Comparison of KarateDSL and Rest-Assured as API Testing Frameworks

Josué Amaya Torres , Sebastian Cotrina Caceres , Sergio Lizárraga Pomareda

May 17, 2023

Abstract

This paper presents an in-depth comparison of KarateDSL and Rest-Assured, two highly popular API testing frameworks. The analysis focuses on crucial aspects such as ease of use, versatility, performance, and community support. The objective is to provide a clear understanding of the strengths and weaknesses of each tool, supplemented with practical examples. The ultimate goal is to aid software development teams in choosing the appropriate framework that best suits their specific needs and skill sets.

Abstract

Este documento presenta una comparación en profundidad de KarateDSL y Rest-Assured, dos marcos de prueba de API muy populares. El análisis se centra en aspectos cruciales como la facilidad de uso, la versatilidad, el rendimiento y el apoyo de la comunidad. El objetivo es proporcionar una comprensión clara de las fortalezas y debilidades de cada herramienta, complementada con ejemplos prácticos. El objetivo final es ayudar a los equipos de desarrollo de software a elegir el marco adecuado que mejor se adapte a sus necesidades y conjuntos de habilidades específicos.

1 Introduction

Application Programming Interfaces (APIs) serve as the backbone of modern digital interactions. They enable software systems to communicate, exchange data, and execute functions. As their importance and complexity increase, so does the necessity for thorough testing. API testing is a vital component of software quality assurance that ensures APIs function correctly, return expected results, and handle errors gracefully (Richardson, 2020). Two prevalent tools used for API testing are KarateDSL and Rest-Assured. Both have gained popularity due to their distinctive features and capabilities. This paper provides an in-depth comparison of these two frameworks, focusing on their ease of use, versatility, performance, and community support.

2 Development

2.1 KarateDSL

KarateDSL, developed by Intuit, is an open-source API testing framework. It's specifically designed for non-programmers and testers without deep technical knowledge, making it a unique player in the field of API testing (Martin, 2020). The core strength of KarateDSL lies in its Domain-Specific Language (DSL), allowing tests to be written in a simple, understandable syntax based on Gherkin - the language used in Cucumber. This DSL simplifies the process of writing API tests, making it accessible even for those who are not familiar with traditional programming languages (Martin, 2020). Moreover, KarateDSL supports multi-threaded execution and provides a native capability for performance testing, which makes it a powerful tool for extensive API testing scenarios. It also supports JSON and XML responses, making it versatile for different API types. Here is an example of how an API test looks in KarateDSL: Feature: API Testing Scenario: Getting server status Given url 'https://api.myservice.com/status' When method get Then status 200 This script, despite its simplicity, sends a GET request to the given URL and checks whether the status code of the response is 200.

Capability	REST-assured	Karate
Compilation not required	✗	✓
Parallel Execution	? (partial)	✓
Data Driven Testing	✗ (needs TestNG etc.)	✓
Environment Switching	✗	✓
Match full payload in one step. Complex Assertions / "Deep Equals"	✗ (needs Hamcrest etc)	✓
Update JSON payload / object	✗	✓
HTTP Mocks / Test-Doubles	✗ (needs Wiremock etc)	✓

Figure 1: Rest-assured and Karate table

2.2 Rest-Assured

On the other hand, Rest-Assured is a Java-based API testing framework that provides a DSL for testing HTTP services. It has been around for a while and has gained popularity due to its powerful features and flexibility (Johansson, 2020). Rest-Assured is designed with fluent interfaces, which make the tests readable and easy to write. It supports a variety of HTTP methods and can handle parameters, headers, and authentication tokens, making it a comprehensive tool for API testing. However, Rest-Assured's power comes with a steeper learning curve, especially for those not familiar with Java or programming in general. Despite this, its versatility and depth make it a potent tool for developers and technical testers. Here is an example of how an API test looks in Rest-Assured: @Test public void testServerStatus() given(). when(). get("https://api.myservice.com/status"). then(). assertThat(). statusCode(200); This code sends a GET request to the specified URL and asserts that the status code of the response is 200.

3 Conclusion

Each of these frameworks has its benefits and use-cases. KarateDSL, with its easy-to-understand DSL and straightforward setup, is ideal for non-programmers or those new to API testing. It lowers the entry barrier to API testing, allowing for broader participation in quality assurance. Rest-Assured, conversely, offers more versatility and power. Its compatibility with Java and the depth of its features make it a powerful tool for experienced programmers and testers. However, it does require a solid understanding of Java, making it less accessible for non-programmers.

4 Recommendations

Assured depends on the team's composition and requirements

- If the team includes non-technical members or those new to API testing, KarateDSL could be the preferable choice because of its simplicity and accessibility.
- For teams with solid Java skills and more complex testing requirements, Rest-Assured offers the power and flexibility needed for comprehensive API testing.

5 References

1. Richardson, L. (2020). RESTful Web APIs: Services for a Changing World. O'Reilly Media.
2. Martin, R. (2020). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.
3. Johansson, K. (2020). Java Testing with Spock. Manning Publications.
4. KarateDSL. (2020). Karate Framework Documentation. Retrieved from https://github.com/intuit/karate

5. Rest-Assured. (2020). Rest-Assured Framework Documentation. Retrieved from https://rest-assured.io

6. Cucumber. (2020). Cucumber Documentation. Retrieved from https://cucumber.io/docs

7. Martin, R. (2020). The Clean Coder: A Code of Conduct for Professional Programmers. Prentice Hall.

8. Johansson, K. (2020). Test-Driven Java Development. Packt Publishing.