

Automatisiertes Testen mit Karate

Datum	16.09.2025
Ort	Hannover
Verfasser	Daniel Vergien

Agenda

1. Über mich
2. Überblick über Karate
3. Browser-Tests
4. API-Tests
5. Tests organisieren
6. Erweiterung
7. Last-Tests
8. Demo
9. Fragen

Über mich

- Daniel Vergien
-  @dve@norden.social
-  dve@vergien.net
- Maurer Electronics / Bundesdruckerei seit 2019
- Java Entwickler seit ca. 15 Jahren
 - Backend
 - Ein wenig Angular
 - Viel Tests und Automatisierung
- Orga-Team javaland4kids, java-forum-nord4kids, kids4it

Karate im Überblick

Überblick

- <https://karatelabs.github.io/karate/>
- MIT-Lizenz
- Version: 1.5.1, 1.5.2-RC2
- Erstes Release auf Github im Februar 2017

Was deckt Karate ab?

- API-Testing
- Webbrowser Automatisierung
- Last-Tests
-
- API-Test doubles (Mocks)
- Desktop Automatisierung

Wieso sollte ich Karate nutzen?

- Einheitliche Beschreibungssprache für Testfälle (Gherkin)
- Wiederverwendung von API-Tests für Last-Tests
- Einheitliche Konfiguration für verschiedene Testarten
- Basiert auf etablierten Techniken
 - Junit 5
 - GraalVM
 - Webdriver/Chromedriver Protokoll
 - Gatling

UI-Tests

Testbeschreibung für einen UI-Test

```
Run
1 Feature: owners api
2
3 Background:
4
5 Run
6 Scenario: Add a new owner
7   Given driver frontendUrl
8   And click('li.dropdown:nth-child(2) > a:nth-child(1)')
9   And click('li.dropdown:nth-child(2) > ul:nth-child(2) > li:nth-child(2) > a:nth-child(1) > span:nth-child(2)')
10  And input('#firstName', 'Hans')
11  And input('#lastName', 'Gruber')
12  And input('#address', 'Lange Str. 7')
13  And input('#city', 'Berlin')
14  And input('#telephone', '0301234567')
15  When click("button.btn:nth-child(2)")
    Then match driver.url == frontendUrl + '/petclinic/owners'
```

Gherkin DSL

- Schlüsselworte wie:
 - driver
 - input
 - click
 - waitForUrl
 - screenshot
 - pdf

API-Tests

Gherkin DSL

- Schlüsselworte wie:
 - url
 - path
 - method
 - status
 - soap action
- Für Requests:
 - param
 - header
 - cookie
- Für responses:
 - responseStatus
 - responseCookies
 - responseHeaders

```
Run
1 Feature: owners api
2
3 Background:
4   * url backendUrl
5
6 Run
7 Scenario: Add a new owner
8   * def owner =
9     """
10    {
11      "firstName": "George",
12      "lastName": "Franklin",
13      "address": "110 W. Liberty St.",
14      "city": "Madison",
15      "telephone": "6085551023"
16    }
17    """
18   Given path 'api', 'owners'
19   And request owner
20   When method post
21   Then status 201
22   And match response contains owner
23   And match response contains {id: '#nonnull'}
```

Tests verwalten

Tags und Reports

- Tags
- JUNIT-XML
- Cucumber JSON
- Gatling

```
@requirement=PET-2  
@test=PET-23|  
Scenario: Add a new owner
```

Karate erweitern

Möglichkeiten zur Erweiterung

- Aufruf von andern Features
- Aufruf von JavaScript
- Aufruf von Java

```
* def owner = call read('classpath:examples/api/randomOwner.feature@random-user')
```

```
* def ownerUtils = Java.type('examples.api.OwnersUtils')  
* def owner = {}  
* set owner.firstName = ownerUtils.firstName();
```

Last-Tests

Last-Tests mit Karate

- Karate wird in Gatling integriert
 - ProtocolBuilder
 - ActionBuilder
 - PauseBuilder
- Einzelne Features aus den Tests werden als Gatling Action ausgeführt
- Seit 1.5.0 auch Java-DSL

```
package examples.perf;

import static com.intuit.karate.gatling.javaapi.KarateDsl.*;
import static io.gatling.javaapi.core.CoreDsl.*;

import io.gatling.javaapi.core.*;

public class OwnersSimulation extends Simulation {
    public OwnersSimulation() {
        ProtocolBuilder protocol = karateProtocol();
        ScenarioBuilder create = scenario(name:"create").exec(karateFeature(name:"classpath:examples/api/owners2.feature"));

        setUp(create.injectOpen(rampUsers(users:10).during(durationSeconds:5)).protocols(protocol));
    }
}
```

Demo

Zeit für Fragen

- <https://gitlab.com/dve/karate-talk>

