

A Case Study on Effectiveness of Test Framework For API Testing

Surabhi Tripathi¹, Prof. R.N.Sharma²

¹Dept of CSE

²Asst. Professor, Dept of CSE

^{1, 2}Maharana Pratap College of Technology, Gwalior, M.P., India

Abstract- This study aims to work on a generalized data driven based test automation framework for API testing . Automation process is the matter of the few hour and it can reduces manual testing efforts . In the automation process we will consider any number of test cases and will run the same without human need. This must be done to check whether the application has been developed based on the Software requirements. This step is very crucial one as one must follow to the SRS while developing any application. In this paper, we will observe the performance of testing third party API on execution of large no of test cases using RESTASSURED in Eclipse IDE .It will give us a faster and smarter approach of testing & validating our API in the application . The manual testing process need to be automated so that the testing time , effort and money of organization can be reduced. The test automation framework is implemented for testing multiple test data from different sources like excel file, CSV file, Database(JDBC) & make a API request, then validate the data ,then writing the result to the file & lastly generate the report . Since RESTASSURED is flavor of java,several libraries have been imported to implement the test automation framework.

Keywords- API Testing, Test Automation Framework, Manual Testing, JSON Response,Test Automation Framework, API URL, Response Time, Actual Response Code

I. INTRODUCTION

The testing process is an essential step for any application before it is sent for delivery or in production phase. The testing process at initial phase is carried out manually, where a continuous availability of human effort is needed, to visually test the application or software. Testers need to prepare various document like test case document, test case scenarios and prepare input data. On a particular test case execution, the expected and the actual outcome is compared and if both are matched then the test is set to pass otherwise the test is failed but the results are entered in the document manually. The testers must repeat this test again and again with few modification in the application after each test, which also increases time, effort and cost.

Keeping this as a base, the concept of API testing is one where a request to be sent to the server from the client and in return response is captured and it is verified for errors. Format of request and response is in the JSON or in XML format. In API testing, the testing process is carried out by sending a input data and then observe the response body ,code and its time returned from the API and validating it with the expected response. After validation the results are entered for a test. The process of sending request then validate response and then observe result is need to be done manually for each test, which eventually increases time ,cost & effort,. This approach of manually test API becomes inefficient and useless when either any little modification occurs in the API or full functional behavior is changed. So keeping all these issues and difficulties in mind the testing process must be automated to improve efficiency and reliability and also to decrease manual effort, time and application cost. Once this testing process is automated ,test scripts can be easily re executed at any time and with any attempts and multiple cases can be tested with multiple test inputs. This will prove a faster, reliable and more accurate approach for API Testing.

II. RESEARCH PROBLEM

Testing proposed for finding out new error after modification in the code of software development. In other terms, it ensures that no other side effects were introduced in the process of fixing other issues and the software still works as it worked before. However, the execution of the full test suite for this purpose can be little expensive in the terms of cost and time. Therefore, it is essential to reduce the cost involved in testing. One way to achieve this is by test case automation. API Based automation testing contains too many challenges and problems for producing a error free APIs. testing requires a user interface for Testing an application which need to be test. The test scripts those are created separately are basically used by the automation framework for the validation for a specific test. The Automation Framework should be generic as much as possible so that it can be use in future for any application or on any platform which leads to minimize the effort not only of tester but of organization as well and thus save their time and cost. The test automation

can be scheduled for any time which can perform the regression tests on the application. The API test automation framework should produce results at the end of test suite with detailed information in the report about the passed and failed test cases. The actual values are captured in screenshot and saved by the framework for the verification by the testers .

III. RESEARCH METHODOLOGY

In order to find a suitable approach for testing a API, a framework is designed on Eclipse IDE with Maven which is a build automation tool used primarily for JAVA Projects, TestNG which is a testing framework designed to simplify the testing .The language used for automation is Java. After doing a complete study on all these tools we get an idea on how to integrate all these into one framework. In this research ,the Excel file containing data that are used for automation testing in validating API. This framework is designed in a generic way such that it exports results in a separate file but will store inside the framework structure after the automation is executed . All the validation functions are created and stored as a separate library file .

IV. IMPLEMENTATION

The Test automation framework is developed using RESTASSURED libraries getting test data from External Sources like excel file ,SQL Queries .The framework is responsible for getting the test data as input and automates the test scenario and produce test result in a report. In order to call a API or to make a request we need a API URI, method, and request data . The framework makes the request and gets response and then uses it for validation. In the extension of this , the framework takes necessary action, expected error code, expected response code data field from the input test data file & do the verification on the response returned from that API. The validation is done based on the value which is contained in the action column in the sheet. The validation functions are defined in a separate java file. Then the framework calls the validation function. API request values from the input test data file takes as a query parameter. The validation function then validates the parameters with the endpoint and then returns whether “pass” or “fail” to the framework. The validation function which validates the result by parsing the actual JSON response and then compares it with the expected JSON value and returns the outcome. Now the framework stores the actual responses like value, error code value, and response body returned from the type of request (GET,POST,DELETE,PUT)made and the result returned. The framework uses the API URI and other fields and it could execute 'n' no of test cases.

The framework reads the data row by row from the file and tests automatically until the input test data in test input file either excel file or database file ends. Now to take input from database the first thing which is to be done is to convert each SQL query output into JSON format. This can be achieved by adding a script in the query ‘JSON AUTO’. This script automatically converts the query into JSON format and the result will be compared with the API output. Now two inputs are taken ahead for the next step which is to compare both the inputs this can be carried out by using TestNG Assertion functions like 'assert'. This will make the process Data Driven. In this we can read the multiple inputs from the excel sheet ,here in our case i.e. the API URL and the file location of the SQL query output and then automate the whole process for 'n' number of test cases.

V. RESEARCH OUTCOME

At the end the result, the result file containing the actual values responded by the API and the validated result i.e. either pass or fail will be produced by the framework. The tester can observe the test cases result that are passed and analyze the failed result. The result file also contains some additional field with captured response time taken on the execution of each test case i.e the API in respond for a particular request which works as a verification for performance testing.

VI. CONCLUSION

The manual testing is automated to increase efficiency, reusability and to reduce manual testing effort & time. The process is automated with the help of a designed generic data driven framework ,which makes the complete process simple. At the end of execution process the framework produces test report which shows the detailed test report to identify and analyze the failed cases and store the result in a file with the reason which makes it fail. A descriptive document can be send to the respective authorized person in organization who can rectify the failed cases as earliest. Since the test report is generated in a good excel format, it is easy to present in a tabular format to another user. Since there might occur some changes in test data and may increase in API test cases, which can be easily done by updating or modifying the input test data. Thus this process increases the maintainability of testing. Any modification needed in the test data and in the expected result can be write directly in the input test data file. This test automation framework becomes very useful when a tester need to perform regression or Sanity testing where the already tested APIs need to be retest again to crosscheck any side effect in existing working functionality. The framework handles JSON response from the locally saved excel files as

well as if needed fetch the data from database as well which makes this as more generic. API's are using in almost every application for the user identity verification that need to be validated. This might create a huge number of complex test cases and hence manual testing can become cumbersome. Hence we need to automate the testing of third party API calls process and make it more feasible.

VII. ACKNOWLEDGEMENTS

I would like to express my gratitude and thanks to Prof. R.N. Sharma for all his help during this research work.

REFERENCES

- [1] Arcuri, A. (2017). Full API Automated Test Case Generation. 2017 IEEE International Conference on software Quality, Reliability and Security(QRS). doi:10.1109/qrs.2017.11
- [2] Asha K.R. and Shwetha, D.J., 2015. API Testing: Picking the Right Strategy. Pacific Northwest Software Quality Conference, p.1–20.
- [3] SunilL. Bangare, Seema Borse, Pallavi S. Bangare and Shital Nandedkar, 2012. Automated API testing approach. International Journal of Engineering Science and Technology (IJEST), 4,pp.673–676.
- [4] Xiong Zhen-hai, & Yang Yong-zhi. (2014). Automatic updating method based on Maven. 2014 9th International Conference on Computer Science & Education. doi:10.1109/iccse.2014.6926628
- [5] Sunil Bangare, Seema Borse, Pallavi S Bangare, Shital Nadedkar (2012). Automated Testing Approach, 465-469. doi: 10.1037/0893-3200.19.3.465
- [6] S. L. Bangare, A. R. Khare, P. S. Bangare, “Measuring the quality of Object oriented software Modularization: Defining metrics and algorithm”, International Journal on Computer Science and Engineering (IJCSE), ISSN: 0975-3397 Vol. 3