, “Automating Web Application Testing from the Ground Up: Experiences and Lessons Learned in an Industrial Setting”, 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Vasteras, 2018.

[2] Q. Na and D. Huaichang, "Extension based on robot framework and application on Linux server”, 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2015.

[3] Qiu Na and Du Huaichang, “Extension based on robot framework and application on Linux server”

[4] IEEE Standard for Local and Metropolitan Area Network--Bridges and Bridged Networks”, in IEEE Std 802.1Q-2018.

[5] RFC 5517 - Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment.A

[6] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, Network Working Group, Request for Comments: 3261, June 2002

[7] M. Wahl, T. Howes, and S. Kille, Network Working Group Request for Comments:2251,Lightweight Directory Access Protocol (v3), December 1997

[8] H. Hakala, L. Mattila, J-P. Koskinen, M. Stura, and J. Loughney, Network Working Group Request for Comments: 4006, August 2005