

Tester ses APIs web avec



Karate



Jean-Baptiste Cotard

 @jbcotard

Développeur

 sesam-vitale



@agilemans



@gdglemans

Le Mans School of IA

@IALeMans

Framework nativement orienté web



- Comprend les verbes HTTP
- REST, SOAP
- JSON, XML (et CSV, YAML)

Un DSL basé sur Gherkin

Scenario: create and retrieve a cat

Given url 'http://myhost.com/v1/cats'

And request { name: 'Billie' }

When method post

Then status 201

And match response == { id: '#notnull', name: 'Billie' }

Given path response.id

When method get

Then status 200

JSON is 'native'
to the syntax

Intuitive DSL
for HTTP

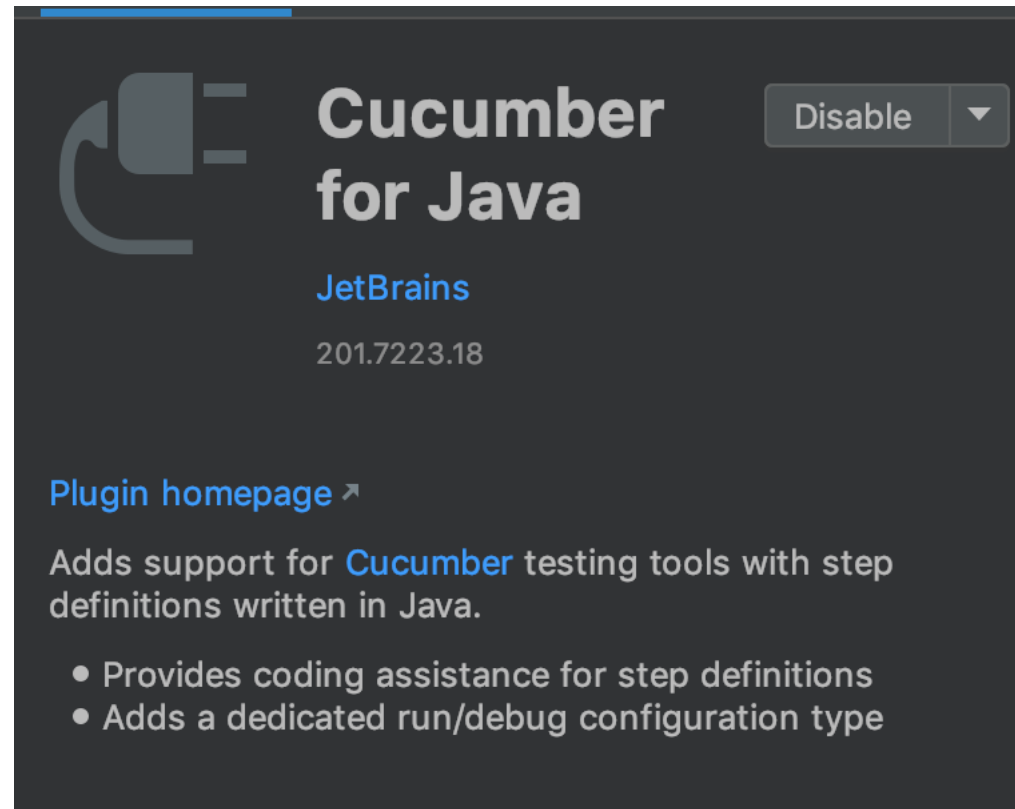
Payload
assertion in
one line

Second HTTP
call using
response data



Un premier exemple

Support IDE - IntelliJ



The image shows the 'Cucumber for Java' plugin interface within the IntelliJ IDEA IDE. The interface is dark-themed. On the left is the Cucumber logo, a stylized 'C' with a plug. To its right, the text 'Cucumber for Java' is displayed in a large, bold, white font. Below this, 'JetBrains' is written in a smaller blue font, and the version number '201.7223.18' is shown in a small grey font. In the top right corner of the plugin panel, there is a 'Disable' button and a small downward-pointing triangle icon. Below the version information, there is a link 'Plugin homepage' with an external link icon. A descriptive paragraph follows: 'Adds support for Cucumber testing tools with step definitions written in Java.' At the bottom, there is a bulleted list of features.

Cucumber for Java

JetBrains

201.7223.18

[Plugin homepage](#)

Adds support for **Cucumber** testing tools with step definitions written in Java.

- Provides coding assistance for step definitions
- Adds a dedicated run/debug configuration type



Un premier exemple - feature

```
Feature: recuperation simple d'un produit
```

```
Scenario: recupere la liste des produits
```

```
Given url "http://localhost:9090/Produits"
```

```
When method get
```

```
Then status 200
```

```
And match response[0] ==
```

```
"""
```

```
{"id":1,"nom":"Ordinateur portable","prix":350,"prixAchat":120}
```

```
"""
```



Un premier exemple - résultat

▼ ✓ Test Results	229 ms
▼ ✓ [recup-produit-simple] recuperation simple d'un produit	229 ms
✓ [1:3] recupere la liste des produits	229 ms

```
=====
elapsed:   0,84 | threads:    1 | thread time: 0,27
features:    1 | ignored:    0 | efficiency: 0,32
scenarios:   1 | passed:     1 | failed: 0
=====
```

```
1 > GET http://localhost:9090/Produits
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Host: localhost:9090
1 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)

1 < 200
1 < Content-Type: application/json;charset=UTF-8
1 < Date: Thu, 23 Jul 2020 09:44:44 GMT
1 < Transfer-Encoding: chunked
[{"id":1,"nom":"Ordinateur portable","prix":350,"prixAchat":120},{"id":2,"nom":"Tablette tactile","prix":250,"prixAchat":80},{"id":3,"nom":"Smartphone","prix":450,"prixAchat":150},{"id":4,"nom":"Montre connectée","prix":150,"prixAchat":50},{"id":5,"nom":"Casque audio","prix":100,"prixAchat":30},{"id":6,"nom":"Appareil photo","prix":600,"prixAchat":200},{"id":7,"nom":"Console de jeu","prix":300,"prixAchat":100}, {"id":8,"nom":"Lecteur MP3","prix":80,"prixAchat":20}, {"id":9,"nom":"Chargeur sans fil","prix":40,"prixAchat":10}, {"id":10,"nom":"Câble USB","prix":10,"prixAchat":2}]]
```




Exemples avec plusieurs requêtes

Cinématique simple



```
Feature: recuperation d'un produit

Scenario: recupere un produit par son id

    * def produit =
        """
        {
            "nom": "mon Produit",
            "prix": 10,
            "prixAchat": 100
        }
        """

    Given url 'http://localhost:9090/Produits'
    And request produit
    When method post
    Then status 201

    * def urlProduit = responseHeaders['Location'][0]
    * print 'created produit : ', urlProduit

    Given url urlProduit
    When method get
    Then status 200
```

Cinématique simple - résultat



```
1 > POST http://localhost:9090/Produits
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Content-Length: 47
1 > Content-Type: application/json; charset=UTF-8
1 > Host: localhost:9090
1 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)
{"nom":"mon Produit","prix":10,"prixAchat":100}
1 < 201
1 < Content-Length: 0
1 < Date: Thu, 23 Jul 2020 09:56:53 GMT
1 < Location: http://localhost:9090/Produits/21
2 > GET http://localhost:9090/Produits/21
2 > Accept-Encoding: gzip,deflate
2 > Connection: Keep-Alive
2 > Host: localhost:9090
2 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)
2 < 200
2 < Content-Type: application/json; charset=UTF-8
2 < Date: Thu, 23 Jul 2020 09:56:53 GMT
2 < Transfer-Encoding: chunked
{"id":21,"nom":"mon Produit","prix":10,"prixAchat":100}
```

Cinématique / réutilisation



```
Feature: recuperation d'un produit

Background:
    * def creator = read('create-produit-test.feature')

Scenario: recupere le produit test apres l'avoir créé
    * def result1 = call creator
    * def urlProduit = $result1.responseHeaders['Location'][0]
    * print 'created produit : ', urlProduit

    Given url urlProduit
    When method get
    Then status 200
```

Cinématique / réutilisation



```
@ignore
Feature:

    Scenario: cree le produit test
        * def produit =
            """
            {
                "nom": "test",
                "prix": 100,
                "prixAchat": 120
            }
            """

        Given url 'http://localhost:9090/Produits'
        And request produit
        When method post
        Then status 201
```

Cinématique / réutilisation - résultat



Karate

```
12:05:19.647 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - request:
1 > POST http://localhost:9090/Produits
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Content-Length: 41
1 > Content-Type: application/json; charset=UTF-8
1 > Host: localhost:9090
1 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)
{"nom":"test","prix":100,"prixAchat":120}

12:05:19.672 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - response time in milliseconds
1 < 201
1 < Content-Length: 0
1 < Date: Thu, 23 Jul 2020 10:05:19 GMT
1 < Location: http://localhost:9090/Produits/22

12:05:19.680 [ForkJoinPool-1-worker-3] INFO com.intuit.karate - [print] created produit :
12:05:19.682 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - request:
1 > GET http://localhost:9090/Produits/22
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Host: localhost:9090
1 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)

12:05:19.687 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - response time in milliseconds
1 < 200
1 < Content-Type: application/json; charset=UTF-8
1 < Date: Thu, 23 Jul 2020 10:05:19 GMT
1 < Transfer-Encoding: chunked
{"id":22,"nom":"test","prix":100,"prixAchat":120}
```

Cinématique / paramètres



```
Feature: recuperation d'un produit

  Background:
    * def getter = read('get-produit.feature')

  Scenario: recupere le produit par son id
    * table produits
      | id |
      | 2 |
      | 1 |
      | 3 |
    * def result1 = call getter produits
```

Cinématique / paramètres



```
@ignore
Feature:

    Scenario: recupere un produit

        Given url 'http://localhost:9090/'
        And path 'Produits', id
        When method get
        Then status 200

|
```


Cinématique / paramètres - résultat



Karate

```
12:09:31.239 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - request:
1 > GET http://localhost:9090/Produits/2
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Host: localhost:9090
1 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)

12:09:31.265 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - response time in milliseconds: 24,84
1 < 200
1 < Content-Type: application/json;charset=UTF-8
1 < Date: Thu, 23 Jul 2020 10:09:31 GMT
1 < Transfer-Encoding: chunked
{"id":2,"nom":"Aspirateur Robot","prix":500,"prixAchat":200}

12:09:31.271 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - request:
1 > GET http://localhost:9090/Produits/1
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Host: localhost:9090
1 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)

12:09:31.274 [ForkJoinPool-1-worker-3] DEBUG com.intuit.karate - response time in milliseconds: 2,79
1 < 200
1 < Content-Type: application/json;charset=UTF-8
1 < Date: Thu, 23 Jul 2020 10:09:31 GMT
1 < Transfer-Encoding: chunked
{"id":1,"nom":"Ordinateur portable","prix":350,"prixAchat":120}
```

Fonctions avancées

Reporting - nativement



Test Suite Navigation

of failed tests: 0/10
(0.00%)

of skipped tests: 0/10
(0.00%)

of passed tests: 10/10
(100.00%)

1	2	3	4	5	6
7	8	9	10		

Scenario: [1:3] recupere un produit par son id

Test 1 : * def produit =	0.00008
--------------------------	---------

Test 2 : Given url 'http://localhost:9090/Produits'	0.000094
---	----------

Test 3 : And request produit	0.00001
------------------------------	---------

Test 4 : When method post	0.007693
---------------------------	----------

Test 5 : Then status 201	0.00001
--------------------------	---------

Test 6 : * def urlProduit = responseHeaders['Location'][0]	0.015446
--	----------

Test 7 : * print 'created produit : ', urlProduit	0.000712
---	----------

Test 8 : Given url urlProduit	0.000012
-------------------------------	----------

Test 9 : When method get	0.00431
--------------------------	---------

```
12:15:53.754 request:
2 > GET http://localhost:9090/Produits/27
2 > Accept-Encoding: gzip,deflate
2 > Connection: Keep-Alive
2 > Host: localhost:9090
2 > User-Agent: Apache-HttpClient/4.5.11 (Java/14)
```

```
12:15:53.757 response time in milliseconds: 2,73
2 < 200
2 < Content-Type: application/json;charset=UTF-8
2 < Date: Thu, 23 Jul 2020 10:15:53 GMT
2 < Transfer-Encoding: chunked
{"id":27,"nom":"mon Produit","prix":10,"prixAchat":100}
```

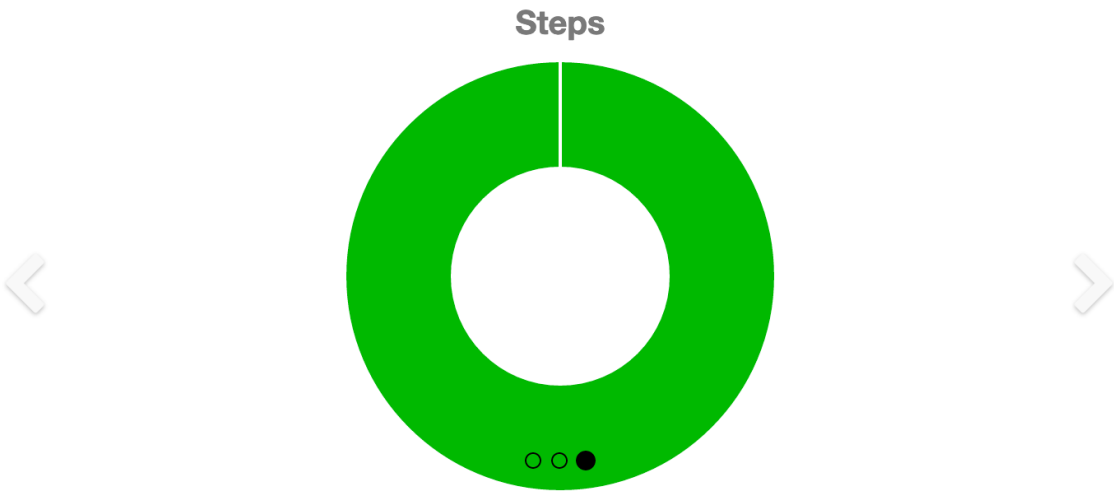
Test 10 : Then status 200	0.000009
---------------------------	----------



Reporting – avec cucumber-reporting

Features Statistics

The following graphs show passing and failing statistics for features



	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
examples/produits/1_recup-produit-simple.feature	4	0	0	0	0	4	1	0	1	259ms	Passed
examples/produits/3_recup-produit-reuse1.feature	13	0	0	0	0	13	1	0	1	508ms	Passed
examples/produits/4_recup-produit-reuse2.feature	18	0	0	0	0	18	1	0	1	533ms	Passed
examples/produits/2_recup-produit.feature	10	0	0	0	0	10	1	0	1	279ms	Passed
4	45	0	0	0	0	45	4	0	4	1s 580ms	
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%



Autres fonctions intéressantes

- Appel de fonctions Java et Javascript
- Mock , hooks
- Websocket/ async
- Test de performance, test UI, ...

Start	Maven Gradle Quickstart Standalone Executable Naming Conventions Script Structure
Run	JUnit 4 JUnit 5 Command Line IDE Support Tags / Grouping Parallel Execution Java API
Report	Configuration Environment Switching Reports JUnit HTML Report Logging
Types	JSON XML JavaScript Functions Reading Files Type / String Conversion Floats and Integers Embedded Expressions JsonPath XPath Karate Expressions
Variables	def text table yaml csv string json xml xmlstring bytes copy
Actions	assert print replace get set remove configure call callonce eval read() karate API
HTTP	url path request method status soap action retry until
Request	param header cookie form field multipart file multipart field multipart entity params headers cookies form fields multipart files multipart fields
Response	response responseBytes responseStatus responseHeaders responseCookies responseTime responseType requestTimeStamp
Assert	match == match != match contains match contains only match contains any match !contains match each match header Fuzzy Matching Schema Validation contains short-cuts
Re-Use	Calling Other *.feature Files Data Driven Features Calling JavaScript Functions Calling Java Code Commonly Needed Utilities Data Driven Scenarios
Advanced	Polling Conditional Logic Before / After Hooks JSON Transforms Loops HTTP Basic Auth Header Manipulation GraphQL Websockets / Async call vs read()
More	Mock Servlet Test Doubles Performance Testing UI Testing Desktop Automation VS Code / Debug Karate vs REST-assured Karate vs Cucumber Examples and Demos



Conclusion



<https://github.com/intuit/karate>