BETTER API TESTING WITH OPENAPI SPECIFICATION

TAYLOR BARNETT — @TAYLOR_ATX
HI!

👋

LEAD COMMUNITY ENGINEER @ STOPLIGHT
TAYLOR@STOPLIGHT.IO
@TAYLOR_ATX

HI! 👋
Re: Multi month coupons on yearly subscriptions...
https://groups.google.com/a/msg/api-discuss/..GII938RcBAAJ

Nov 23, 2015 - I just ran into an unexpected API behavior, so I wanted to document it here and suggest some possible improvements. If you add a multi-month ...

Unexpected behavior from --config option in
https://github.com/

Apr 28, 2016 - Hi, I upgraded to Notebook 4.2 and use the new server extension deployment method (jupyter serverextension enable --py my_extension), ...

Unexpected behavior when callback throws an error · Issue #1159 ...
https://github.com/

May 10, 2018 - urishortener.url.get(params, (err, res) => { if (err) return console.error('The API returned an error: ' + err); console.log('Long url is $ {res.data.'

Unexpected API error (HTTP code 502): Bad response from
https://github.com/

Jul 19, 2017 - Actual behavior After about 30 minutes or so, my Docker container ... db_1 | LOG: autovacuum launcher shutting down Unexpected API error for ...
Dear Support: your docs don't show examples of how to use you software, just long API lists, which appear inaccurate.

Dear Nick: please look at the API

4:21 AM - 15 Feb 2018

Unit testing is not only for testing *your* code. The unit tests for Textcast found a big unexpected change in behavior of a [REDACTED] API.

9:31 AM - 18 Jun 2009

 Seriously though, [COMPANY] api docs are [BLEEPING] trash. Not only incomplete in every way, but also inaccurate and wrong at times.

10:33 AM - 1 Sep 2014

Nothing like a brand new SaaS that has an API doc that's already incomplete and inaccurate. Just generate your doc and be done with it.

7:15 PM - 14 Mar 2013

Don't you hate it when you spend hours refactoring your code to find out that the problem is with a remote API's unexpected behavior

12:15 PM - 10 Nov 2009

Yay undocumented API features! Gotta love that unexpected behavior!

12:34 PM - 20 Jan 2014 from Lansing, MI
SINGLE SOURCE OF TRUTH 👉
SINGLE SOURCE OF TRUTH

- What does everything (humans and machines) look to?
- Messy, outdated design docs?
- Something that everyone agrees to - a contract
Standard, **structured** approach for describing APIs that is both **human** and **machine** readable
OpenAPI 2.0

- info
  - host
  - basePath
  - schemes
  - produces
  - consumes
  - paths
  - tags
  - externalDocs
  - definitions
  - parameters
  - responses

OpenAPI 3.0

- info
  - servers
  - security
  - paths
  - tags
  - externalDocs
  - components

Image by OpenAPI Initiative
OPENAPI SPECIFICATION

- Development contract
- Prototyping and mocking
- Client SDKs and libraries
- Testing
- Server stubs
OPENAPI SPECIFICATION

- Simplifies describing an API
- Standardizes terminology
- "API Fastness"
OPENAPI SPECIFICATION

- Good tutorial to walk through the specification:
  - https://apihandyman.io/openapi/
“OPENAPI IS A BRIDGE TO UNDERSTANDING AND BEING ABLE TO COMMUNICATE AROUND USING HTTP AS A TRANSPORT, AND OUR GREATEST HOPE FOR HELPING [PEOPLE] LEARN THEIR HTTPS AND 123S.”

Kin Lane (@apievangelist)
DESIGN-FRIST
APIS

@taylor_atx
DESIGN-FIRST APIs

- Consistency for users
- Important for critical APIs
- Code-first can be expensive
- Gain the benefits of using OpenAPI
CONTRACT TESTING
Tests that ensure the API implementation meets the standards and definitions described in a contract.
CONTRACT TESTING

- Popularity has increased with the growth of microservices
- Some kind of contract available to all parties
- Enables independent deployments while not breaking other people’s code
- What it is **not** testing:
  - Service availability
  - Load tolerance
  - Deployment integrity
OPENAPI SPECIFICATION

VALIDATION SERVER

API RESPONSE

PASS OR FAIL
BUT I ALREADY TEST MY APIS... 😐
(SOME) DIFFERENT TYPES OF TESTING

- Integration/Functional Tests
- Unit Tests
- End-to-end Tests
(SOME) DIFFERENT TYPES OF TESTING

- UI TESTING
- INTEGRATION TESTING
- UNIT TESTING

CONTRACT TESTING
EXAMPLE: BROKEN SCHEMA 💔
curl --request GET --url http://api.taylorbar.net/hello-world

200 👍
Test Greetings Scenario (0.009s)
Hello World, GET http://api.taylorbar.net/hello-world (0.009s)
✗ hello: hello is required
✓ output.status (eq) 2xx

✗ 1 of 2 tests failed
```json
{
  "ello": "world"
}

"get": {
  "responses": {
    "200": {
      "description": "A greetings message",
      "schema": {
        "type": "object",
        "required": [
          "hello"
        ],
        "properties": {
          "hello": {
            "type": "string"
          }
        }
      }
    },
    "examples": {}
  },
  "produces": [
    "application/json"
  ],
  "consumes": [
    "application/x-www-form-urlencoded"
  ],
  "parameters": []
}
```
Test Greetings Scenario (0.012s)
Hello World, GET http://api.taylorbar.net/hello-world (0.011s)
✓ output.body (validate.contract) against JSON Schema
✓ output.status (eq) 2xx

✓ 2 tests passed
Todo CRUD Scenario (0.831s)
Create Todo, POST http://todos.stoplight.io/todos?apikey=123 (0.244s)
  ✗ user: user is required
  (root): Must validate all the schemas (allOf)
  ✓ output.status (eq) 201
Get Todo, GET http://todos.stoplight.io/todos/61904 (0.13s)
  ✗ user: user is required
  (root): Must validate all the schemas (allOf)
  ✓ output.status (eq) 200
Update Todo, PUT http://todos.stoplight.io/todos/61904?apikey=123 (0.137s)
  ✗ user: user is required
  (root): Must validate all the schemas (allOf)
  ✓ output.status (eq) 200
Delete Todo, DELETE http://todos.stoplight.io/todos/61904?apikey=123 (0.126s)
  ✓ output.status (eq) 204
  ✓ output.body (validate.contract) against JSON Schema
Make Sure Deleted, GET http://todos.stoplight.io/todos/61904 (0.192s)
  ✓ output.status (eq) 404
  ✓ output.body (validate.contract) against JSON Schema

✗ 3 of 10 tests failed
```json
{
  "completed": null,
  "completed_at": null,
  "created_at": "2018-07-18T18:33:19.341Z",
  "id": 61907,
  "name": "feed Yoda",
  "updated_at": "2018-07-18T18:33:19.341Z"
}
```
Todo CRUD Scenario (0.902s)
Create Todo, POST http://todos.stoplight.io/todos?apikey=123 (0.277s)
✓ output.status (eq) 201
✓ output.body (validate.contract) against JSON Schema
Get Todo, GET http://todos.stoplight.io/todos/61908 (0.141s)
✓ output.status (eq) 200
✓ output.body (validate.contract) against JSON Schema
Update Todo, PUT http://todos.stoplight.io/todos/61908?apikey=123 (0.145s)
✓ output.status (eq) 200
✓ output.body (validate.contract) against JSON Schema
Delete Todo, DELETE http://todos.stoplight.io/todos/61908?apikey=123 (0.129s)
✓ output.status (eq) 204
✓ output.body (validate.contract) against JSON Schema
Make Sure Deleted, GET http://todos.stoplight.io/todos/61908 (0.209s)
✓ output.status (eq) 404
✓ output.body (validate.contract) against JSON Schema

✓ 10 tests passed
EXAMPLE: SDKS
ADD CONTRACT TESTING INTO CI/CD PIPELINE

送信レコード / sendgrid-go

Current  | Branches | Build History | Pull Requests
---|---|---|---
✓ Pull Request #258  gofmt
  Run standard gofmt command on project root.
  - go version 1.10.3 darwin/amd64
  ↗ Commit d6564a4
  ➕ #258: gofmt
  ➕ Branch master
  ➕ ia authored and committed

Build Jobs
- # 503.1  ➕ Go: 1.6  ➕ no environment variables set  ➕ 6 min 53 sec
- # 503.2  ➕ Go: 1.7  ➕ no environment variables set  ➕ 4 min 44 sec
- # 503.3  ➕ Go: 1.8  ➕ no environment variables set  ➕ 12 min 29 sec
- # 503.4  ➕ Go: 1.9  ➕ no environment variables set  ➕ 4 min 39 sec
- # 503.5  ➕ Go: tip  ➕ no environment variables set  ➕ 8 min

#503 passed
- Ran for 12 min 29 sec
- Total time 36 min 45 sec
- About a month ago

@taylor_atx
SDKS

- Helps maintainers
- Ensures API and SDK/libraries parity
- Go see Elmer Thomas’ talk at 5:05pm on “Managing SDKs and their communities in multiple programming languages”
WHAT ABOUT PACT TESTING?
SINGLE SOURCE OF TRUTH PITFALLS
SINGLE SOURCE OF TRUTH PITFALLS

▸ Forgetting the “why”

▸ Failing to version
  ▸ “version” is in the Info Object

▸ Bad communication
COMMUNICATION
THEY BECOME THE CODIFICATION OF A SET OF DISCUSSIONS ABOUT WHAT A SERVICE API SHOULD LOOK LIKE, AND WHEN THEY BREAK, THEY BECOME A TRIGGER POINT TO HAVE CONVERSATIONS ABOUT HOW THAT API SHOULD EVOLVE.

Sam Newman, “Building Microservices: Testing”