International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Automated Framework for API Testing

Sherine George[1], Nagaraja G.S[2]

[1]*Final Year Student, Department of CSE, R.V College of Engineering, Karnataka, India*
[2]*Professor and Associate Dean, Department of CSE, R.V College of Engineering, Karnataka, India*

Abstract - *This study aims to find out a more effective way of performing API testing. Automation is the need of the hour and it can help in reducing manual testing efforts by tenfold. The automation will consider any number of test scenarios and can run the same without human intervention. This must be done to check whether the application has been developed based on the requirements. This is very important as we must stick to the SRS while developing any application. By integrating many tools, we can achieve higher efficiency in the testing process. In this study we can see how to use Rest Assured for the process of automation and how we can validate the API calls without manual intervention. It gives us a faster and smarter approach of validating our Application in this case Xcellerate Risk Review Application.*

Keywords— API testing, XRR, Rest Assured, Cucumber, TestNG

1. INTRODUCTION

Today's world is a tech era where it is imperative for all businesses to have an online presence. APIs play a vital role for businesses in all industries. They help two or more softwares to communicate effectively. The cost of integrating disparate systems can be saved by using APIs. In order for any application to grow and reach a wider scale it is important that the application is able to connect to 3rd party applications or services. This is where APIs come to play. Nowadays, for any application it is easier to do API testing rather than UI testing. API testing is similar to testing software at the user interface level but instead of testing with user input and output, you use software to send calls and log the system's response. API test code must send a specific call to the API then log the actual and expected results.

The major issue in today's world is how to handle data effectively. With this increasingly large amount of data, it is important that we come up with a solution to handle it. Powerful analytics is required to handle this data. Similarly, there is large amount of data generated in clinical trials. For this data to be handled effectively Covance has an in house suite of tools called Xcellerate. Xcellerate suite has many tools like Xcellerate Study dashboard, Xcellerate Risk Review. Xcellerate Risk Review is a tool which highlights the risks involved in clinical trials and prioritizes them efficiently. The XRR has different colors to represent different categories of risks. The main goal of the project is to automate the API testing of XRR. The output of the API call must be validated against the XRR database. Automation testing has many benefits which include reliability, cost saving, improved efficiency. Once the process is automated scripts can be easily re executed and multiple cases can be checked. It provides a significantly faster and more accurate approach.

2. Ease of Use

Removing Manual Efforts involved in validating API Calls

The manual intervention involved in comparing the API call output with the SQL output can be completely removed. It makes the process more efficient and reliable.

Multiple Scenarios can be tested easily by using this approach

Numerous test cases can be specified in an external file which the automation framework must take and provide output in the form of reports.

3. Literature Review

In order to perform this automation, a framework is designed integrating Eclipse with Maven, Cucumber, Rest assured, TestNG. The language used for automation is Java. After proper research on all these tools we can get an idea on how to integrate it into one framework.

Rest Assured: There are many tools which can be used for API test automation but this study mainly focuses on Rest Assured. It is a Java based Library used to test RESTful web services. Since REST Assured is a Java library, integrating it into a continuous integration / continuous delivery setup is very easy. It can easily set up a HTTP connection, send a request and receive and parse a response. REST Assured makes your API testing more powerful with its ability to create data-driven tests.

TestNG: TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing to integration testing. It supports annotations which helps to group test cases accordingly. TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set.

Maven: It is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala and other languages. Maven addresses two aspects of building software: how software is built, and its dependencies. In comparison to

writing a simple Java project where you have to download all the jar files manually using Maven you just need to add that as a dependency and it will automatically download that particular file.

Cucumber: It is a testing tool that supports Behavior Driven Development (BDD). Cucumber Framework supports languages like beyond Ruby like Java, Scala, Groovy etc. It is a behavior driven development tool. Plugins in Cucumber works faster.

Rest API: It determines how the API looks like. It stands for Representational State Transfer. It is a set of rules that developers follow when they create their API. One of the rules is that when you specify a URL you must get specific data. Each URL is called a request while the data sent back to you is called a response.

4. Methodology

For API automation we need the above tools as specified. The IDE used is Eclipse and the programming language is Java. The main aim is to automate the process of checking whether the Application was developed based on the requirements or not. In order to achieve this SQL queries need to be written based on the requirements against the XRR database. The output of these SQL queries need to be checked with the XRR API call output i.e. this process needs to be automated.

The SQL query output and the API output are in different formats so the first thing to do is to convert both to the same format i.e. convert the SQL output to JSON format. After this is done the SQL output must be stored in a file so that it can be read by the automation framework

Once the SQL output is in JSON format we need to get the API call output by specifying the URL in the GET request. So the two inputs to the automation framework are supplied (SQL output and API call output).

The next step is to compare the two inputs to the framework. This can be done using Assertion functions provided by TestNG. Based on the comparison results have to be produced in a folder.
Now to make it data driven you need to add the feature of reading the two inputs which must be specified in an external source such as an excel file. This can be read using Cucumber. Further additional features can be added to the input and customize it according to our needs. This can be seen clearly in Fig 1.

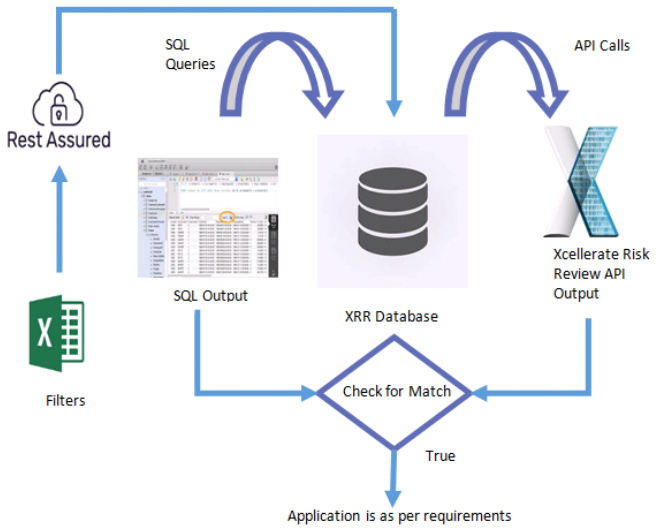

Fig 1. Methodology-Flow Diagram

5. Implementation

The first thing to be done is to convert each SQL query output into JSON format. This can be done by adding script to the query 'FOR JSON AUTO'. This will automatically convert the query into Json format so that it can be compared with the API output. This query then needs to be stored in a file.

This will act as the first input to the automation framework. The second input will be got using Rest Assured GET request, which is the API output. Refer Fig 2 for better understanding.

Once these two inputs are taken the next step is to compare them, this can be done by using TestNG Assertion functions.

The next step is to make the process Data Driven. For this we can read the 2 inputs from the excel sheet i.e. the API URL and the file location of the SQL query output and automate the process for any number of cases.

In Fig 2 we can see the entire process of how the automation is performed for n different test cases.

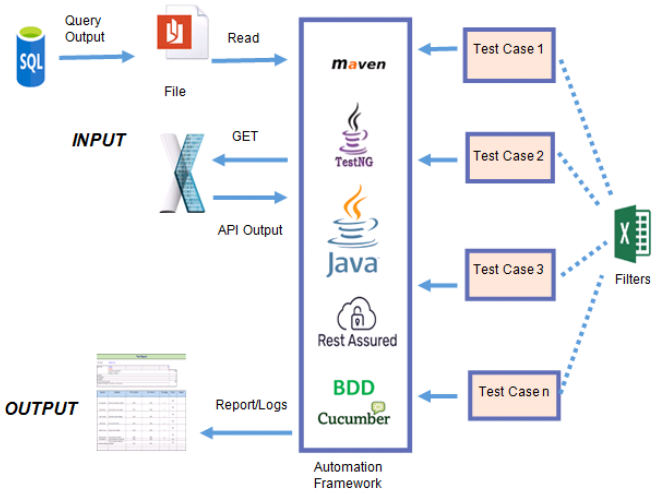

Fig 2. Implementation Details

6. CONCLUSION

Test Automation is widely used in today's corporate world. It helps in reducing time and has greater reliability. It helps in increasing the coverage of testing. Most of the software development projects have automated testing as an essential part of the software development phase. For any application there are multiple API calls which fetch data for the application. All these API calls need to be validated. This might include a huge number and hence manual testing can become cumbersome. Hence we need to automate the process and make it more feasible. By using this method, we can validate any number of API calls in no time without any manual intervention.

7. Future Work

With minor enhancements we can make this automation framework test any API call based on the filter which we apply. The base URL will remain the same, only the filter of the API Calls would change. This project is currently done keeping in mind the Xcellerate Risk Review Application. But with minor modifications the system can be used to automate any similar kind of Application which performs API testing.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my guide Dr Nagaraja G.S Professor of RVCE without whose constant guidance this would have never been possible. I would also like to thank Sailaja Surpani-Manager SQA at Covance under whom I worked on this project. I also wish to acknowledge the help and support given to me by Promod Balakrishnan-Associate Director and Siju Kunnambram-Support Manager IT Informatics at Covance. Special thanks to my parents, family members, peers and colleagues.

REFERENCES

1. Arcuri, A. (2017). RESTful API Automated Test Case Generation. 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). doi:10.1109/qrs.2017.11

2. 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

3. A. Ruiz, and Y. Price, "Test-Driven Development with TestNG and Abbot", IEEE Software, IEEE Computer Society, May/June 2007, pp. 51-57.

4. Ramya, P., Sindhura, V., & Sagar, P. V. (2017). Testing using selenium web driver. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). doi:10.1109/icecct.2017.8117878

5. Sunil Bangare, Seema Borse, Pallavi S Bangare, Shital Nadedkar (2012). Automated API 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

7. S. L. Bangare, A. R. Khare, P. S. Bangare, "Code parser for object Oriented software Modularization", International Journal of Engineering Science and Technology, ISSN: 0975-5462, Vol. 2 (12), 2010, 7262-7265.

8. S. L. Bangare, A. R. Khare, P. S. Bangare, "Quality measurement of modularized object oriented software using metrics", ACM-International Conference ICWET-2011 at Mumbai, ACM 978-1-4503-0449-8/11/02, ISBN: 978-1-4503-0449-8.

9. Wenhui, H., Yu, H., Xueyang, L., & Chen, X. (2017). Study on REST API Test Model Supporting Web Service Integration. 2017 IEEE 3rd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security *(IDS).* doi:10.1109/bigdatasecurity.2017.35