

API Testing and Automation

Pandiri Keerthana¹, Bulusu Satya Sai Bhanu Rutwik², Dr. Preethi Jeevan³

^{1,2} B.tech Scholars, Dept. of Computer Science and Engineering, Hyderabad-501301, India

³ Associate Professor, Dept. of Computer Science and Engineering, SNIST, Hyderabad 501301, India

Abstract: Automated API testing is an essential part of software development and security. This topic provides an overview of automated API testing, its benefits, and key related components. Automated API testing is the process of testing the use of application programming interfaces (APIs) to ensure functionality, reliability, and performance. APIs are the building blocks of modern software systems that enable communication and data exchange between various applications and devices. The summary highlights the benefits of automated API testing such as efficiency, faster time to market, less effort, and improved testing. By working on the testing process, organizations can run more efficient and complex testing, detect bugs early in the development cycle, and deliver quality APIs. The summary also discusses the key concepts of automated API testing. It highlights the importance of a robust testing framework that provides features such as request and authentication, data management, data evaluation, authentication and authorization support, and publishing and authoring capabilities. Integration with CI/CD pipelines and the ability to handle errors and exceptions are also important aspects of automated API testing. In summary, automated API testing plays an important role in ensuring the functionality, reliability, and performance of APIs. It has many benefits and requires a good evaluation framework and necessary products to streamline the testing process and deliver good software.

Index Terms—Application Programming Interfaces, API Automation, SOAP, RESTful APIs

I. Introduction

An application programming interface or API is a piece of code that allows two software programs to interact. Recently software applications are deployed on multiple servers

interacting with electronic devices to recover from related processes. Software applications provide some of their data and business logic through APIs whose ultimate goal is to generate additional revenue is known as API marketing. The number of APIs is increasing exponentially every year. As a result, automated testing of APIs has become an important part of the software development process because poorly functioning or unused APIs can reduce purchases and ultimately lost revenue. API testing is essential to ensure the functionality, reliability, security, and delivery of business logic. Key challenges in automated API testing include connecting API calls, matching failed responses, managing unplanned test cases, executing parallel APIs, and handling buffer errors. Document provides automated API testing tools and case studies of automated API testing.

II. Literature Survey (Background study)

a. Title- Importance of API Testing in Software Development:

Automated API testing is the process of testing application programming interfaces (APIs) using automated tools and techniques. APIs act as intermediaries, allowing different software to communicate and exchange data. Since APIs play an important role in modern software development, it is necessary to ensure their functionality, reliability and functionality. Automated API testing involves using special tools to send requests to the API endpoint, verify the received response, and verify that the API is working as expected. By using the testing process, organizations can achieve better results, save time and effort, and improve entire software systems.

The main purpose of automated API testing is to detect problems such as functional errors, poor performance, and security vulnerabilities early in the development cycle. By completing the testing process, developers and QA teams can quickly complete and repeat the testing process to ensure that the API meets the requirements in different scenarios. Automated API testing has many advantages. First, it provides

faster feedback and reduces the time required for regression analysis. Automated tests can be done more often, which makes it possible to find problems in a timely manner and fix errors easily. This helps accelerate the software development lifecycle and ensure timely delivery of high-quality software. Also, automated API testing provides better support. With automation tools, organizations can easily create and run a variety of tests involving different API functions, bugs, and performance scenarios. This extensive testing improves the reliability and robustness of the API.

By working on the testing process, developers can quickly check for changes to their code, while the QA team can focus on developing more complex scenarios and conducting investigations. Organizations often use specific benchmarks and tools to effectively implement automated API testing. This process provides the ability to participate in request and authentication, data evaluation, data management, authentication and authorization, and continuous/continuous integration additional (CI/CD) pipelines. In summary, automated API testing is an essential part of software development and ensures the functionality, reliability, and functionality of APIs. By leveraging automation tools and frameworks, organizations can achieve greater efficiency, improve evaluation, and quickly recommend high-quality software systems.

b.Title-Benefits of Automated API Testing:

API testing is essential to ensure functionality, reliability, security, and business logic delivery. Testing API is a time consuming and complex task. Because manual API tests are done by humans, there is a high risk of bugs and errors. Big data, book comparison impossible. Large organizations and real-time tasks are not suitable for manual API testing. Even a single faulty API can affect the entire product. Therefore, manual API testing is deprecated and automated API testing is preferred. In addition to speeding up the automated API testing process, also reduces the risk of errors.

Automated API testing does not require the presence of testers and can be done at any time. Web API is also divided into SOAP (Simple Object Access Protocol), RPC (Remote Procedure Call) and REST (Representational State Transfer) based on data exchange. REST or RESTful APIs are the most widely used APIs, so the API test tool is specific to RESTful APIs. The RESTful API is based on REST technology, which defines the communication protocol used in the development

of the Web Service. RESTful API uses HTTP methods such as GET, POST, DELETE and PUT. GET API is used to get resource. PUT API is used to transfer resources. POST API is used to create the resource. The DELETE API is used to delete resources. RESTful API responses include: program, version, version, date, timestamp, HTTP event code, message, info. The JSON (JavaScript Object Notation) format is used to send RESTful API responses. Most current testing software uses HTTP event codes to test domain. This method is not useful because the evaluator did not get the comparable response, which is important in the test. The API automated testing tool validates the API based on the expected JSON response provided by the tester, based on a comparison of with the actual JSON response from the server. This method removes threats associated with API responses and provides testers with more information about the API. In some software programs, API calls need to be executed in a specific order.

Parallel API execution interrupts API calls and invalidates tests. To avoid this confusion, the Request API Test tool analyzes continuous API calls and executes them in the order, not simultaneously. This automated API testing tool analyzes all individual API calls and makes them consistent. This automated API testing tool uses the banker algorithm to avoid any conflicts in API calls.

- Input Validation - Test the API with different input parameters and validate the response. Information for each set of input parameters, countries, languages, etc. should be. Does the API return the correct HTTP error code, such as 200 for invalid input parameters and 400 for invalid input parameters? JSON Format Validation - Verifies that the API's JSON or XML response is well-formed.
- Business Logic - Is the API doing what is supposed to do? Let's say we have a Load Balance API that provides the user with the current balance. If returns the correct HTTP country, but the balance of returned to the user is from months, the business logic of this API is broken.
- Negative Tests - Use false / false false, null / extra values, null values for fields that need to access the API and evaluate output. Will API gracefully handle unreasonably large payload data, special and non-ASCII characters, long strings and integers, the wrong type of data? Do you have a problem? How does

uptime behave in cases such as server failure? Exception holding mechanism in place? Is the error message clear and relevant?

- Reliability testing - Check if the API returns the correct response or if the response is inconsistent.
- Call Sequence Control - If the output of the API control includes changing the data structure, changing the state of the resource (eg DB file), executing the event, or calling other APIs, then should always work correctly.
- Security Assessment - Every business critical API must pass a security assessment to ensure that its code and services cannot be accessed and used by customers disagree.

III.Existing System

The proposed automated API testing tool is an easy to use tool that helps to automate the process of RESTful API testing.

Below mentioned are the major limitations of the existing system.

Limitations

- No support for code-less testing.
- No support for parallel execution of APIs while maintaining the orderliness of API calls.
- No support for handling unpredictable JSON response.
- No support for sequencing API calls.
- No support for pass/fail test case judgement.
- No support for scheduled tests.

IV. Proposed System

The proposed automated API testing tool is an easy to use tool that helps to automate the process of RESTful API testing.

Below mentioned are the major limitations of the existing system.

Limitations:

- No support for code-less testing.

- No support for parallel execution of APIs while maintaining the orderliness of API calls.
- No support for handling unpredictable JSON response.
- No support for sequencing API calls.
- No support for pass/fail test case judgement.
- No support for scheduled tests.

Evolution of API Testing

The application of API testing over the years has been driven by technological advancements, changing software development practices, and the growing importance of APIs in modern systems. Here is an overview of the key stages in the evolution of API testing:

1. Manual Testing: In the early stages, API testing was predominantly performed manually. Testers would manually construct API requests, send them to the API endpoints, and verify the responses. This approach was time-consuming, prone to errors, and limited in terms of test coverage.

2. Scripting-Based Testing: The next stage involved the use of scripting languages, such as JavaScript or Python, to automate API testing. Test scripts were created to send API requests, validate responses, and assert expected behaviors. This approach improved efficiency and test coverage to some extent, but it still required significant manual effort for script creation and maintenance.

3. Testing Frameworks and Tools: The advent of dedicated testing frameworks and tools specifically designed for API testing brought significant advancements. Frameworks such as Postman, SoapUI, and RestAssured provided intuitive interfaces, reusable test components, and features for request/response validation, data-driven testing, and test reporting. These tools simplified API testing and allowed for faster test creation and execution.

4. Shift to Test Automation: As organizations realized the benefits of test automation, API testing became an integral part of the test automation strategy. Test automation frameworks and tools were developed to integrate API testing seamlessly into continuous integration/continuous deployment (CI/CD) pipelines. This integration enabled the execution of API tests

as part of the build and deployment processes, ensuring the stability and reliability of APIs.

5. **Emphasis on Contract Testing:** With the rise of microservices architecture and the need for inter-service communication, contract testing gained prominence in API testing. Contract testing focuses on verifying the conformance of an API to its documented specification or contract. Tools such as Pact and Swagger helped automate contract testing, ensuring consistency between API documentation and implementation.

6. **Performance and Security Testing:** As APIs became critical components of highly scalable and secure systems, API testing expanded to include performance testing and security testing. Performance testing tools, such as JMeter and Gatling, enabled the simulation of heavy loads and measurement of response times. Security testing tools, such as OWASP ZAP and Burp Suite, provided capabilities to identify and address security vulnerabilities in APIs.

7. **Shift-Left Testing and API Virtualization:** API testing has embraced the shift-left testing approach, where testing is conducted early in the development lifecycle. This allows for early bug detection and faster feedback loops. Additionally, API virtualization tools like WireMock and Mountebank have emerged, enabling developers and testers to simulate API responses and dependencies during testing, even when actual APIs are unavailable.

8. **AI and ML in API Testing:** The future of API testing is likely to see the integration of artificial intelligence (AI) and machine learning (ML) techniques. AI-driven testing tools can analyze large volumes of API data, detect patterns, and suggest improvements in API performance and security. ML algorithms can generate realistic test data, optimize test coverage, and assist in test case creation.

Automated API testing tools address all major challenges in automated API testing, such as executing APIs simultaneously, processing JSON responses, tracking API calls, skipping/failing commands, and comparing responses. This tool also helps test takers. Therefore, the desire to use the API testing tool reduces the time required for by 90%, requires 95% fewer people, and reduces costs by 70% for compared to manual API testing.

V. Acknowledgement

For her insightful suggestion, knowledgeable counsel, and moral support during the writing of this work, we would like to express our gratitude to Dr. Preethi Jeevan.

VI. Conclusion

The recommended automated API testing tool is a fully automated RESTful API testing tool that addresses all the major challenges of API test automation and provides an easy to use GUI that supports codeless testing, parallel execution of APIs, identifier. unpredictable JSON responses, session API calls, pass/fail data processing, benchmark responses, and scheduled tests. The tool eliminates the job of API testing, so allows testers to understand API objects and responses. These tools help ensure performance, reliability, security, and business logic delivery of the requested an automated API testing tool to help make products more stable and deliver quality products to customers. Effective APIs lead to more products and ultimately increase revenue.

VII. References

- [1]. <https://www.linkedin.com/pulse/importance-api-testingankur-gupta/>
- [2]. <https://www.gcreddy.com/2014/01/disadvantages-of-manual-testing.html>
- [3]. <https://www.mulesoft.com/resources/api/types-of-apis>
- [4] "SofTReL Software Testing Guide Book Part1," by Ajitha and Amrit Shah
- [5] Authorization (<https://learning.getpostman.com/docs/postman/send-ingapi-requests/authorization/>), Postman Learning Centre
- [6] P.P. Ray (n.d.) (n.d.) (n.d.) Computer and Information Sciences Journal of King Saud University (2018) 30, 291–319
- [7]. Sunil L. Bangare, Seema Borse, Pallavi S. AUTOMATED API TESTING APPROACH. International Journal of Engineering Science and Technology (IJEST). BANGARE, SHITAL NANDEDKAR.

[8]. Asha KR and Shwetha DJ. API Testing: Picking the Right Strategy.

[9]. <https://www.linkedin.com/pulse/rest-api-testing-from-manual-approach-automation-saurabh-agarwal/>

[10]. <https://www.programmableweb.com/news/researchshows-interest-providing-apis-still-high/research/2018/02/23>