

apidays
LIVE
AUSTRALIA
2020

API Unit Testing in DevOps Pipelines



Uli Hitzel
Developer Advocate
Axway



Developer Advocate



/in/uhitzel



/u1i/slides



<https://apigeek.net>

11:15 AM

Projects

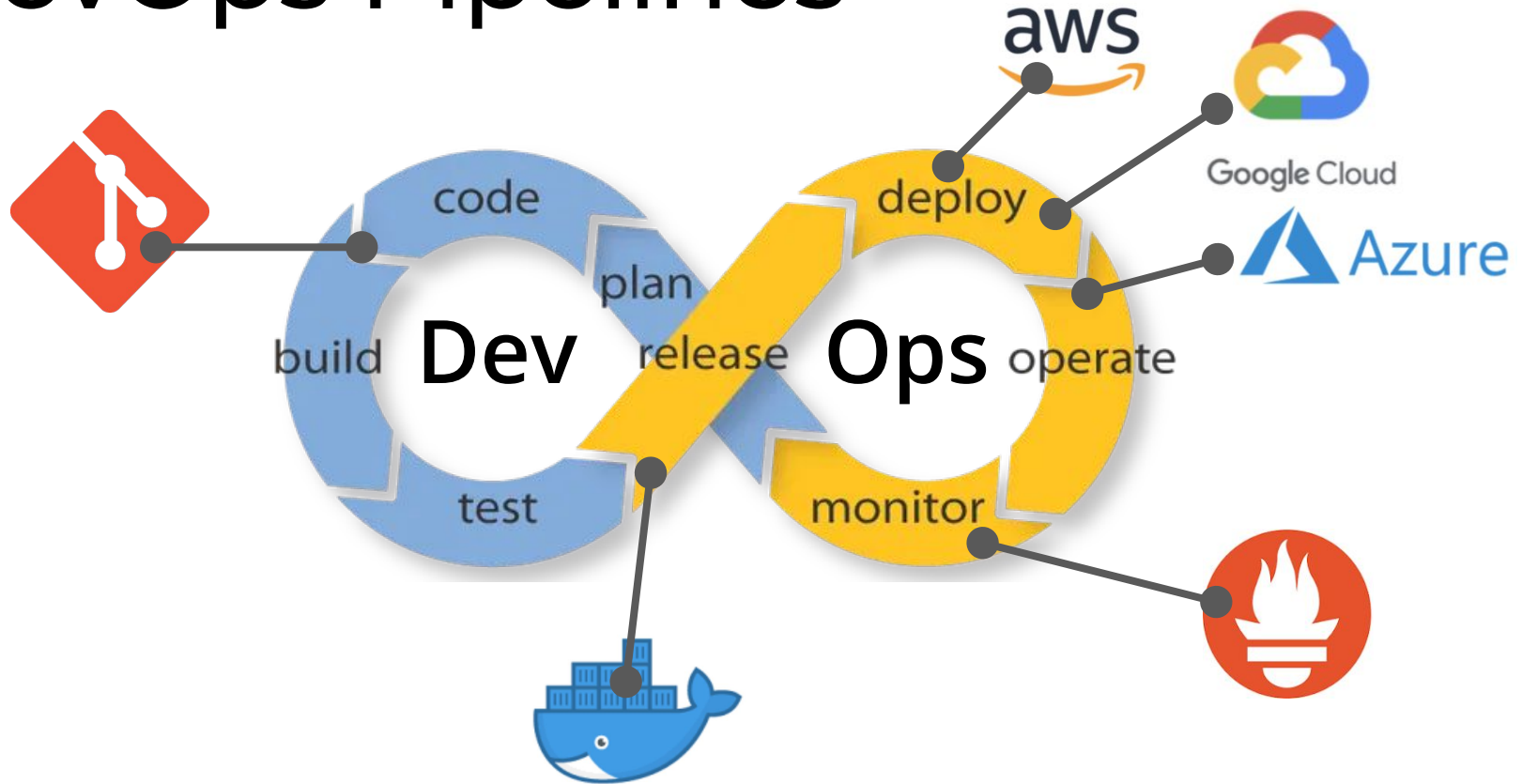
- [Yoisho Open Banking Project](#)
- [Bambleweeny](#): A lightweight key-value store and message broker based on HTTP/REST
- [keiju](#) - Minimalist API Gateway in a ~25 MB Container.
- [samsa](#) converts between OpenAPI v2 and v3 formats (YAML/JSON)
- [majime](#) - Dead Simple API Unit Testing

Speaking Engagements - Slides

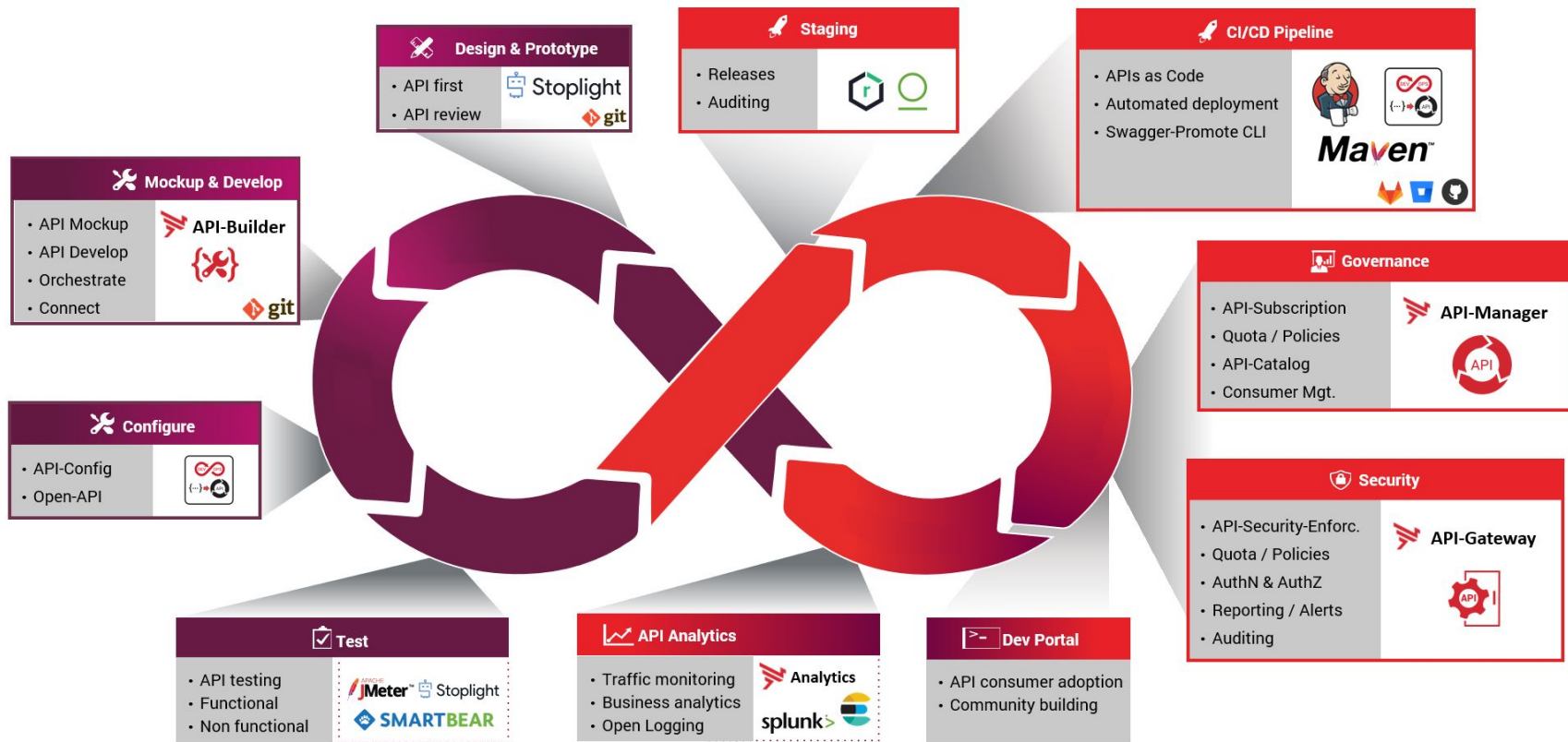
- [Redis Meetup Singapore](#) (April 2018)
- [PyCon Bangkok](#) (June 2018)
- [Stripe - API for Humans](#) (March 2019)
- [RedisConf San Francisco](#) (April 2019)



DevOps Pipelines



API Lifecycle



- Run test cases on API endpoints
- Test cases are defined in YAML files
- majime can generate test cases from Swagger specs

The logo for 'majime' is written in a black, cursive, handwritten-style font.

/ma-ji-me/

noun

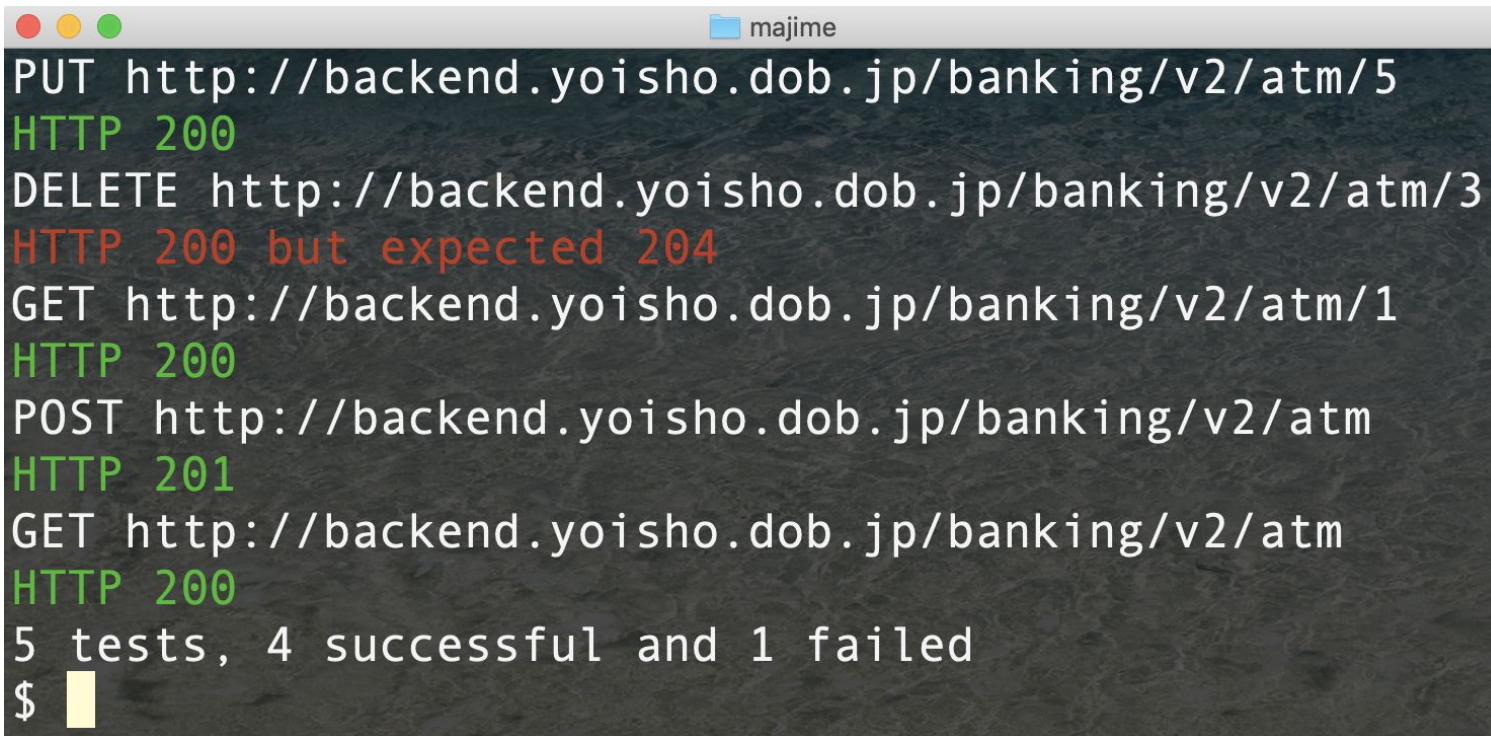
1. an earnest, reliable person who can get things done
2. dead simple API Unit Testing

python 3.7

license MIT

issues 0 open

Run Test Cases against Endpoints

A terminal window with a title bar containing three colored circles (red, yellow, green) and a folder icon labeled 'majime'. The terminal displays the results of five HTTP test cases. The first three are successful (200), the fourth fails (200 instead of 204), and the fifth is successful (200). A summary line indicates 4 successful and 1 failed test. The prompt '\$' is followed by a yellow cursor block.

```
PUT http://backend.yoisho.dob.jp/banking/v2/atm/5
HTTP 200
DELETE http://backend.yoisho.dob.jp/banking/v2/atm/3
HTTP 200 but expected 204
GET http://backend.yoisho.dob.jp/banking/v2/atm/1
HTTP 200
POST http://backend.yoisho.dob.jp/banking/v2/atm
HTTP 201
GET http://backend.yoisho.dob.jp/banking/v2/atm
HTTP 200
5 tests, 4 successful and 1 failed
$
```

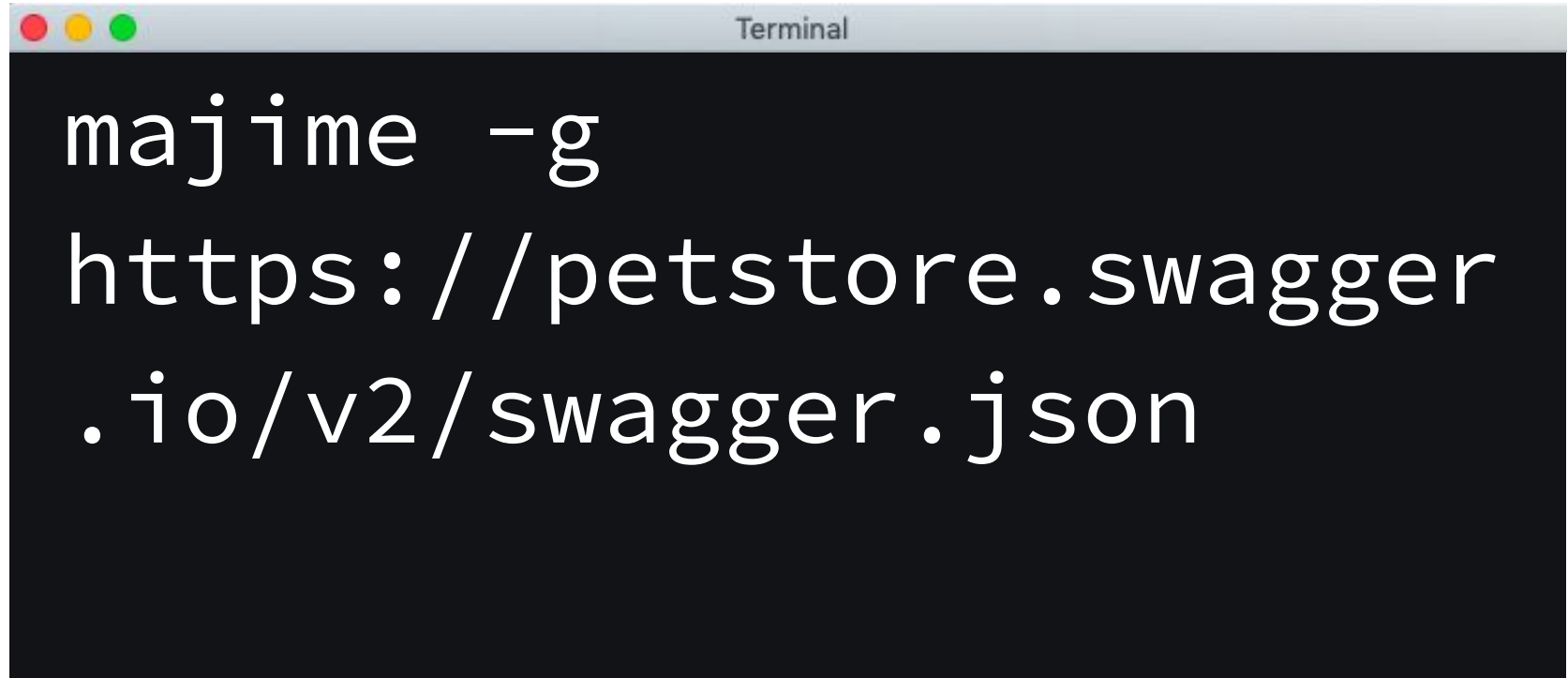
Test Cases Defined In YAML

Base: 'http://backend.yoisho.dob.jp/banking/v2'

Tests:

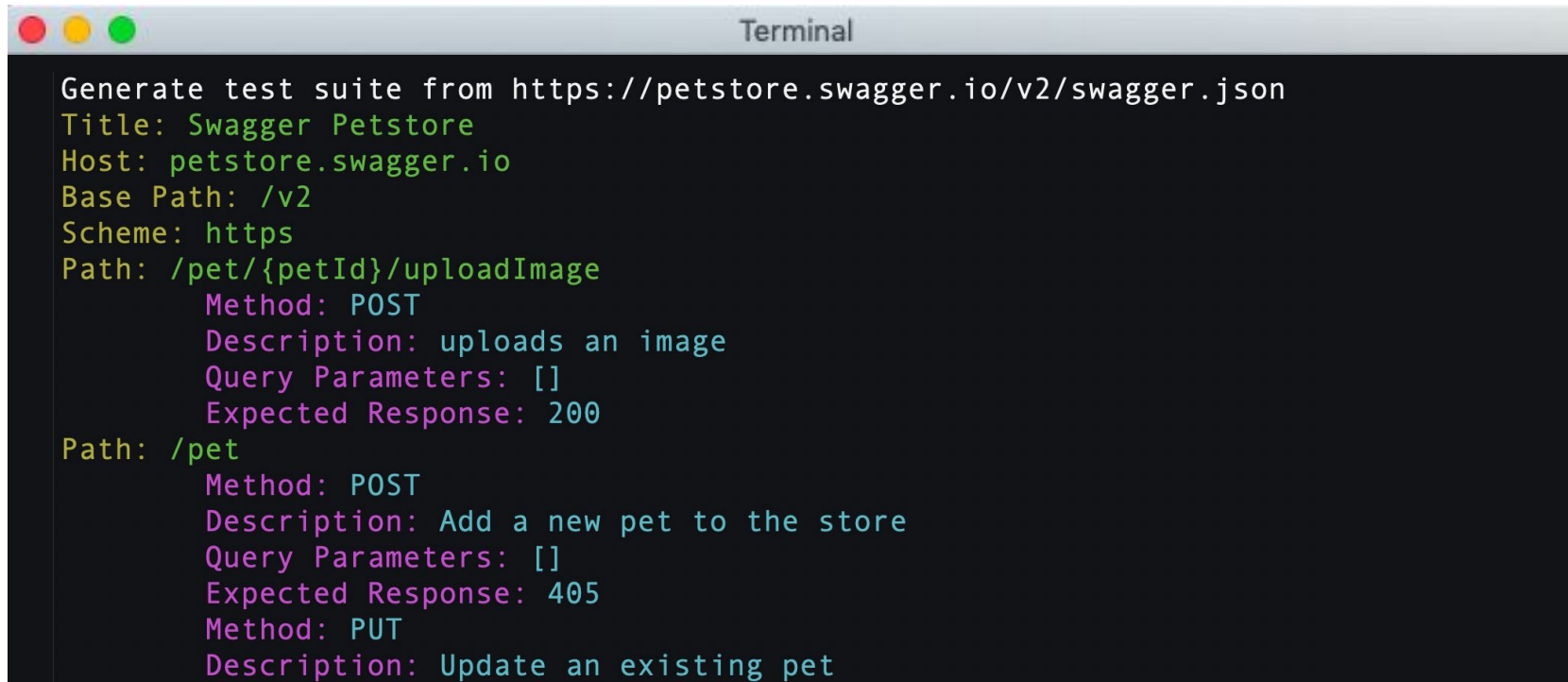
- path: '/atm/{id}'
method: GET
headers: ''
expect-response: '200'
expect-body: json
- path: '/atm/{id}'
method: PUT
headers: ''
content-type: application/json
body: {}
expect-response: '200'

Generate Test Cases from Swagger

A terminal window with a dark background and a light gray title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left and the word "Terminal" in the center. The terminal displays a command in white text, split across three lines: "majime -g", "https://petstore.swagger", and ".io/v2/swagger.json".

```
Terminal  
majime -g  
https://petstore.swagger  
.io/v2/swagger.json
```

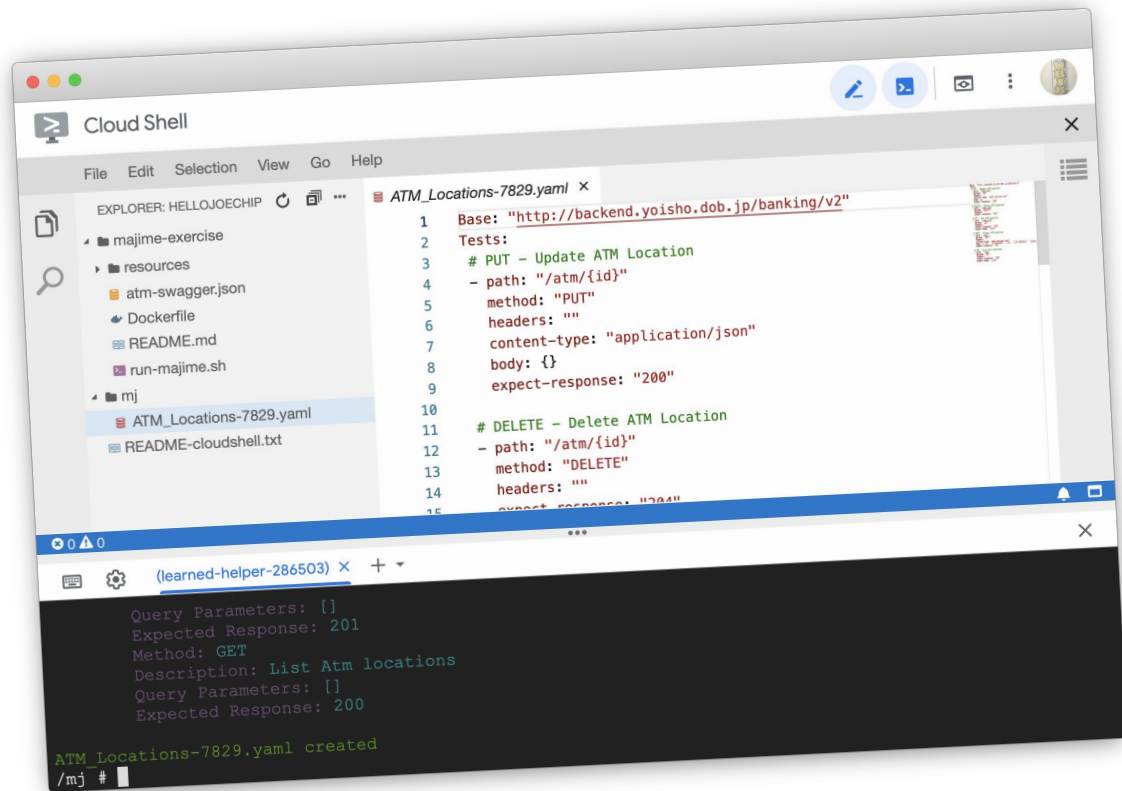
Generate Test Cases from Swagger

A terminal window with a title bar labeled "Terminal" and three colored window control buttons (red, yellow, green) on the left. The terminal displays the output of a command to generate a test suite from a Swagger API definition. The output lists the title, host, base path, and scheme of the API, followed by details for two endpoints: a POST endpoint for uploading an image and a POST endpoint for adding a new pet to the store. The output is color-coded: green for titles and hosts, yellow for base paths and schemes, and purple for methods and descriptions.

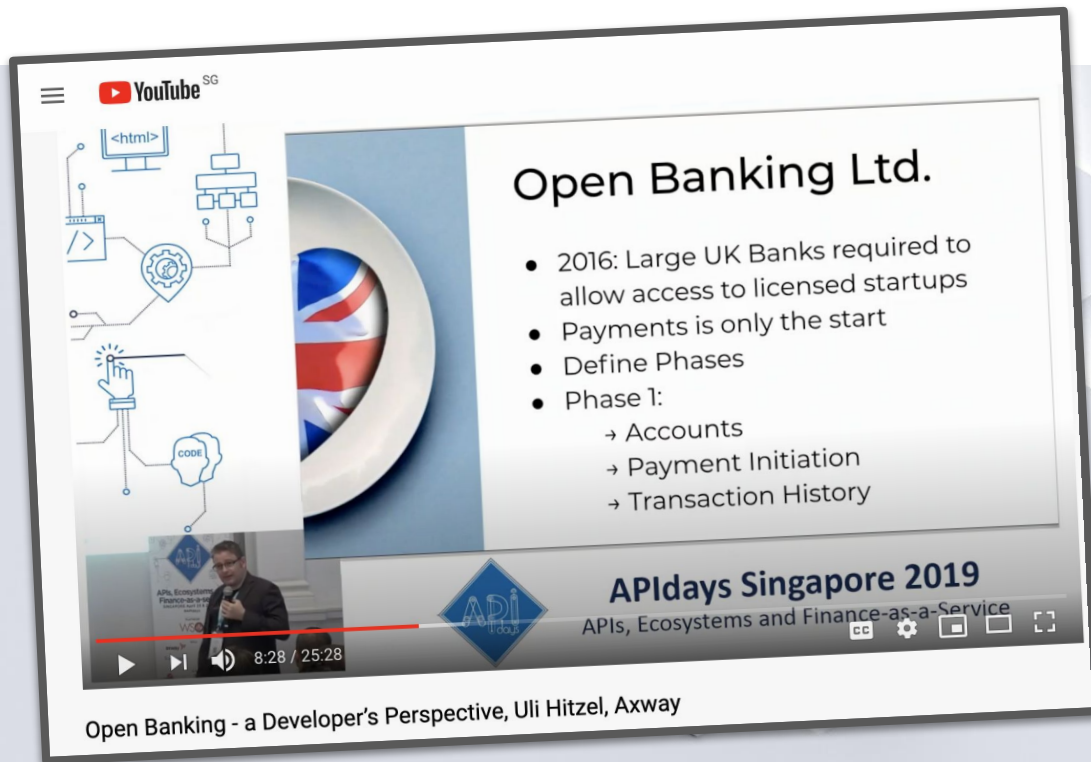
```
Generate test suite from https://petstore.swagger.io/v2/swagger.json
Title: Swagger Petstore
Host: petstore.swagger.io
Base Path: /v2
Scheme: https
Path: /pet/{petId}/uploadImage
    Method: POST
    Description: uploads an image
    Query Parameters: []
    Expected Response: 200
Path: /pet
    Method: POST
    Description: Add a new pet to the store
    Query Parameters: []
    Expected Response: 405
    Method: PUT
    Description: Update an existing pet
```

<https://github.com/u1i/majime-exercise>

Let's Try It Out!



API Days Singapore, 2019



The image shows a YouTube video player interface. The video is titled "Open Banking - a Developer's Perspective, Uli Hitzel, Axway". The video player shows a presentation slide with the title "Open Banking Ltd." and a list of bullet points. The slide also features a graphic of a Union Jack flag inside a circular frame. The video player includes a progress bar at the bottom, showing the video is at 8:28 out of 25:28. The YouTube logo and "SG" are visible in the top left corner of the player.

YouTube SG

Open Banking Ltd.

- 2016: Large UK Banks required to allow access to licensed startups
- Payments is only the start
- Define Phases
- Phase 1:
 - Accounts
 - Payment Initiation
 - Transaction History

API Days Singapore 2019
APIs, Ecosystems and Finance-as-a-Service

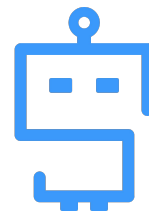
Open Banking - a Developer's Perspective, Uli Hitzel, Axway


```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server
4     out more about Swagger at [http://swagger.io]
5     or on [irc.freenode.net, #swagger]
6     this sample, you can use the api to
7     authorization filters."
8   version: "1.0.0"
9   title: "Swagger Petstore"
10  termsOfService: "http://swagger.io/terms/"
11  contact:
12    email: "apiteam@swagger.io"
13  license:
14    name: "Apache 2.0"
15    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
16  host: "petstore.swagger.io"
17  basePath: "/v2"
18  tags:
19    - name: "pet"
20      description: "Everything about your
21        externalDocs:
22          description: "Find out more"
23          url: "http://swagger.io"
24    - name: "store"
25      description: "Access to Petstore orders"
26    - name: "user"
27      description: "Operations about user"
28      externalDocs:
29        description: "Find out more about user"
30        url: "http://swagger.io"
31  securityDefinitions:
32    api_key:
33      type: "apiKey"
34      name: "api_key"
35      in: "header"
36  security:
37    - api_key: []
38  externalDocs:
39    description: "Find out more about Swagger"
40    url: "http://swagger.io"
41  swagger: "2.0"
```

Design First

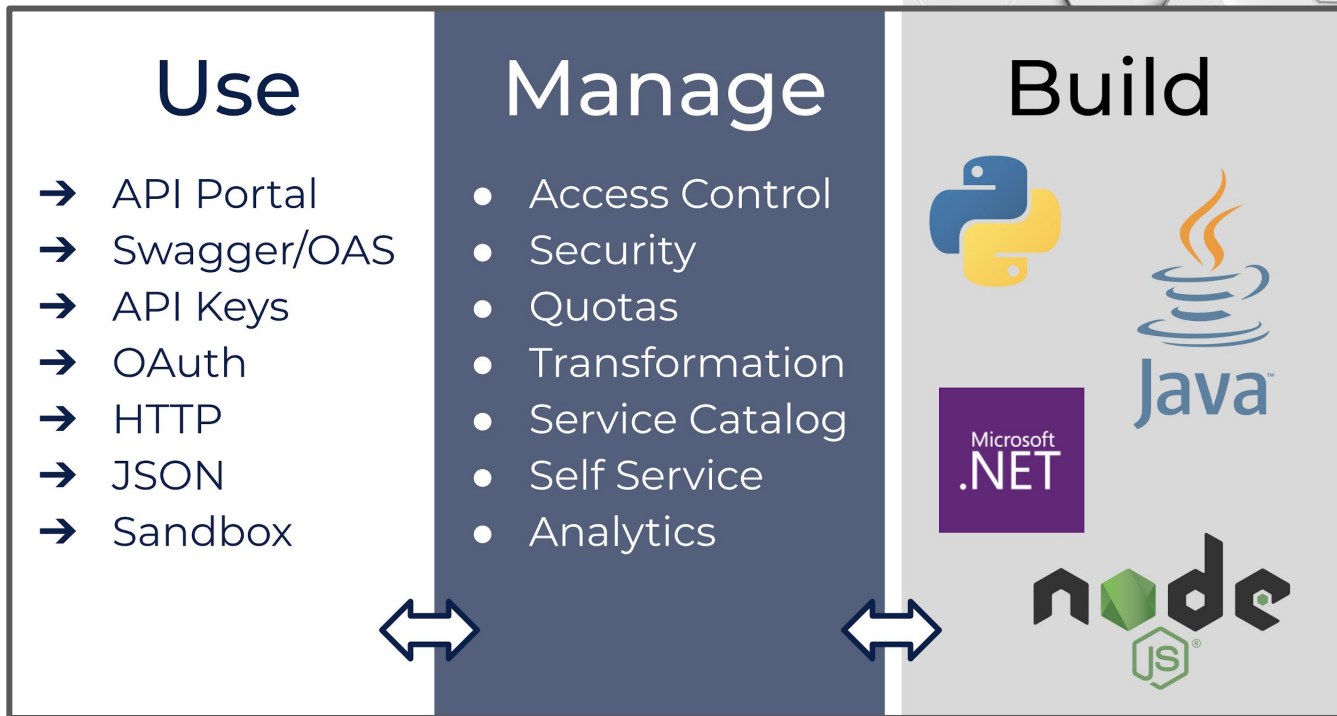


- What can I do?
- What do I get?
- Data Models
- Authentication
- License



Divide & Conquer

2





YOISHO
偽の会社

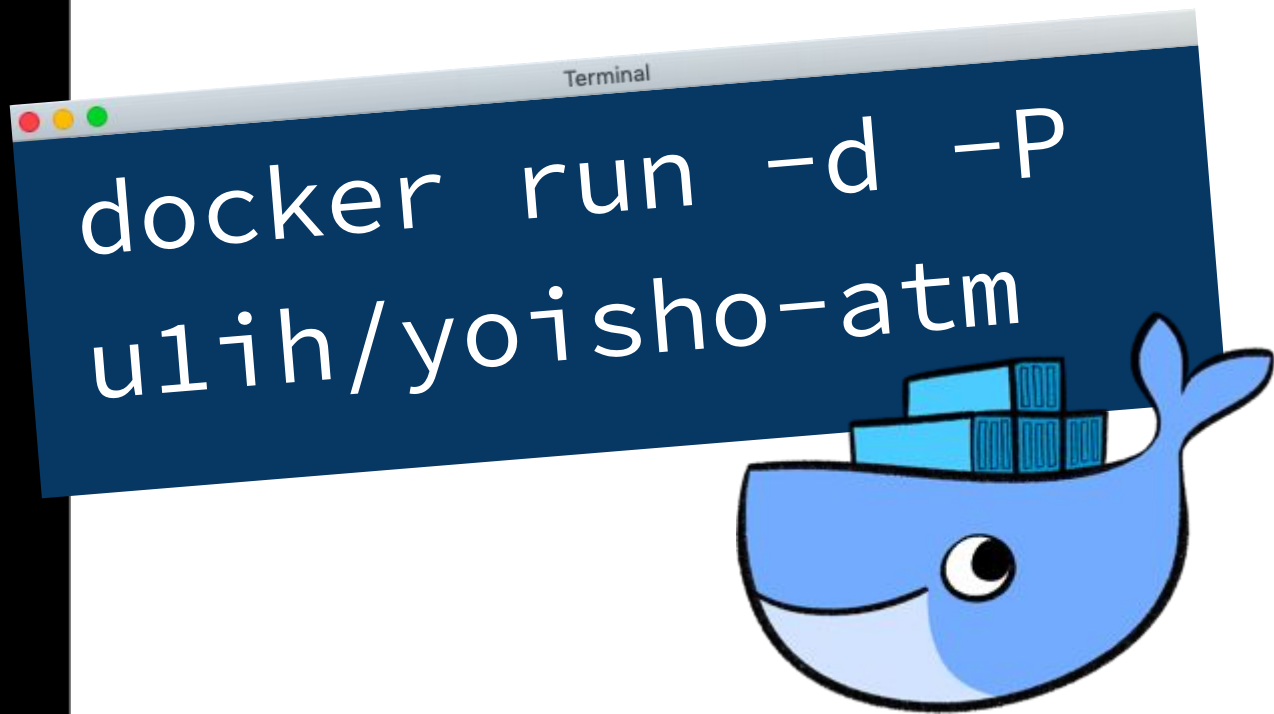
SOAP & REST
Webservices / APIs
for a fictional
japanese bank. Easy
to run docker
images that expose
endpoints you can
use for demos and
testing.



View On GitHub

Use Mock Tools

3



Open Banking - Live Endpoints



Webservices and REST APIs that expose bank related data services with dynamic content, use them for testing and demos. Available over [HTTP](#) and [HTTPS](#)

[Full Documentation](#)

ATM

Full CRUDL REST interface (Create, Read, Update, Delete, List), /v1 and /v2 endpoints with respective Swagger specs

[API Version 1 - Swagger](#)

[API Version 1 - Swagger Editor](#)

[API Version 2 - Swagger](#)

[API Version 2 - Swagger Editor](#)

[Sample Request](#)

Currency Exchange

REST/JSON, 1 parameter. Response is different each time.

[Swagger](#)

[Swagger Editor](#)

[Sample Request](#)

Assets (SOAP)

Bank Assets & Debt- SOAP/XML, 2 methods, dynamic output.

[WSDL](#)

Open Banking - Live Endpoints



Webservices and REST APIs that expose bank related data services with dynamic content, use them for testing and demos. Available over [HTTP](#) and [HTTPS](#)

[Full Documentation](#)

Fixed Deposit

Calculator - REST/JSON, 2 parameters, semantic error handling, complex output

[Swagger](#)

[Swagger Editor](#)

[Sample Request](#)

Investment

/v1 and /v2 endpoints with respective Swagger specs. Static responses.

[API Version 1 - Swagger](#)

[API Version 1 - Swagger Editor](#)

[API Version 2 - Swagger](#)

[Swagger Editor](#)

[API Version 2 - Sample Request](#)

Stock Quote

Returns the stock price. Data updates once every minute, it's also sending additional Cache-Control headers.

[Swagger](#)

[Swagger Editor](#)

[Sample Request](#)

ATM Locations^{2.0}

[Base URL: backend.yoisho.dob.jp/banking/v2]

<https://backend.yoisho.dob.jp/banking/v2/swagger>

List of ATM locations for Yoisho Banking Corporation

PUT

/atm/{id} Update ATM Location

DELETE

/atm/{id} Delete ATM Location

GET

/atm/{id} Get ATM Location

POST

/atm Create ATM Location

GET

/atm List Atm locations

Over To You!

- **Try it out:** Use Majime for Testing at Work
- **Feedback:** Please open issues for stuff that doesn't work
- **Join In:** Looking for collaborators!



/u1i

