

ToiletPaper #103

Schlagkräftige HTTP End-to-End Tests mit Karate

Autor: Alexandros Panagiotidis / Senior Software Architect / Standort Stuttgart

✗ Problem

E2E-Testing von HTTP-Endpunkten kann aufwendig werden, insbesondere wenn man mit elaborierten JSON- oder XML-Requests hantieren muss. Klar gibt es einige Frameworks wie [Postman](#), [Rest-Assured](#) oder [Spring MockMvc](#), aber die bringen zusätzliche Probleme wie Nachvollziehbarkeit von Änderungen, Aufsetzen der Umgebung, Verwalten der Testdaten, Ausführung der Tests und dem zugehörigen Reporting mit sich.

✓ Lösung

[Karate](#) ist ein Framework, um E2E-Tests geschmeidig umzusetzen mit direkter Unterstützung von [JSON](#) und [XML](#) für Requests und Validierung der Antworten sowie [Interoperabilität mit Java](#) und [JavaScript](#). Testfälle werden [semi-natürlich-sprachlich in Gherkin](#) definiert – bei guten POs fallen diese direkt aus den Stories als Akzeptanzkriterien heraus! Testergebnisse werden in [Reports](#) zusammengefasst, welche den HTTP-Dialog und die Assertions enthalten. Mit ein wenig Zusatzaufwand kann man [Tests wiederverwenden](#) und Antworten vorhergehender Tests verwenden. Zusätzlich bringt Karate einen [Mock-Server](#) mit und kann sogar [Lasttests](#) durchführen.

➔ Beispiel

Im folgenden Beispiel wird eine XML-Antwort überprüft, bei der die Reihenfolge keine Rolle spielt (vergleicht dazu mal die tatsächliche Antwort von [mocky.io](#) mit der Erwartung im Test):

```
Feature: jambit API
```

```
Scenario: Mission und Motto gemeinsam überprüfen
  Given url "http://www.mocky.io/v2/5bd9ab282f00005c0006d262"
  And header Accept = "application/xml"
  When method get
  Then status 200
  And match response contains
    ""
```

```
<jambit>
<mission>100% Begeisterung</mission>
<motto>Where innovation works</motto>
</jambit>
    ""
```

Hier eine Testsuite, bei der zunächst die URL für die weiteren Testfälle gesetzt wird. Der parametrisierte Test überprüft stichprobenartig drei Erwartungen:

```
Feature: Users API
```

```
Background:
  * url "https://jsonplaceholder.typicode.com/"
```

```
Scenario Outline: Given user ID, validate company name
  Given path "users", <id>
  When method get
  Then status 200
  And match response.company.name == "<companyName>"
```

```
Examples:
```

```
| id | companyName |
| 1  | Romaguera-Crona |
| 3  | Romaguera-Jacobson |
| 7  | Johns Group |
```

+ Weiterführende Aspekte

- [Karate auf GitHub](#)
- [Gherkin Syntax](#)
- [Acceptance test-driven development bei Wikipedia](#)