

@ailuj876

A futuristic blue circular device with a glowing beam of light emanating from its center, set against a dark blue background with a network of glowing nodes and lines.

# Next Level API Automation



# Julia Pottinger

Head of Training and Development  QualityWorks



ailuj876



youtube.com/juliapottinger



juliapottinger.com



linkedin.com/in/julia-pottinger/



TestAutomationU



# What is an API?

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.

# API AUTOMATION *Benefits*

Greater test stability

Language independence

Faster test results

Reduced costs

Improved Test Coverage

Earlier Bug Fix



# API TESTING Tools



PACTUM JS



REST ASSURED



SoapUI



Katalon





API ▶

🔮 Requests ▶

🧶 Assertions ▶

🔧 Utils ▶

📎 Matching ▶

🔨 Settings ▶

🧰 Handlers ▶

🤖 Mock ▶

👛 Stash ▶

Specifies basic authentication.

## Syntax

1

```
withAuth(username, password)
```

js

## Usage



### Correct Usage

1

```
await spec()
```

2

```
  .get('/api/users')
```

3

```
  .withAuth('my-username', 'super-secret-password')
```

4

```
  .expectStatus(200);
```

js



# HTTP API Requests

|        |  |
|--------|--|
| GET    | Retrieves resources                              |
| POST   | Creates resources                                |
| PUT    | Changes and/or replaces resources or collections |
| DELETE | Deletes resources                                |



# API Responses

|           |                        |
|-----------|------------------------|
| 100 - 199 | Information Response   |
| 200-299   | Successful Responses   |
| 300-399   | Redirection Messages   |
| 400-499   | Client Error Responses |
| 500-599   | Server Error Responses |



# API Responses

|     |   |
|-----|---|
| 200 | OK - The request has succeeded.                                       |
| 201 | Created - The request has been fulfilled.                             |
| 400 | Bad Request - The request could not be understood by the server       |
| 401 | Unauthorized - requires user authentication or, authorization refused |
| 404 | Not Found - The requested resource could not be found.                |
| 500 | Internal Server Error.  |
| 502 | Bad Gateway   |

# API Testing Checklist





@ailuj876



1. Prioritize APIs

2. Ensure API's are  
properly documented



# Define the types of tests that you want to run

- Positive scenarios return a valid response
- Invalid requests return the correct error message
- Schema match
- Workflow and data persistence
- Response Payload - valid JSON body, correct field names, types, and values
- Application state before and after API call
- Security and authorization
- Response time
- Response Headers



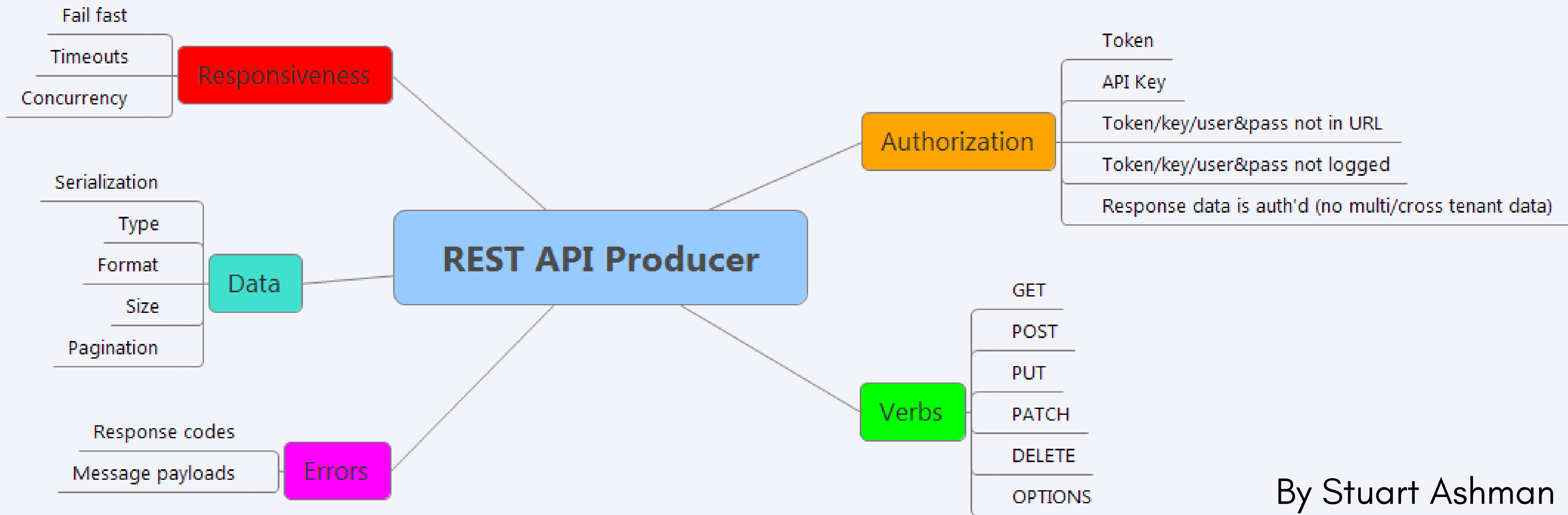
# 4. Evaluate and select an API testing tool

[juliapottinger.com/api-testing-checklist](https://juliapottinger.com/api-testing-checklist)



# VADER API Heuristics

@ailuj876



By Stuart Ashman

# API Automation

@ailuj876



# Repository

<https://github.com/jpott77/api-automation-supertest>



@ailuj876

# Repository

<https://github.com/jpott77/api-automation-pactum>



@ailuj876



# Postman Collection

<https://www.getpostman.com/collections/1d13a7a0eb5056cd6e04>



@ailuj876

# Postman Collection

## Automating API's 👤

GET GetBookingByID

POST Create Booking

PUT Update Booking

PUT Unauthorized Update

GET <https://jsonplaceholder.typicode.com/posts>

POST SpaceX GraphQL

@ailuj876



# NPM Packages Used

**supertest** - make the API calls

**mocha** - test runner

**jsonschema** - validate the API response against a schema

**chai** - assertions

# Supertest Keywords

- **request** keyword is used to make a request
  - you pass in the URL for the API
- You then chain the request type - PUT, GET, POST
- **send** contains the detail of the request body
- **set** is used for headers and authorization
- **expect** validates the response body



**1**

**Positive Scenarios**

**Return Valid**

**Response**





POST



https://restful-booker.herokuapp.com/booking

Params

Authorization

Headers (10)

Body ●

Pre-request Script

Tests

Settings

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON ▼

```
1 {
2   "firstname": "Julia",
3   "lastname": "Brown",
4   "totalprice": 567,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2020-01-01",
8     "checkout": "2021-01-01"
9   },
10  "additionalneeds": "Breakfast,Lunch"
11 }
```

Body

Cookies

Headers (8)

Test Results



Status: 200 OK Time: 891 ms

Pretty

Raw

Preview

Visualize

JSON ▼



```
1 {
2   "bookingid": 10145,
3   "booking": {
4     "firstname": "Julia",
5     "lastname": "Brown",
6     "totalprice": 567,
7     "depositpaid": true,
8     "bookingdates": {
9       "checkin": "2020-01-01",
10      "checkout": "2021-01-01"
11     },
12    "additionalneeds": "Breakfast,Lunch"
13  }
```

@ailuj876



```
1 const request = require('supertest');
2 const expect = require('chai').expect
3
4 describe('Create Booking -', ()=>{
5   it('positive request', async () =>{
6     let url = 'https://restful-booker.herokuapp.com/booking'
7
8     let res = await request(url)
9       .post('/')
10      .send({
11        "firstname" : "Julia",
12        "lastname" : "Brown",
13        "totalprice" : 567,
14        "depositpaid" : true,
15        "bookingdates" : {
16          "checkin" : "2020-01-01",
17          "checkout" : "2021-01-01"
18        },
19        "additionalneeds" : "Breakfast,Lunch"
20      })
21      .set('Accept', 'application/json')
22      .expect(200)
23
```

supertest

```

1  const { spec } = require('pactum');
2  const bookingSchema = require('./schema/bookingSchema')
3
4  describe('Positive Booking Scenario -', ()=>{
5      it('verify creating a sucessful booking', async () =>{
6          let url = 'https://restful-booker.herokuapp.com/booking'
7
8          await spec()
9              .post(url)
10             .withHeaders({
11                 'Content-Type': 'application/json',
12                 'Accept': 'application/json'
13             })
14             .withJson({
15                 "firstname" : "Julia",
16                 "lastname" : "Brown",
17                 "totalprice" : 567,
18                 "depositpaid" : true,
19                 "bookingdates" : {
20                     "checkin" : "2020-01-01",
21                     "checkout" : "2021-01-01"
22                 },
23                 "additionalneeds" : "Breakfast,Lunch"
24             })
25             .expectStatus(200)
26         })
27     })

```

# Pactum

@ailuj876



2

**Invalid requests  
returns correct error  
message**

```
it('Verify that the post is not updated with an invalid body', async ()=>{
  const url = 'https://restful-booker.herokuapp.com/booking/10'

  request(url)
    .put('/')
    .send({
      "lastname" : "Potting",    First name is missing
      "totalprice" : 111,
      "depositpaid" : true,
      "bookingdates" : {
        "checkin" : "2018-01-01",
        "checkout" : "2019-01-01"
      },
      "additionalneeds" : "Breakfast"
    })
    .set('Accept', 'application/json')
    .set('Authorization', 'Basic YWRtaW46cGFzc3dvcmQxMjM=')
    .expect(400)
    .expect('Bad Request')
  })
})
```



**Do you want to  
automate all the  
negative API  
scenarios?**



# Security and Authorization

**responds securely to possible security attacks**

**internal data representations are not shared in response body**

**Fails securely**

**Rate Limiting and throttling**

**refuse calls to endpoints if user not permitted**

**responds as expected to correct authorization-  
Bearer token, cookies etc**



```
it('Verify that the post is not updated with an invalid token', async () => {  
  
  const url = `https://restful-booker.herokuapp.com/booking/${bookingId}`  
  
  request(url)  
    .put('/')  
    .send({  
      "firstname" : "Julia",  
      "lastname" : "Browns",  
      "totalprice" : 567,  
      "depositpaid" : true,  
      "bookingdates" : {  
        "checkin" : "2020-01-01",  
        "checkout" : "2021-01-01"  
      },  
      "additionalneeds" : "Breakfast,Lunch"  
    })  
    .set('Accept', 'application/json')  
    .set('Authorization', 'Basic YWRtaW46cGFzc3dvcmQxMjM') Invalid token  
    .expect(403)  
    .expect('Forbidden')  
  })  
})
```

4

**API Response matches  
the expected schema**



# What is a Schema

A schema is an outline, diagram, or model. In computing, schemas are often used to describe the structure of different types of data. Two common examples include database and XML schemas.

# What is JSON Schema

JSON Schema is a vocabulary that allows you to annotate and validate JSON documents.

Validates data which is useful for:

- Automated testing.
- Ensuring quality of client submitted data



PUT Update Booking

POST Create Booking

GET https://jsonplaceholder.typi...

GET GetBookingIDs

+

...

No Environment

👁

⚙

▶ Create Booking

Examples 0

BUILD

✎

💬

POST

https://restful-booker.herokuapp.com/booking

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

Query Params

|  | KEY | VALUE | DESCRIPTION | ⋮ | Bulk Edit |
|--|-----|-------|-------------|---|-----------|
|  | Key | Value | Description |   |           |

Body

Cookies

Headers (8)

Test Results

🌐

Status: 200 OK

Time: 103 ms

Size: 448 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

≡

1

{

2

"bookingid": 57,

3

"booking": {

4

"firstname": "Julia",

5

"lastname": "Brown",

6

"totalprice": 567,

7

"depositpaid": true,

8

"bookingdates": {

9

"checkin": "2020-01-01",

10

"checkout": "2021-01-01"

11

},

12

"additionalneeds": "Breakfast,Lunch"

13

}

14

}

@ailuj876



## JSON

Settings

✓ Valid JSON



```
1 {
2   "bookingid": 13,
3   "booking": {
4     "firstname": "Julia",
5     "lastname": "Brown",
6     "totalprice": 567,
7     "depositpaid": true,
8     "bookingdates": {
9       "checkin": "2020-01-01",
10      "checkout": "2021-01-01"
11    },
12    "additionalneeds": "Breakfast,Lunch"
13  }
14 }
```

## Schema

Untitled schema created 2022-09-20 03:16:56

```
1 {
2   "$schema": "https://json-schema.org/draft/2019-09/schema",
3   "$id": "http://example.com/example.json",
4   "type": "object",
5   "default": {},
6   "title": "Root Schema",
7   "required": [
8     "bookingid",
9     "booking"
10  ],
11  "properties": {
12    "bookingid": {
13      "type": "integer",
14      "default": 0,
15      "title": "The bookingid Schema",
16      "examples": [
17        13
18      ]
19    },
20    "booking": {
21      "type": "object",
22      "default": {},
23      "title": "The booking Schema",
24      "required": [
25        "firstname",
26        "lastname",
27        "totalprice",
28        "depositpaid",
29        "bookingdates",
30        "additionalneeds"
31      ]
32    }
33  }
34 }
```



@ailuj876



```

32 1,
33   "properties": {
34     "firstname": {
35       "type": "string",
36       "default": "",
37       "title": "The firstname Schema",
38       "examples": [
39         "Julia"
40       ],
41     },
42     "lastname": {
43       "type": "string",
44       "default": "",
45       "title": "The lastname Schema",
46       "examples": [
47         "Brown"
48       ],
49     },
50     "totalprice": {
51       "type": "integer",
52       "default": 0,
53       "title": "The totalprice Schema",
54       "examples": [
55         567

```

Schema has the properties

@ailuj876

```
module.exports = {
  "$schema": "http://json-schema.org/draft-07/schema",
  "$id": "http://example.com/example.json",
  "type": "object",
  "title": "The root schema",
  "description": "The root schema comprises the entire JSON document.",
  "default": {},
  "examples": [
    {
      "bookingid": 13,
      "booking": {
        "firstname": "Julia",
        "lastname": "Brown",
        "totalprice": 567,
        "depositpaid": true,
        "bookingdates": {
          "checkin": "2020-01-01",
          "checkout": "2021-01-01"
        },
        "additionalneeds": "Breakfast,Lunch"
      }
    }
  ],
  "required": [
    "bookingid",
    "booking"
  ],
}
```



```

1  const request = require('supertest');
2  var validate = require('jsonschema').validate
3  const expect = require('chai').expect
4
5  const bookingSchema = require('./bookingSchema')
6
7  describe('Positive Booking Scenario -', ()=>{
8    it('verify creating a sucessful booking', async () =>{
9      let url = 'https://restful-booker.herokuapp.com/booking'
10     let res = await request(url)
11     .post('/')
12     .send({
13       "firstname" : "Julia",
14       "lastname" : "Brown",
15       "totalprice" : 567,
16       "depositpaid" : true,
17       "bookingdates" : {
18         "checkin" : "2020-01-01",
19         "checkout" : "2021-01-01"
20       },
21       "additionalneeds" : "Breakfast,Lunch"
22     })
23     .set('Accept', 'application/json')
24     .expect(200)
25
26     expect(validate(res.body, bookingSchema).errors.length).to.be.equal(0)
27   })

```



**Handles workflows and  
data persists  
throughout as expected**



# Workflow



```
describe('Full Booking Path', ()=>{
```

```
  it('verify multistep API workflow ', async () =>{
```

```
    let url = 'https://restful-booker.herokuapp.com/booking'
```

```
    let res = await request(url)
```

```
    .post('/')
```

```
    .send({
```

```
      "firstname" : "Julia",
```

```
      "lastname" : "Brown",
```

```
      "totalprice" : 567,
```

```
      "depositpaid" : true,
```

```
      "bookingdates" : {
```

```
        "checkin" : "2020-01-01",
```

```
        "checkout" : "2021-01-01"
```

```
      },
```

```
      "additionalneeds" : "Breakfast,Lunch"
```

```
    })
```

```
    .set('Accept', 'application/json');
```

```
    bookingId = res.body.bookingid
```

```
    expect(res.statusCode).to.be.equal(200)
```

```
    expect(validate(res.body, bookingSchema).errors.length).to.be.equal(0)
```

1

@ailuj876



2

```
// Use Id to get booking details
request(`https://restful-booker.herokuapp.com/booking/${bookingId}`)
  .get('')
  .set('Accept', 'application/json')
  .expect(200)
  .then(response => {
    expect(response.body.firstname, 'Julia')
    expect(response.body.lastname, 'Brown')
    expect(response.body.additionalneeds, 'Breakfast,Lunch')
  })
```

```
// Use Id to update booking details
url = `https://restful-booker.herokuapp.com/booking/${bookingId}`

let putResponse = await request(url)
  .put('/')
  .send({
    "firstname" : "JuliaUpdate",
    "lastname" : "Browns",
    "totalprice" : 567,
    "depositpaid" : true,
    "bookingdates" : {
      "checkin" : "2020-01-01",
      "checkout" : "2021-01-01"
    },
    "additionalneeds" : "Breakfast,Dinner"
  })
  .set('Accept', 'application/json')
  .set('Authorization', 'Basic YWRtaW46cGFzc3dvcmQxMjM=')
  .expect(200)

expect(putResponse.body.firstname).to.equal('JuliaUpdate')
expect(putResponse.body.lastname).to.equal('Browns')
expect(putResponse.body.additionalneeds).to.equal('Breakfast,Dinner')
})
```

3



# The time it takes to deliver a response

@ailuj876



Scenarios also  
apply to GraphQL  
API's

Examples

Explorer

☐mission\_patch\_small

☐presskit

☐reddit\_campaign

☐reddit\_launch

☐reddit\_media

☐reddit\_recovery

☒video\_link

☐wikipedia

☐mission\_id

☒mission\_name

☐rocket

☐ships

☐static\_fire\_date\_unix

☐static\_fire\_date\_utc

☐telemetry

☐tentative\_max\_precision

☐upcoming

☐launchesPastResult

☐launchesUpcoming

☐launchpad

☐launchpads

☐mission

☐missions

☐missionsResult

☐payload

☐payloads

☐roadster

☐rocket

☐rockets

☐rocketsResult

☐ship

☐ships

☐shipsResult

☐users

☐users\_aggregate

☐users\_by\_pk

mutation

☐delete\_users

☐insert\_users

☐update\_users

GraphiQL

Prettify

History

Explorer

1 {

2   launchesPast(limit: 10) {

3     mission\_name

4     launch\_date\_local

5     launch\_site {

6       site\_name\_long

7     }

8     links {

9       article\_link

10      video\_link

11    }

12 } }

13 }

14 |

https://api.spacex.land/graphql/

QUERY VARIABLES

< Docs

{

"data": {

"launchesPast": [

{

"mission\_name": "Starlink-15 (v1.0)",

"launch\_date\_local": "2020-10-24T11:31:00-04:00",

"launch\_site": {

"site\_name\_long": "Cape Canaveral Air Force Station Space Launch Complex 40"

},

"links": {

"article\_link": null,

"video\_link": "https://youtu.be/J442-ti-Dhg"

}

},

{

"mission\_name": "Sentinel-6 Michael Freilich",

"launch\_date\_local": "2020-11-21T09:17:00-08:00",

"launch\_site": {

"site\_name\_long": "Vandenberg Air Force Base Space Launch Complex 4E"

},

"links": {

"article\_link": "https://spaceflightnow.com/2020/11/21/international-satellite-launches-to-extend-measurements-of-sea-level-rise/",

"video\_link": "https://youtu.be/aVFPzTDCihQ"

}

},

{

"mission\_name": "Crew-1",

"launch\_date\_local": "2020-11-15T19:27:00-05:00",

"launch\_site": {

"site\_name\_long": "Kennedy Space Center Historic Launch Complex 39A"

},

"links": {

"article\_link": "https://spaceflightnow.com/2020/11/16/astronauts-ride-spacex-crew-capsule-in-landmark-launch-for-commercial-spaceflight/",

"video\_link": "https://youtu.be/bnChQbxLkKI"

}

},

{

"mission\_name": "GPS III SV04 (Sacagawea)",

"launch\_date\_local": "2020-11-05T18:24:00-05:00",

"launch\_site": {

"site\_name\_long": "Cape Canaveral Air Force Station Space Launch Complex 40"

}

}

]

}

}

@ailuj876



POST https://api.spacex.land/graphql/



No Environment



## Untitled Request

BUILD



POST

https://api.spacex.land/graphql/

Send

Save

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Code

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

No schema



### QUERY

```
1 {
2   launchesPast(limit: 10) {
3     mission_name
4     launch_date_local
5     launch_site {
6       site_name_long
7     }
8     links {
9       article_link
10      video_link
11    }
12  }
13 }
```

### GRAPHQL VARIABLES

1

Body

Cookies

Headers (9)

Test Results



Status: 200 OK

Time: 3.49 s

Size: 3.44 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "data": {
3     "launchesPast": [
4       {
5         "mission_name": "Starlink-15 (v1.0)",
6         "launch_date_local": "2020-10-24T11:31:00-04:00",
7         "launch_site": {
8           "site_name_long": "Cape Canaveral Air Force Station Space Launch Complex 40"
9         },
10        "links": {
11          "article_link": null,
12          "video_link": "https://youtu.be/J442-ti-Dhg"
13        }
14      }
15    ]
16  }
17 }
```

@ailuj876

JSON

Settings

Your JSON is valid

```
1 {
2   "data": {
3     "launchesPast": [{
4       "mission_name": "Starlink-15 (v1.0)",
5       "launch_date_local": "2020-10-24T11:31:00-04:00",
6       "launch_site": {
7         "site_name_long": "Cape Canaveral Air Force Station Space Launch Complex 40"
8       },
9       "links": {
10        "article_link": null,
11        "video_link": "https://youtu.be/J442-ti-Dhg"
12      }
13    }]
14  }
15 }
```

JSON Schema

Code Graph Edit

JSON

COPY TO CLIPBOARD

EMAIL SCHEMA

```
1 {
2   "$schema": "http://json-schema.org/draft-07/schema",
3   "$id": "http://example.com/example.json",
4   "type": "object",
5   "title": "The root schema",
6   "description": "The root schema comprises the entire JSON document.",
7   "default": {},
8   "examples": [
9     {
10      "data": {
11        "launchesPast": [
12          {
13            "mission_name": "Starlink-15 (v1.0)",
14            "launch_date_local": "2020-10-24T11:31:00-04:00",
15            "launch_site": {
16              "site_name_long": "Cape Canaveral Air Force Station Space Launch Complex 40"
17            },
18            "links": {
19              "article_link": null,
20              "video_link": "https://youtu.be/J442-ti-Dhg"
21            }
22          }
23        ]
24      }
25    }
26  ],
27  "required": [
28    "data"
29  ],
30  "properties": {
31    "data": {
32      "$id": "#/properties/data",
33      "type": "object",
34      "title": "The data schema",
35      "description": "An explanation about the purpose of this instance.",
36      "default": {},
37      "examples": [
38        {
39          "launchesPast": [
40            {
41              "mission_name": "Starlink-15 (v1.0)",
42              "launch_date_local": "2020-10-24T11:31:00-04:00",
43              "launch_site": {
44                "site_name_long": "Cape Canaveral Air Force Station Space Launch Complex 40"
45              },
46              "links": {
47                "article_link": null,
48                "video_link": "https://youtu.be/J442-ti-Dhg"
49              }
50            }
51          ]
52        }
53      ]
54    }
55  }
56 }
```

@ailuj876



```

1  const request = require('supertest');
2  var validate = require('jsonschema').validate
3  const expect = require('chai').expect
4
5  const spaceXSchema = require('./spaceXSchema')
6
7  describe('Space GraphQL API -', () => {
8      it('verify getting past launch data', async () => {
9          let res = await request('https://api.spacex.land/graphql')
10         .post('')
11         .send({ 'query': `query {
12             launchesPast(limit: 10) {
13                 mission_name
14                 launch_date_local
15                 launch_site {
16                     site_name_long
17                 }
18                 links {
19                     article_link
20                     video_link
21                 }
12             }
13         }` })
22         .set("Accept", "application/json")
23         .expect("Content-Type", 'application/json; charset=utf-8')
24         .expect(200)
25
26         expect(validate(res.text.data, spaceXSchema).errors.length).to.be.equal(0)
27     })
28 })

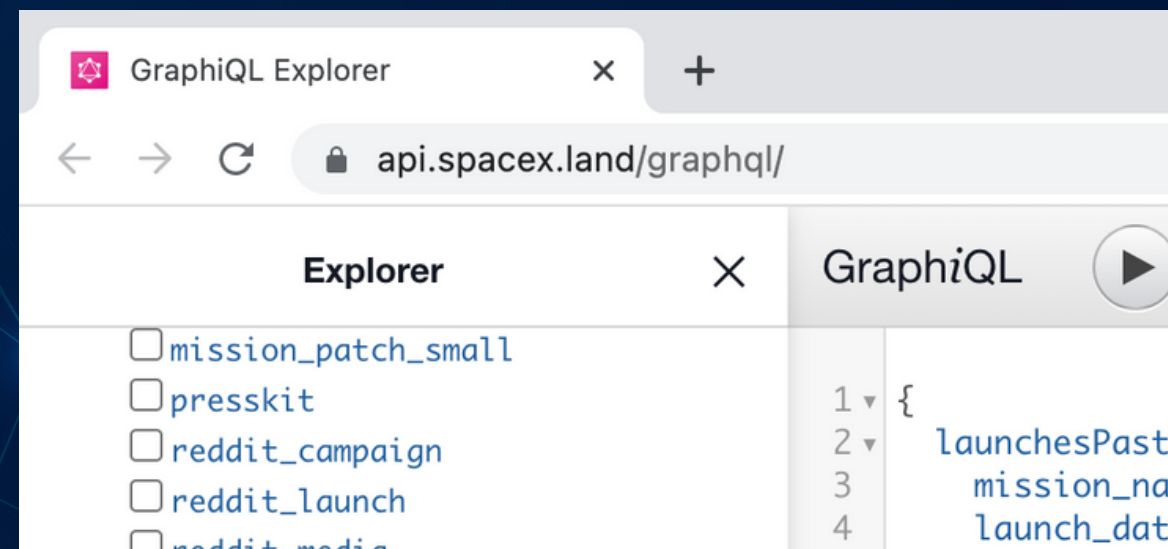
```

# WEBSITES FOR Practise

## Welcome to Restful-Booker

An API playground created by [Mark Winteringham](#) for those wanting to learn more about API testing and tools

[@2bittester](#) | [Website](#) | [Code](#) | [API Docs](#)



## {JSON} Placeholder

Free fake API for testing and prototyping.

Powered by [JSON Server](#) + [LowDB](#)

As of Oct 2021, serving ~1.7 billion requests each month.



<https://petstore.swagger.io/v2/swagger.json>

## Swagger Petstore 1.0.5

[ Base URL: petstore.swagger.io/v2 ]

<https://petstore.swagger.io/v2/swagger.json>

## httpbin.org 0.9.2

[ Base URL: httpbin.org/ ]

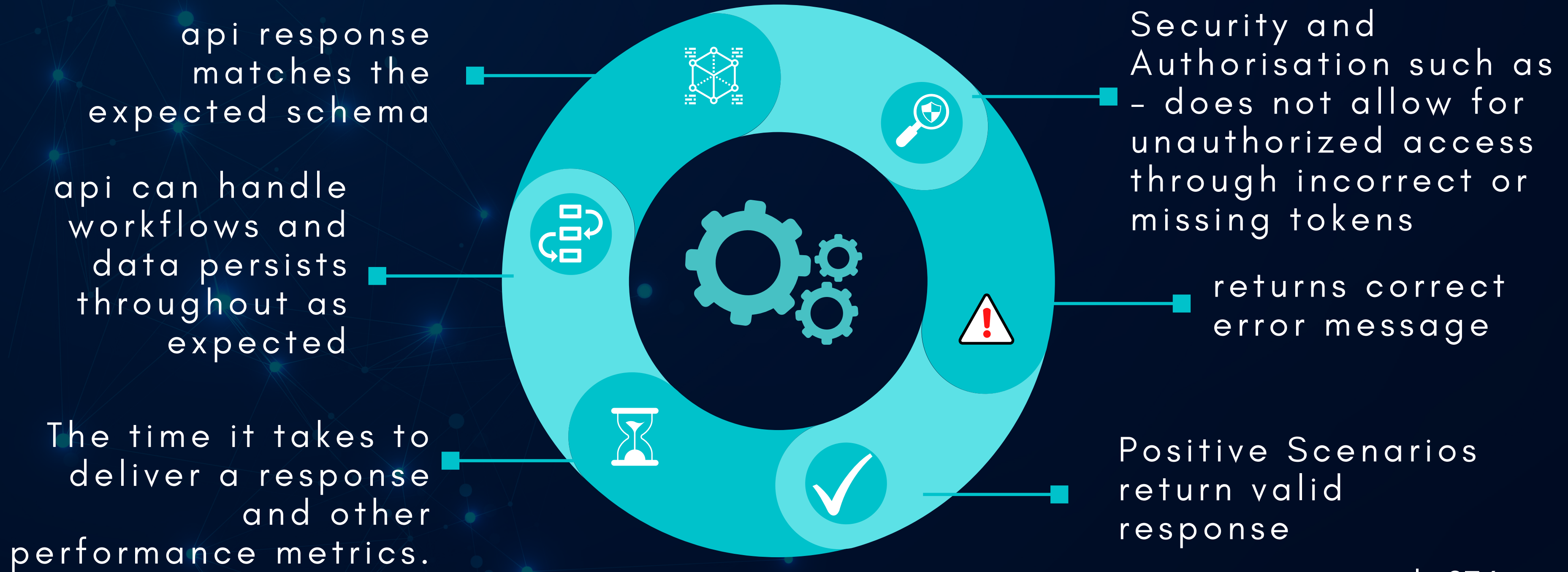
A simple HTTP Request & Response Service.

Run locally: `$ docker run -p 80:80 kennethreitz/httpbin`

@ailuj876



# Scenarios to Automate





# Thank You

## Stay Connected



ailuj876



youtube.com/juliapottinger



juliapottinger.com



linkedin.com/in/julia-pottinger/



TestAutomationU