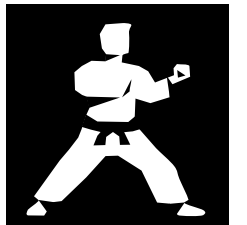


# Karate

## Open Source API Test Automation

**Peter Thomas** | Distinguished Engineer | Intuit

@ptrthomas | @KarateDSL



# Karate

[github.com/intuit/karate](https://github.com/intuit/karate)

**Intuit Open Source**

**Released Feb-2017**

**A DSL for writing web-service acceptance tests**

**Inspired by Cucumber | BDD syntax**



# Karate enters the ThoughtWorks Tech Radar - April 2019

The screenshot shows a browser window with the URL <https://www.thoughtworks.com>. The page displays the 'ASSESS' section of the Tech Radar, which is a quadrant chart divided into four stages: ADOPT, TRIAL, ASSESS, and HOLD. The chart contains various technologies represented by numbered triangles. Karate is highlighted with a black callout box and is positioned at rank 91, located in the ASSESS quadrant. A legend in the bottom right corner indicates that a triangle symbol represents 'New or moved' and a circle represents 'No change'.

**ASSESS** ?

- 83. Aeron
- 84. Arrow
- 85. Chaos Toolkit
- 86. Dask
- 87. Embark
- 88. fastai
- 89. http4k
- 90. Immer
- 91. Karate**

Given our experience that tests are the only API specifications that really matter, we're always on the lookout for new tools that might help. **Karate** is an API testing framework whose unique feature is that tests are written directly in Gherkin without relying on a general-purpose programming language to implement test behavior. Karate is really a domain-specific language for describing HTTP-based API tests. Although this approach is

ADOPT TRIAL ASSESS HOLD

▲ New or moved  
● No change

# Hello World

**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`

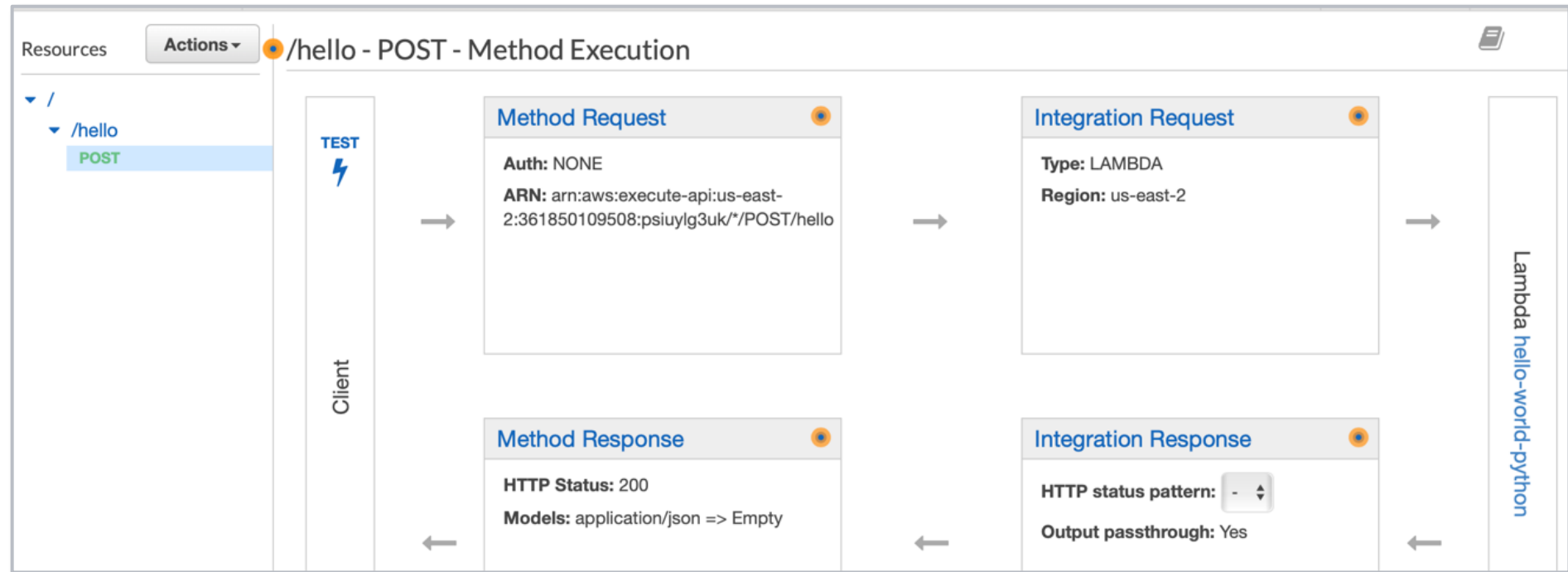
JSON is 'native'  
to the syntax

Intuitive DSL  
for HTTP

Payload  
assertion in  
one line

Second HTTP  
call using  
response data

# Demo: AWS Lambda



```
import json
```

```
def lambda_handler(event, context):  
    print("Received event: " + json.dumps(event, indent=2))  
    return { "message": "Hello " + event["name"] + "!" }
```

```
Given url 'https://xxx.amazonaws.com/default'
```

```
And path 'hello'
```

```
And request { name: 'Bob' }
```

```
When method post
```

```
Then status 200
```

```
And match response == { message: 'Hello Bob!' }
```

# Java vs JSON

```
package com.intuit.karate.demo.domain;

import java.util.ArrayList;
import java.util.List;

public class Cat {

    private int id;
    private String name;
    private List<Cat> kittens;

    public void addKitten(Cat kitten)
        if (kittens == null) {
            kittens = new ArrayList<>()
        }
        kittens.add(kitten);
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name)
        this.name = name;
    }

    public List<Cat> getKittens() {
        return kittens;
    }

    public void setKittens(List<Cat>
        this.kittens = kittens;
    }
}
```

```
Cat billie = new Cat();
billie.setName("Billie");
Cat bob = new Cat();
bob.setId(23);
bob.setName("Bob");
billie.addKitten(bob);
Cat wild = new Cat();
wild.setId(42);
wild.setName("Wild");
billie.addKitten(wild);
```

```
* def billie =
.....
{
    name: 'Billie',
    kittens: [
        { id: 23, name: 'Bob' },
        { id: 42, name: 'Wild' }
    ]
}
.....
```

```
* match billie.kittens contains { id: '#? _ > 25', name: '#string' }
```

```
private static boolean hasKitten(Cat cat, Cat kitten) {
    if (cat.getKittens() != null) {
        for (Cat kit : cat.getKittens()) {
            if (kit.getId() == kitten.getId()) {
                if (kit.getName() == null) {
                    if (kitten.getName() == null) {
                        return true;
                    }
                } else if (kit.getName().equals(kitten.getName())) {
                    return true;
                }
            }
        }
    }
    return false;
}
```

# Validation examples: JsonPath & RegEx

Type	Directive
Ignore / Exists	#ignore   #null   #notnull
Primitive Type	#boolean   #number   #string
Complex Type	#array   #object
Special	#uuid
RegEx	#regex <regex string>
Predicate	#? <expression>

```
Given def cat =
  """"
  {
    name: 'Billie',
    kittens: [
      { id: 23, name: 'Bob' },
      { id: 42, name: 'Wild' }
    ]
  }
  """"
```

Then match cat.kittens[\*].id == [23, 42]

Then match cat.kittens[\*].id contains 23

Then match cat.kittens[\*].id contains [42, 23]

Then match cat..name == ['Billie', 'Bob', 'Wild']

Then match each cat.kittens contains { id: '#number' }

Then match each cat.kittens == { id: '#notnull', name: '#regex [A-Z][a-z]+' }

\* def isLessThanFifty = function(x) { return x < 50 }

Then match each cat.kittens contains { id: '#? isLessThanFifty( )' }

# Demo

Cucumber Reports - Features

file:///Users/pthomas3/dev/zcode/karate/karate-demo/target/cucumber-html-r 80%

Cucumber Report Features Tags Steps Failures

## Features Statistics

The following graphs show passing and failing statistics for features

Features

Feature	Steps					Total	Scenarios			Duration	Status
	Passed	Failed	Skipped	Pending	Undefined		Passed	Failed	Total		
demo/callarray/call-json-array.feature	52	0	0	0	0	52	1	0	1	1s 539ms	Passed
demo/calldynamic/call-dynamic-json.feature	66	0	0	0	0	66	1	0	1	1s 737ms	Passed
demo/callfeature/call-feature.feature	27	0	0	0	0	27	1	0	1	1s 366ms	Passed
demo/callfeature/create-two-cats.feature	11	0	0	0	0	11	1	0	1	1s 256ms	Passed
demo/callonce/call-once.feature	42	0	0	0	0	42	3	0	3	1s 576ms	Passed
demo/calltable/call-table.feature	53	0	0	0	0	53	1	0	1	571ms	Passed
demo/cats/cats-xml.feature	14	0	0	0	0	14	1	0	1	353ms	Passed
demo/cats/cats.feature	12	0	1	0	0	13	0	1	1	148ms	Failed
demo/cats/kittens.feature	22	0	0	0	0	22	1	0	1	247ms	Passed
demo/cats/syntax-demo.feature	3	0	0	0	0	3	1	0	1	062ms	Passed
demo/delete/delete.feature	19	0	0	0	0	19	2	0	2	186ms	Passed
demo/encoding/encoding.feature	9	0	0	0	0	9	2	0	2	216ms	Passed
demo/greeting/greeting.feature	11	0	0	0	0	11	2	0	2	208ms	Passed
demo/headers/call-isolated-config.feature	45	0	0	0	0	45	5	0	5	545ms	Passed

Cucumber Reports - Feature

file:///Users/pthomas3/dev/zcode/karate/karate-demo/target/cucumber-html-r 80%

Cucumber Report Features Tags Steps Failures

## Feature Report

Feature	Steps					Total	Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined		Passed	Failed	Total	Duration	Status
demo/callfeature/call-feature.feature	27	0	0	0	0	27	1	0	1	1s 366ms	Passed

Feature demo/callfeature/call-feature.feature

calling another feature file

Background 654ms

Steps

- \* url demoBaseUrl 654ms

Scenario create kittens and then create parent cat 712ms

Steps

- \* call create-two-cats.feature 000ms
- \* url demoBaseUrl 000ms
- Given path 'cats' 000ms
- And request { name: 'Bob' } 000ms
- When method post 000ms

Doc string

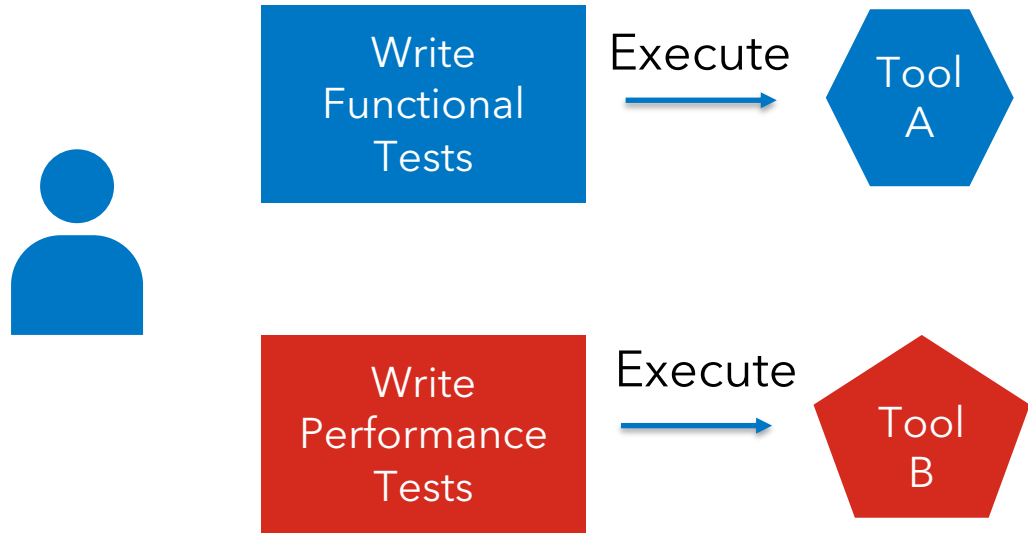
```

08:10:05.182 DEBUG -
1 > POST http://127.0.0.1:58317/cats
1 > Accept-Encoding: gzip,deflate
1 > Connection: Keep-Alive
1 > Content-Length: 14
1 > Content-Type: application/json
1 > Host: 127.0.0.1:58317
1 > User-Agent: Apache-HttpClient/4.5.3 (Java/1.8.0_112)
{"name":"Bob"}

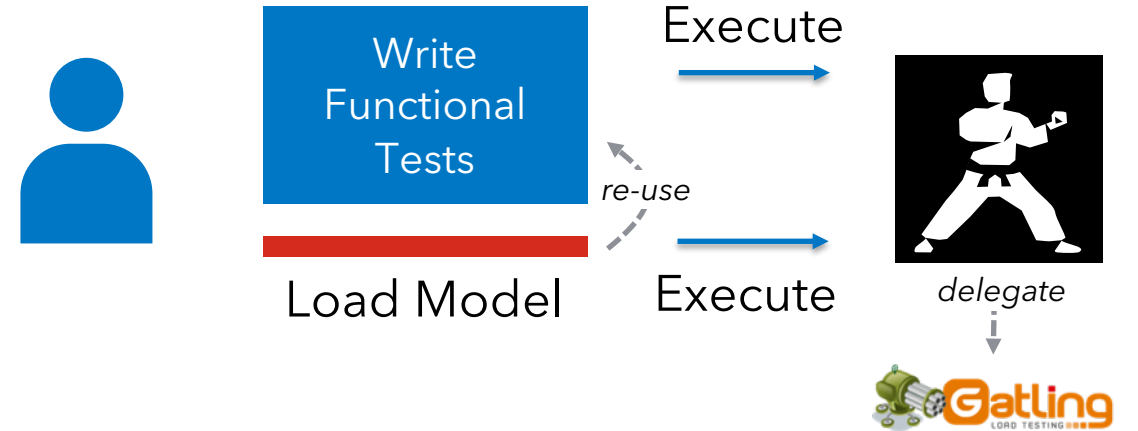
08:10:05.418 DEBUG -
1 < 200
1 < Cache-Control: no-cache, no-store, max-age=0, must-revalidate
1 < Content-Type: application/json;charset=UTF-8
1 < Date: Tue, 05 Sep 2017 02:40:05 GMT
1 < Expires: 0
1 < Pragma: no-cache
  
```



# Performance Testing



The Old Way



With Karate

**Demo**

# Components

Inception Feb 2017

Released Feb 2018

Released Jul 2018

(RC Available)

API Testing

Test Doubles

Perf

UI Automation

JUnit  
(Runner)

Karate Test Script

Karate Mock Script

Gatling  
Script

Reports

Gherkin Parser  
Karate DSL interpreter

Netty  
Adapter

Gatling  
Adapter

Unified  
Driver Interface

HTTP Client  
Interface

JS  
Engine

JSON

XML

Async

HTTP  
Server  
(Netty)

Gatling

Chrome  
DevTools  
Protocol

W3C  
Web  
Driver  
Adapters

Mock  
Servlet

Apache

Jersey

Data &  
Assertions

Web  
Sockets

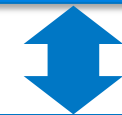
Java 8 JRE



HTTP/S



HTTP/S



Chrome



+ other  
browsers

# Karate Features

- REST, GraphQL as well as XML / SOAP support
- User defined extensibility via functions (Java or JavaScript)
- Configuration switching (e.g. dev | e2e | pre-prod)
- Simple 'plug and play' HTTP Header & Auth management
- Comprehensive HTTP support: SSL, Proxy, File-Upload
- JUnit XML reports compatible with all CI tools
- Speed - Parallel Execution of Tests
- Test Doubles - state-ful "smart" mocks for any HTTP(S) API
- Performance Testing - re-use Karate Scripts as Gatling Scripts
- WebSocket / Asynchronous message listener support
- Cross-Browser UI Automation

API Tests

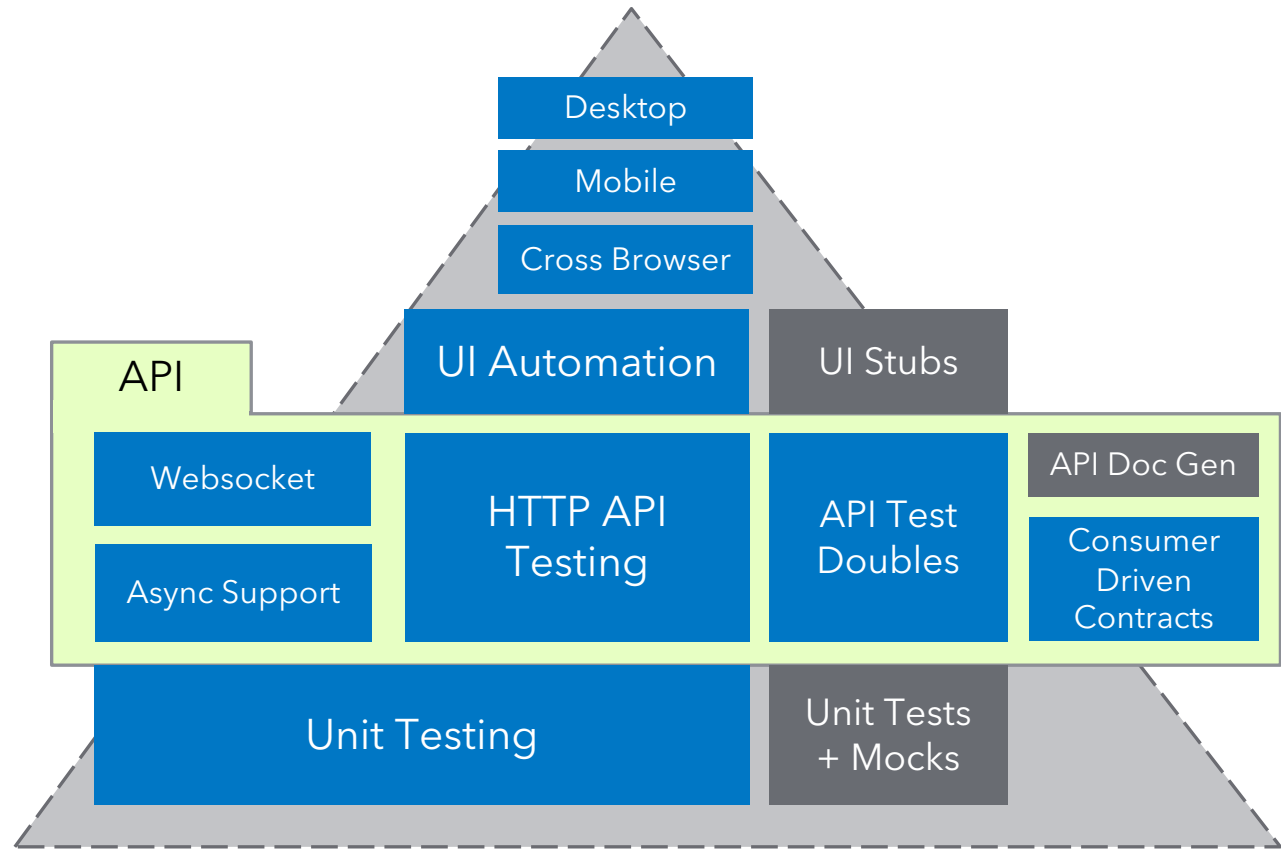
API Mocks

API Perf

UI  
Automation

- Run History & Insights
- Record - Replay
- Test Debugging
- Human Friendly Tests
- Video Recording
- Data Driven Testing
- Assertions
- Parallel Execution
- CI Integration / Compat
- HTML Reports
- Grouping / Tags
- Test Runner / CLI
- Test Data import / CSV
- Traceability / Logging
- Config from Files / Env
- Environment Switching
- Capabilities

- AI / ML Visual Validation
- Visual Navigation
- Accessibility
- UI and UX
- UX Performance
- Chaos Testing
- Distributed Testing
- Perf / Load Testing
- Scale
- AI / ML
- Property Based Testing
- Fuzz Testing
- Fake Data Generation
- Security Testing
- Test Generation



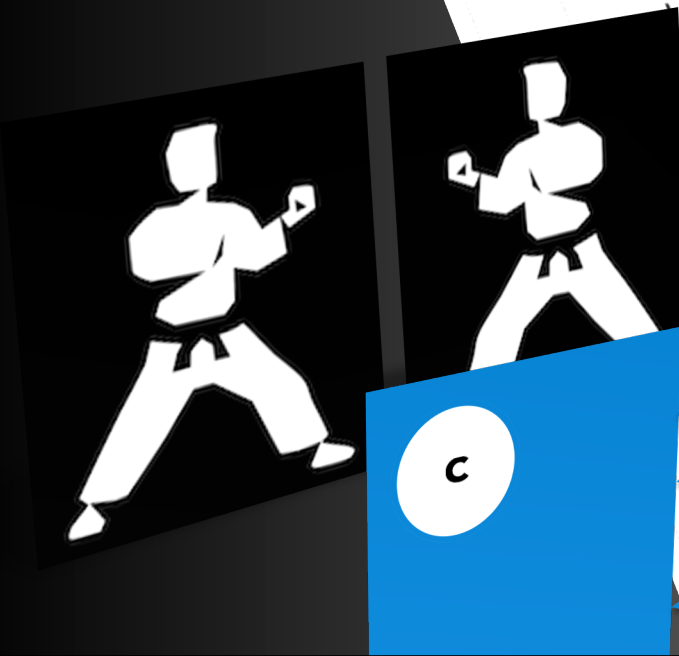


# Thank You !

```
Background.  
* def nextId = call read('increment.js')  
* def cats = {}
```

```
Scenario: pathMatches('/cats') && methodIs('post') && typeContains('xml')  
* def cat = request  
* def id = nextId()  
* set cat /cat/id = id  
* set catJson
```

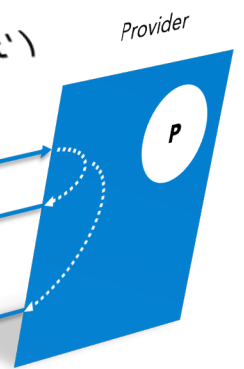
path	value
id	id
name	cat.cat.name
val cats[id + ""]	= catJson
def response	= cat



```
Scenario: pathMatches('/cats') && methodIs('post')  
* def cat = request  
* def id = nextId()  
* set cat.id = id  
* eval cats[id + ""] = cat  
* def response = cat
```

```
Scenario: pathMatches('/cats')  
* def response = $cats.*
```

```
Scenario: pathMatches('/cats/{id}') && acceptContains('xml')  
* def cat = call read('cats/{pathParams.id}')  
* def response = call read('cats/{id}</id><name>#{cat.name}</name></cat>')  
Scenario: pathMatches('/cats/{id}')  
* def response = call read('cats/{id}</id><name>#{cat.name}</name></cat>')  
Scenario: pathMatches('/cats/{id}/kittens')  
* def response = cats[pathParams.id].kittens
```



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

@ptrthomas | @KarateDSL