

API ESSENTIAL TOOLKIT

10 MUST HAVE TOOLS
FOR API DESIGN &
MANAGEMENT



"

Research is creating new knowledge. To do so effectively, the right tools are as important as the intellect.

Neil Armstrong

**PRAIRIE
DEV CON**
WEB | DEV | CLOUD | AI

 NORTHFIELD

 Johnston
group

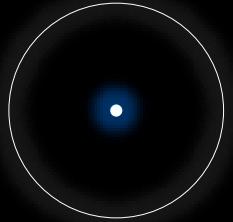
 Lotlinx

 online
business systems

 LAMBERT
NEMEC
GROUP

 CONQUEST

 ATASSIAN



TRAVIS GOSELIN

DISTINGUISHED SOFTWARE ENGINEER

DEVELOPER EXPERIENCE

travisgosselin.com

linkedin.com/in/travisgosselin

@travisjgosselin



SPS COMMERCE

INFINITE RETAIL POWER™

The screenshot displays the SPS Commerce platform's dashboard. At the top, there are four summary cards: 'Pending Partner Acknowledgment' (13), 'New Orders' (218), 'Errors' (21), and 'Ready for Change Acknowledgment' (59). Below these are two main sections: 'Open Orders' and 'Orders Missing a Shipment'. The 'Open Orders' section lists 13 orders from various trading partners, including Dick's Sporting Goods, Apex Sports, Walmart, Cabela's, Champ's Sports, Finish Line, Foot Locker, Gander Mountain, Bass Pro Shops, and Mills Fleet Farm. The 'Orders Missing a Shipment' section lists 218 orders. To the right, a 'RECENT ACTIVITY' sidebar shows a log of events, such as a purchase order being added and vendor compliance requirements being updated.

"

Developer Experience is the activity of studying, improving and optimizing how developers get their work done.

"

theappslab.com (2017)

DEVELOPER EXPERIENCE

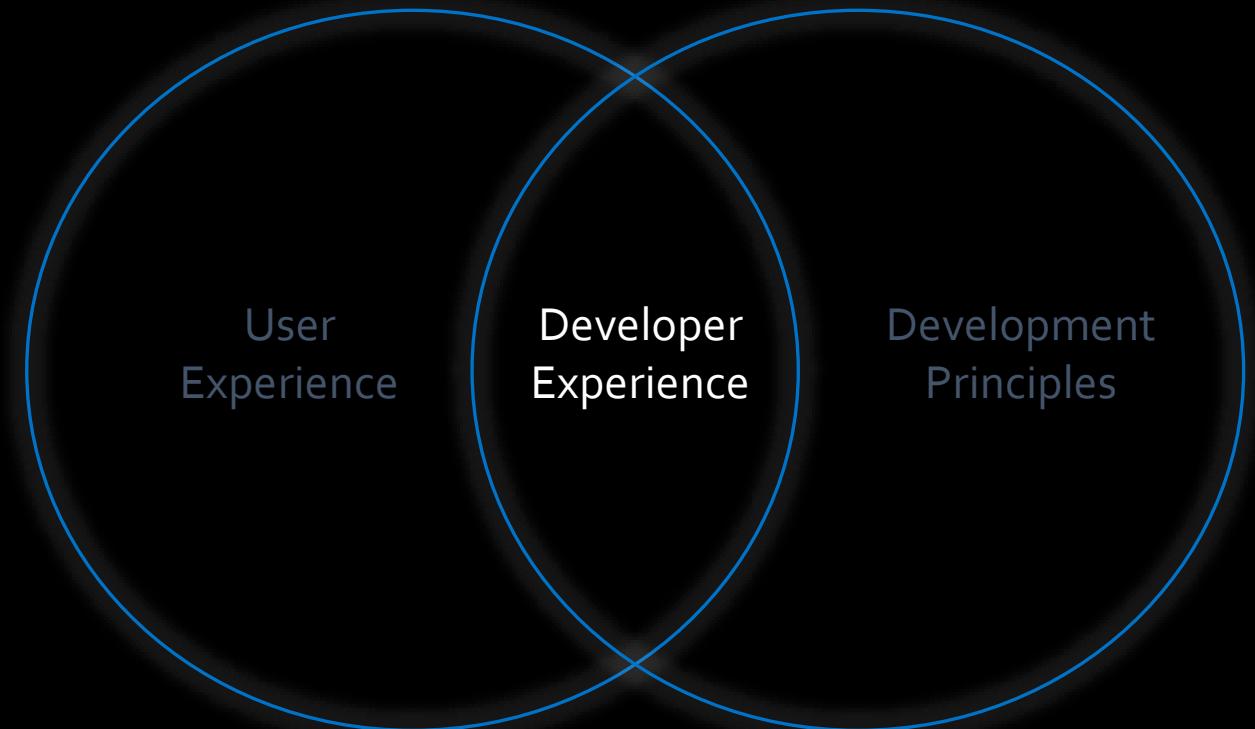
WHAT IS THAT...EXACTLY?

"

Developers work in rainforests, not planned gardens.

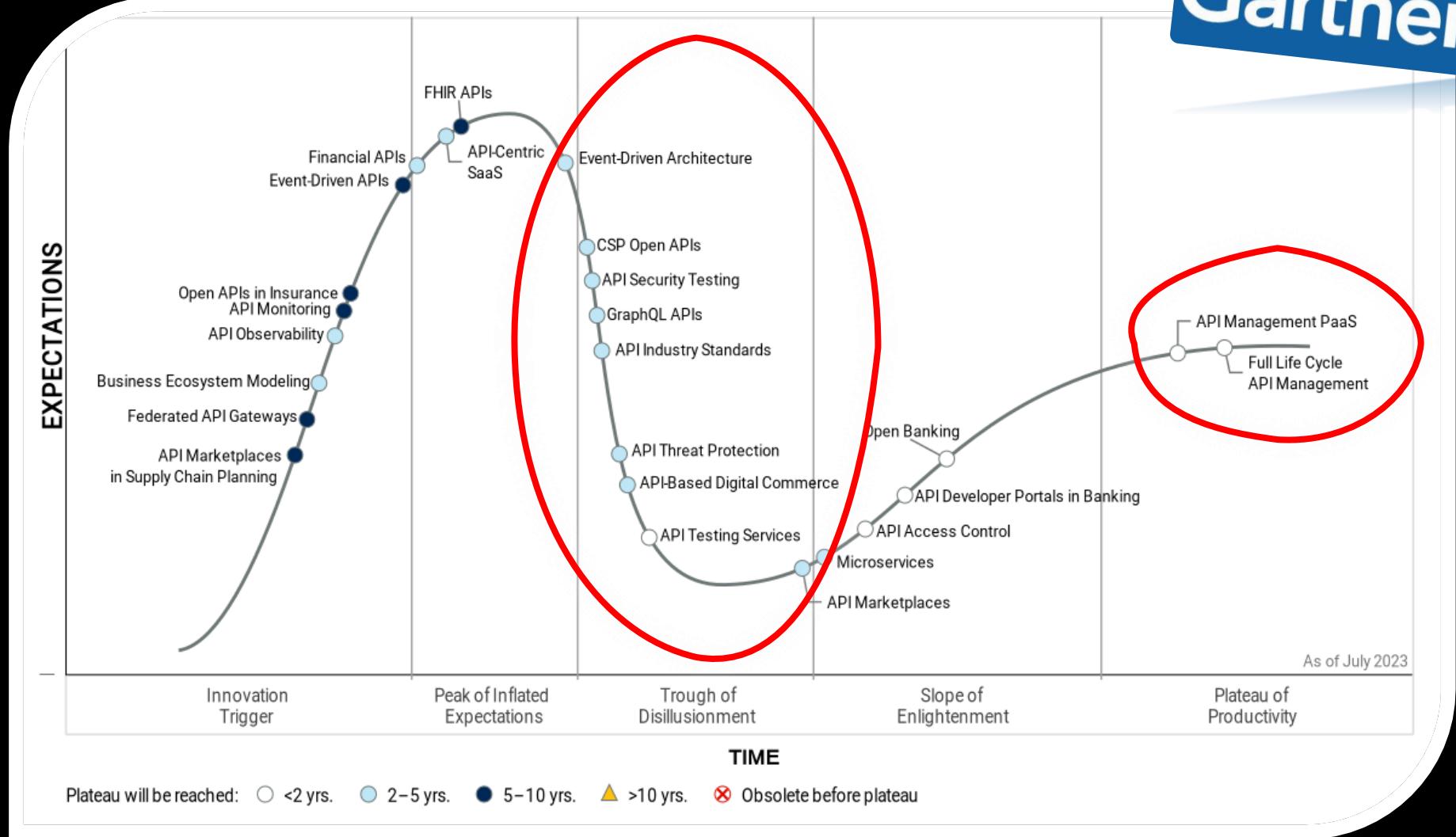
"

a16z.com



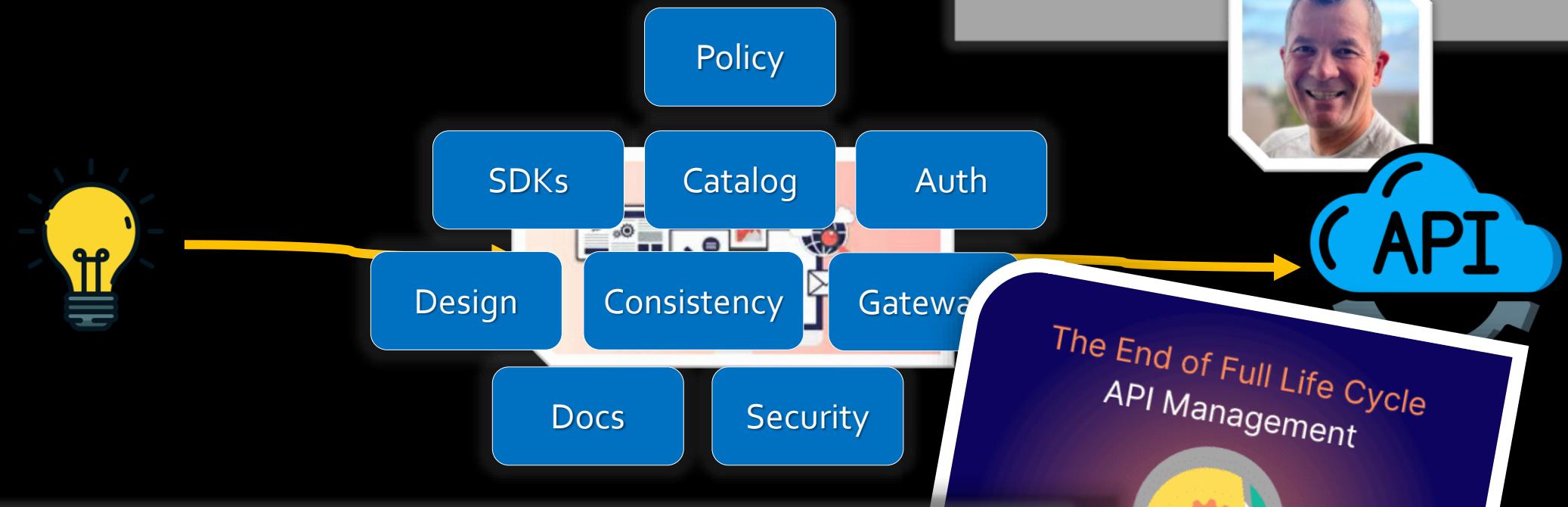
API PRODUCTIVITY OUTLOOK

API Hype Cycle



APICOMPONENTIZATION

Commoditization of API Toolsets



“
As a consumer I don't approach my day considering what I want to unbundle. I go out in the world and obtain what I need. The fact that many of us fall for the bundling of all the products I need into a single grocery store, or fast-food chain, does not make bundling or unbundling a thing.

Kin Lane (API Evangelist)

“

The Great API Unbundling



Erik Wilde

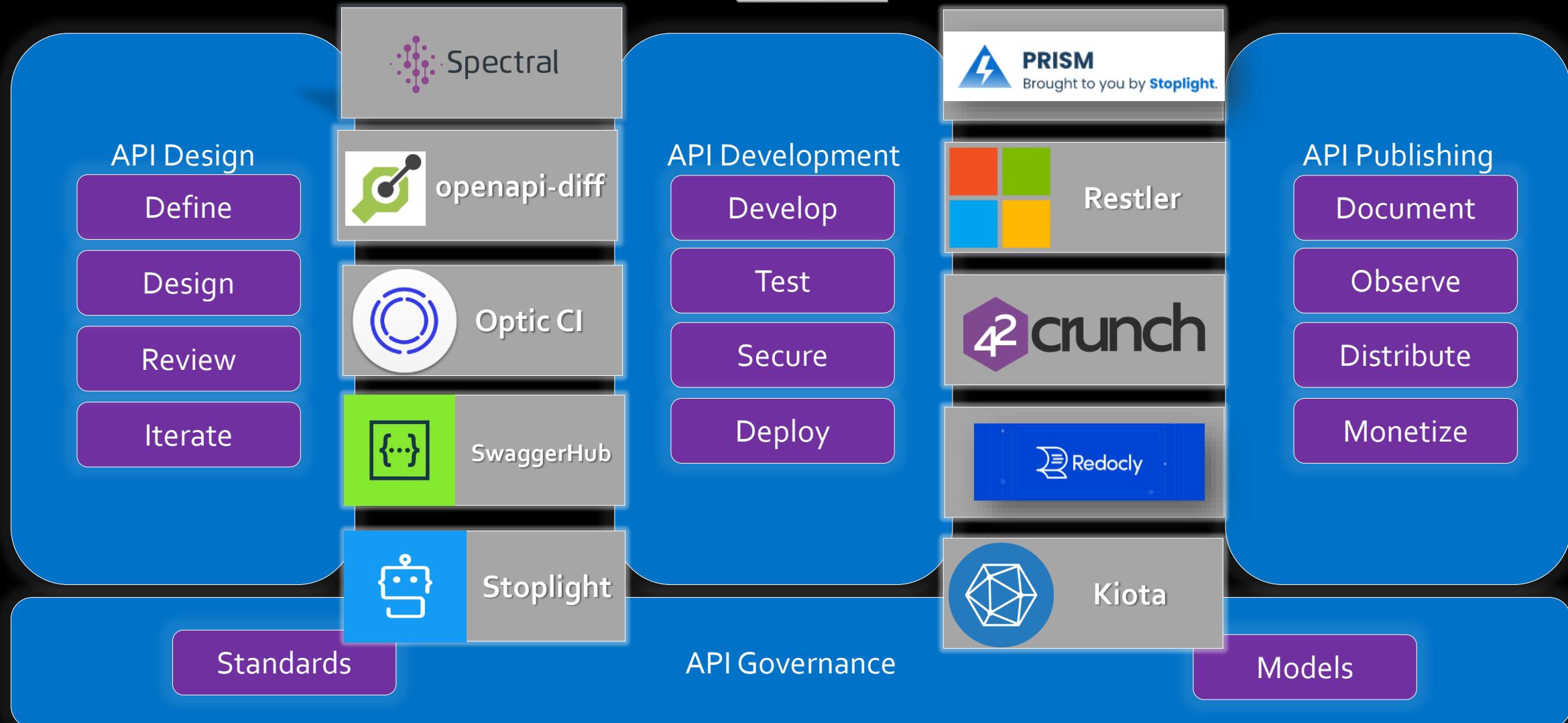
*The End of Full Life Cycle
API Management*



”

API LIFECYCLE

Pillars



Postman being used by 75% of respondents!



An open-source API style guide enforcer and linter.

`spectral lint api.yml --ruleset .spectral.yml`

JSON/YAML Linter with Custom Rulesets

```
--- .spectral.yml
extends:
  - "spectral:oas"
  - your-other-rules.yml
rules:
  operation-description: off
```

Inherit Rulesets

Default OAS Ruleset

Team or Organizational Rulesets

Library of 3rd Party Rulesets

Teams Can Override

Customize Severity

```
problem-details-error-response:
  description: "Every error response SHOULD support RFC 9457"
  severity: error
  given: $.paths.[*].responses[?(@property.match(/^\^(4|5)\$/))].content.*~
  then:
    function: enumeration
    functionOptions:
```



Custom Rules

YAML

JavaScript

```
1:1  warning oas3-api-servers      OpenAPI "servers" must be present and non-empty array.
2:6  warning info-                Info object must have "contact" object.          info
2:6  warning info-                Info object must have "contact" object.          info
7:9  warning op-                 Operation object must be defined in global tags.
7:9  warning op-                 Operation object must be defined in global tags.
9:9  warning op-                 Operation object must be defined in global tags.
20:30 error  unknown-error-format Every error response SHOULD support RFC 7807.      paths./v1/users.get.responses[400].content.application
```



YOU NEED DEFINED API STANDARDS WITH AUTOMATION!

The tool you use matters less, but Spectral has a good community / ecosystem!

OPENAPI-DIFF

Overview



Open API Diff Detection is Critical in Your Automation.

What's Deleted

- POST /thing/{id}

What's Changed

- GET /user/{username}

Return Type:

- Changed 200 OK

Media types:

- Changed application/xml

Schema: Broken compatibility

- PUT /user/{username}

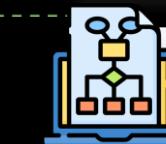
Request:

- Changed application/json

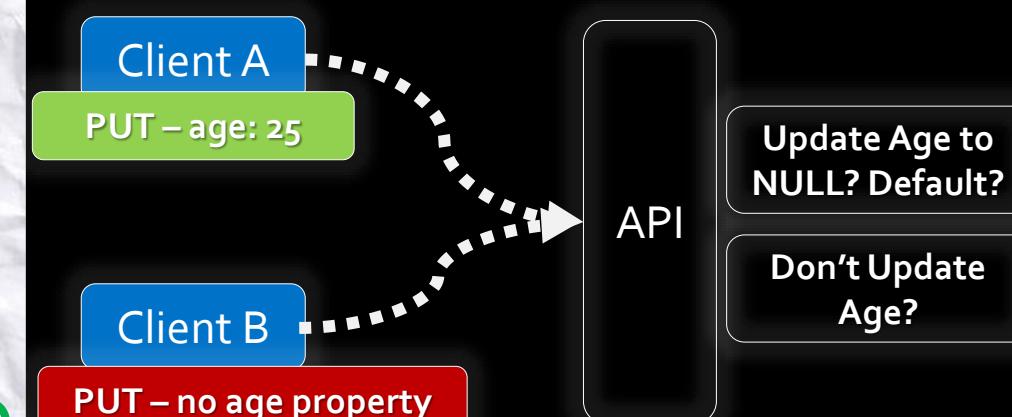
Schema: Backward compatible

Result

API changes broke backward compatibility



User:
type: object
properties:
id:
type: string
required: true
name:
type: string
required: true
age:
type: integer



API Governance

API Development

Open Source

CLI / Java

OpenAPITools/
openapi-diff

Utility for comparing two OpenAPI specifications.

53
Contributors

71
Issues

30
Discussions

717
Stars

153
Forks



API Governance

Community & Paid

API Design / Development

CLI / NPM / Platform



Optic

Developer-friendly API Governance

optic diff openapi.yml --check --web



Optic creates a curated ecosystem of tools that is extremely developer friendly that enables Governance capability but fails to provide Enterprise level enforcement and observability of Governance... yet!

JavaScript Rules

Spectral Support

Incremental Changes Only

API Contract Testing

Reverse Proxy

Traffic vs Open API

Document Existing API

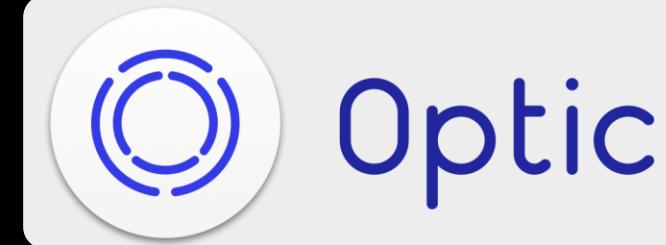


GIT Native



CI / Pull Request

The screenshot shows the Optic UI interface. On the left, there's a sidebar with various tool icons and labels: JavaScript Rules, Spectral Support, Incremental Changes Only, API Contract Testing, Reverse Proxy, Traffic vs Open API, and Document Existing API. The main area displays a Slack message from 'optic-ci' bot: 'commented 4 minutes ago' followed by 'APIs Changed'. It lists changes for the Slack Web API (2 changed) and Payments API (9 changed), with a summary of 6 failed and 21 passed checks. Below this, a detailed error message is shown for a 'internal_account_id' field: 'internal_account_id stringnumber required' with a note about its format being 'uuid'. A warning message states: '[Error] expected request body property 'internal_account_id' to not change type. This is a breaking change. Prevent request property type changes.'



Spectral Rule

```
json-field-names-snake-case:  
  message: Field name must be lower snake_case  
  severity: error  
  given:  
    "$.paths.*.*[responses,requestBody]..content..schema..properties.*~"  
  then:  
    function: pattern  
    functionOptions:  
    match: ^[a-z_][a-z_0-9]*$
```

Optic LintGPT

Write an API Style Guide as bullet points, Optic will enforce it during design and in CI.

```
lintgpt:  
  - Property names must be lower snake case ie "snake_case"  
  - "Responses that return lists must have a pagination offsets"  
  - "MUST use nouns in URIs to represent resources"  
  - "All resources in a response should have an `id` `created_at` property"
```



SWAGGERHUB

Overview

API Governance

API Design

Commercial

UI / API

The Single Source of Truth
for API Development



SMARTBEAR
SwaggerHub..

petStoreSwagger v 1.0.0_Design v

Info

Tags

Search

pet ^

- POST** /pet
- PUT** /pet
- GET** /pet/findByStatus
- GET** /pet/findByTags
- POST** /pet/{petId}
- DELETE** /pet/{petId}
- POST** /pet/{petId}/uploadImage

store ^

- GET** /store/inventory
- POST** /store/order
- GET** /store/order/{orderId}
- DELETE** /store/order/{orderId}

user ^

- POST** /user
- POST** /user/createWithArray
- POST** /user/createWithList
- GET** /user/login
- GET** /user/logout
- GET** /user/{username}
- PUT** /user/{username}
- DELETE** /user/{username}

Models ^

- MODEL Order
- MODEL Category
- MODEL User
- MODEL Tag
- MODEL Pet

```
1 swagger: '2.0'
2  info:
3    description: |
4      This is a sample Petstore server. You can find
5      out more about Swagger at
6      [http://swagger.io](http://swagger.io) or on
7      [irc.freenode.net, #swagger](http://swagger.io/irc/).
8      version: "1.0.0_Design"
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 Pets
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 our store
30           url: http://swagger.io
31     # schemes:
32     # - http
33     paths:
34       /pet:
35         post:
36           tags:
37             - pet
38             - application/json
39             - application/xml
40             summary: Add a new pet to the store
41             operationId: addPet
42             consumes:
43               - application/json
44               - application/xml
45             parameters:
46               - in: body
47                 name: bodydf wefwefwfef wef
48                 schema:
49                   $ref: '#/definitions/Pet'
50             responses:
51               '201':
52                 description: Pet object that needs to be added to the store
53                 required: true
54               '400':
55                 description: Invalid input
56               '404':
57                 description: Pet not found
58               '500':
59                 description: Internal server error
```

Last Saved: 4:08:41 pm - Jan 24, 2020

VALID

...

Centralized Catalog

Swagger Petstore

1.0.0_Design

[Base URL: virtserver.swaggerhub.com/SwaggerOrg/petStoreSwagger/1.0.0]

This is a sample Petstore server. You can find
out more about Swagger at
<http://swagger.io> or on
<irc.freenode.net, #swagger>.

Terms of service
Contact the developer
Apache 2.0
Find out more about Swagger

Schemes

HTTPS

Authorize

pet Everything about your Pets

Find out more: <http://swagger.io>

POST /pet Add a new pet to the store

PUT /pet Update an existing pet

GET /pet/findByStatus Finds Pets by status

GET /pet/findByTags Finds Pets by tags

POST /pet/{petId} Updates a pet in the store with form data

DELETE /pet/{petId} Deletes a pet

POST /pet/{petId}/uploadImage uploads an image

store Access to Petstore orders

API First Product

OpenAPI / AsyncAPI

Collaboration

Interactive

Mocking

Documentation

SWAGGERHUB

Governance & Reuse

```
openapi: 3.0.1
info:
  title: HTTP API Design Management API
  description: Access to Open API spec catalog
  version: gh-main
  x-sps-service-id: bc9b3ffe-3023-4c52-a407-d35f384276c0
  x-sps-repository: http-api-design-mgmt-api
paths:
  /pipelines/api-design/v1/docs/projects/{serviceId}:
    get:
      operationId: get-project-by-service-id
      summary: Get Project by Service ID
      description: Retrieve a documentation project by service ID
      parameters:
        - name: serviceId
          schema:
            type: string
      responses:
        '200':
          description: Success
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DocumentationProject'
        '404':
          $ref: https://api.swaggerhub.com/domains/spscommerce/Errors/2.0#/components/responses/NotFound
        '500':
          $ref: https://api.swaggerhub.com/domains/spscommerce/Errors/2.0#/components/responses/Error
```



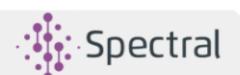
SwaggerHub doesn't visually look the best, but the features it provides are foundational to any API Program without being overly complicated with incredibly reasonable pricing!

```
openapi: 3.0.0
info:
  title: SPS Errors Domain
  description: Common components for Standard API Error responses.
  version: "2.0"
components:
  responses:
    Error:
      description: 'Internal Server Error'
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/Error'
      example:
        title: Internal Server Error
        status: 500
        requestId: b6d9a290-9f20-465b-bcd3-4a5166eeb3d7
        detail: Request for resource failed unexpectedly.
        instance: https://example.com/account/12345/resource/23
        context: [
```

Versioning & Lifecycle

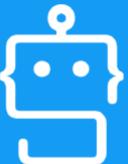
Impact Analysis

Standardization



Prevent Save / Publishing

Reusable Models



Stoplight is a powerful and opinionated API Design platform that has innovated a ton of new capabilities in the space but still requires some maturity in being more flexible.

The screenshot shows the Stoplight API Design interface. On the left, there's a sidebar with navigation links like Home, Examples, and Connect to. The main area displays an API endpoint for retrieving user information:

- Request:** GET `http://localhost:3000/users/{userId}`. Description: Retrieve the information of the user with the matching user ID.
- Path Parameters:** `userId` integer (required). Description: Id of an existing user.
- Responses:** 200, 404. Description: User Found.
- Body:** application/json. Fields:
 - `id` integer (required). Description: Unique identifier for the given user.
 - `firstName` string (required).
 - `lastName` string (required).
 - `email` string<email> (required).
 - `dateOfBirth` string<date> (required).

On the right side of the interface, there are sections for Parameters, Send API Request (with a Live Server button), Request Sample (Shell / cURL), Response Example (JSON code), and other developer tools.

Markdown + OpenAPI Spec(s)

Search

Git Integration

Dynamic Mocking

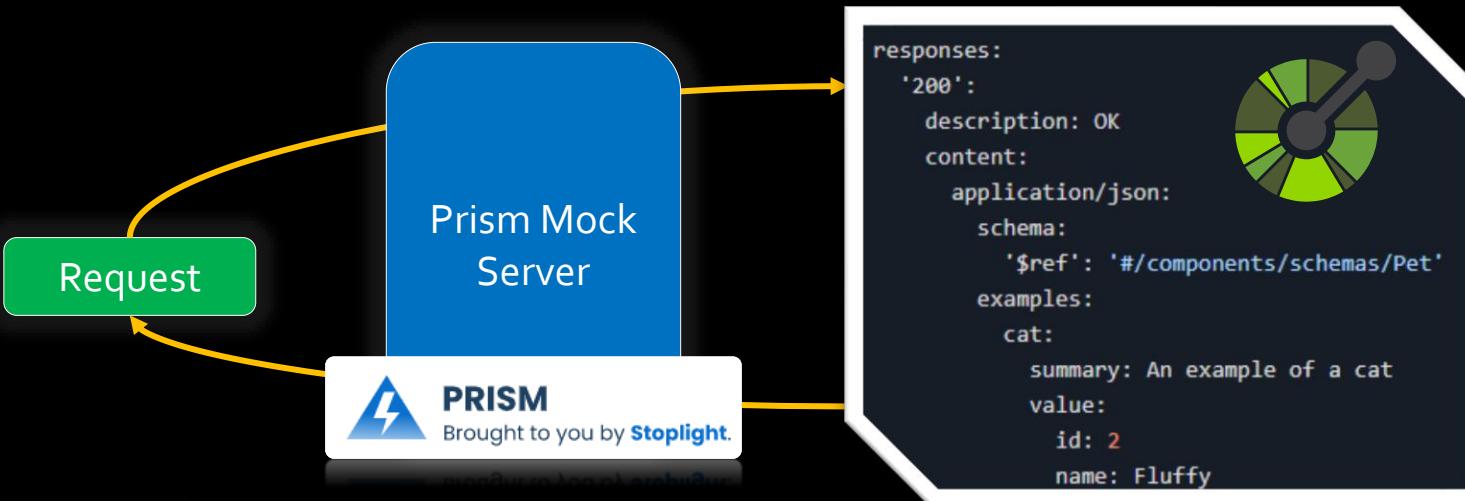
Interactive & Example Code

CLI for Automation



An open-source [HTTP mock and proxy server](#).

```
prism mock -p 4010 myopenapi.yaml
```

[OpenAPI Examples](#)[Example Selection](#)

```
curl /pets/123 -H "Prefer: example=cat"
```

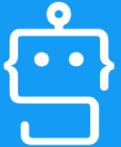
```
prism mock --ignoreExamples myopenapi.yaml
```

Static[Default Value](#)[Examples](#)[NULL](#)[Format Compliant](#)["string"](#)**Dynamic**

```
ten-or-eleven:
  type: number
  example: 10
  x-faker:
    datatype.number:
      min: 10
      max: 11
```

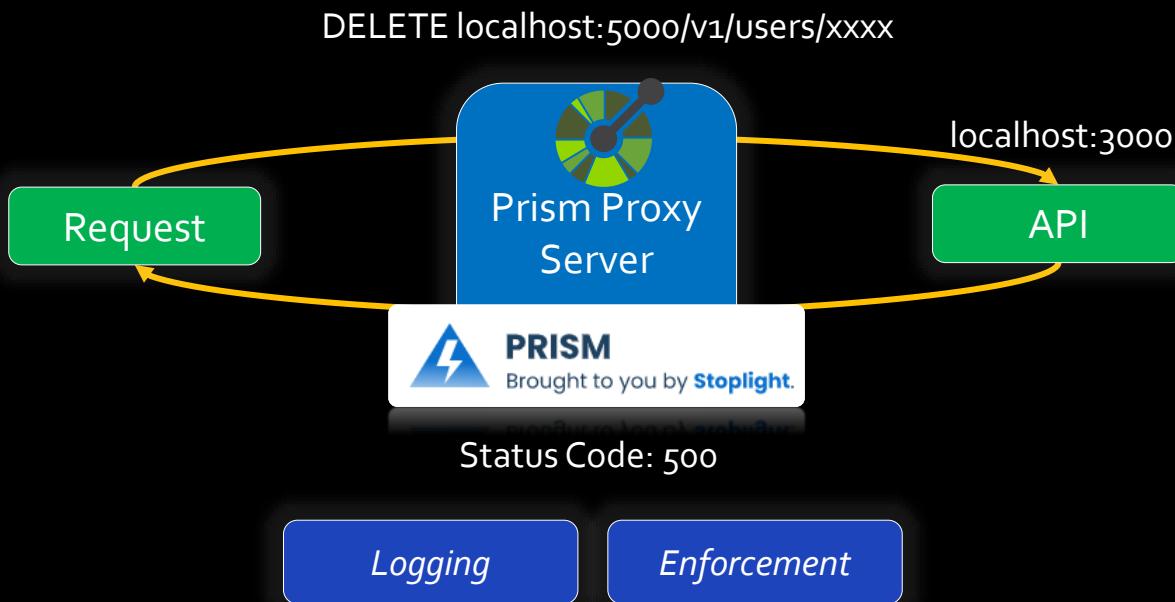
X-faker

```
slug:
  type: string
  example: two-words
  x-faker:
    helpers.slugify: [ "two words" ]
```



An open-source HTTP mock and proxy server.

```
prism proxy -p 5000 myopenapi.yaml localhost:3000 --errors
```



#1 API Design First Concern:
Does my spec match my actual
implementation?



Contract Testing

Build vs Runtime

Mock Not Implemented Operations (501)

Request Validation

RESTLER

Overview

API Development

Open Source

API Governance

CLI / Python / .NET

microsoft/restler-fuzzer



RESTler is the first stateful REST API fuzzing tool for automatically testing cloud services through their REST APIs and finding security and reliability bugs in these services.



OpenAPI
Spec

compile

Reliability



test

fuzz-lean

API



RESTler is a very promising approach to commoditized conformance and contract testing without having to write custom tests for each resource, but is clearly open source and not enterprise ready out of the box.

Restler.exe compile -api_spec api.oas.yml

Compile

Restler.exe test --grammar_file grammar.py
--dictionary_file dict.json --settings
engine_settings.json --no_ssl

Smoketest

Restler.exe fuzz-lean ...

Quick Fuzz

Restler.exe fuzz ...

Takes Hours

Full Fuzz

Request coverage (successful / total): 6 / 17
Attempted requests: 17 / 17
Bugs were found!
Bug buckets:

AP

A

API coverage.



42Crunch is hands-down the best fully-integrated experience for security, governance and quality to accompany any API program if your budget supports it.

API Audit

Open API Spec Static Analysis Scanner



API Scan

Open API Dynamic Scanning Tool



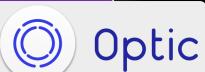
API Protect

Dynamic Runtime Firewall based on Open API Spec



API Capture

Capture Live Traffic to Assemble an Open API Spec



Security audit score: 61 / 100

Security: 15 / 30

Code validation: 46 / 70

Planned validation cycle: 2024-01-15

Issues: 15 / 30

Severity: Medium

Score: 61 / 100

TECH RADAR (2024): 42 Crunch Conformance Scan

- Transporting API keys in a header over network allowed
- Issue ID: v3-global-security-scheme-a-pikey-inheader
- sample2.yaml:60: Security: Info.

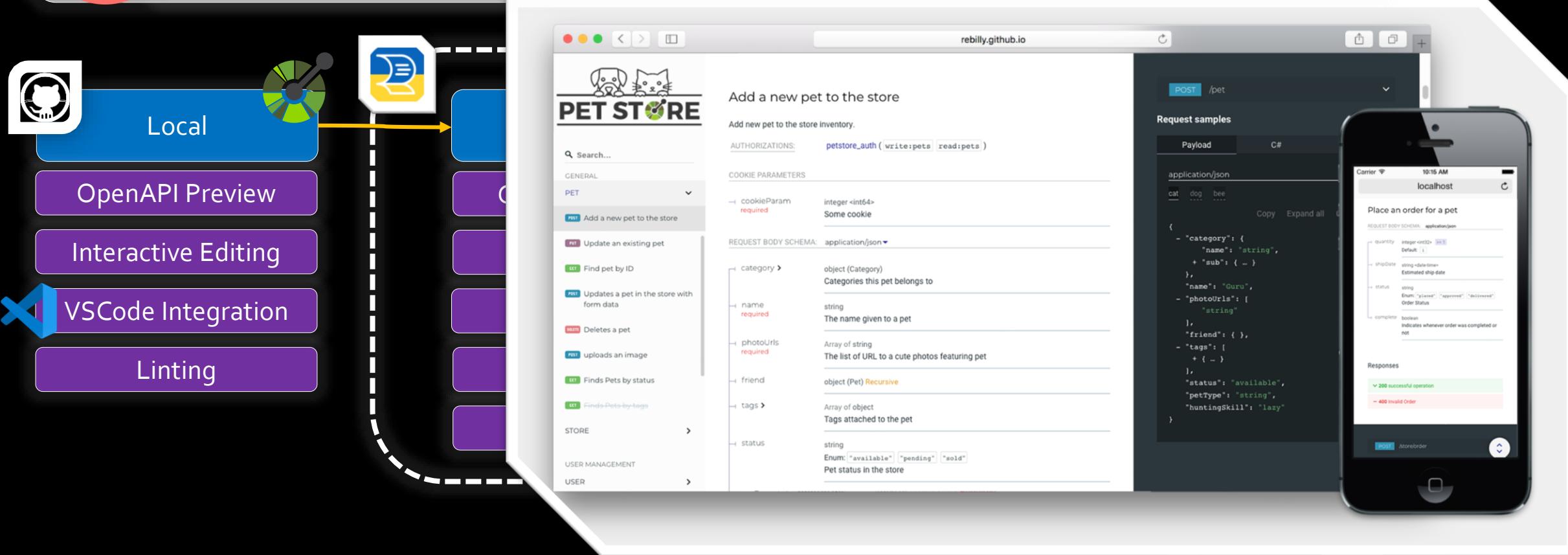
OpenAPI

```
{
  "swagger": "2.0",
  "info": {
    "title": "Pixi API",
    "version": "1.0"
  },
  "tags": [
    {
      "name": "admins",
      "description": "Secured Admin-only calls"
    }
  ],
  ...
}
```

42Crunch Platform



Redocly ecosystem is refreshing in their end-to-end developer to portal product, but it is pretty inflexible, limited features, and has a lot of waitlist features...



The diagram illustrates the Redocly ecosystem. On the left, a vertical stack of colored boxes represents various tools and integrations: a GitHub icon (Local), a megaphone icon (OpenAPI Preview), a gear icon (Interactive Editing), a Visual Studio Code icon (VSCode Integration), and a purple box (Linting). A dashed arrow points from the Local box to a central vertical stack of API definitions. This stack includes sections for PET, STORE, and USER, each with several endpoints listed. A solid arrow points from the Local box to the top of this stack. To the right of the definitions is a screenshot of the Redocly interface, showing the 'Add a new pet to the store' endpoint. It displays authorization details ('petstore_auth (write:pets read:pets)'), cookie parameters ('cookieParam required'), and a detailed request body schema for 'category'. The schema defines 'category' as an object (Category) with properties like 'name' (string) and 'photoUrls' (Array of string). Below the interface is a screenshot of an iPhone displaying the 'Place an order for a pet' screen, which shows a similar request body schema for ordering a pet.

Local

OpenAPI Preview

Interactive Editing

VSCode Integration

Linting

PET

STORE

USER

Add a new pet to the store

AUTHORIZATIONS: petstore_auth (write:pets read:pets)

COOKIE PARAMETERS: cookieParam required

REQUEST BODY SCHEMA: application/json

category

name

photoUrls

friend

tags

status

POST /pet

Request samples

Payload C#

application/json

cat dog bee

Responses

200 successful operation

400 Invalid Order

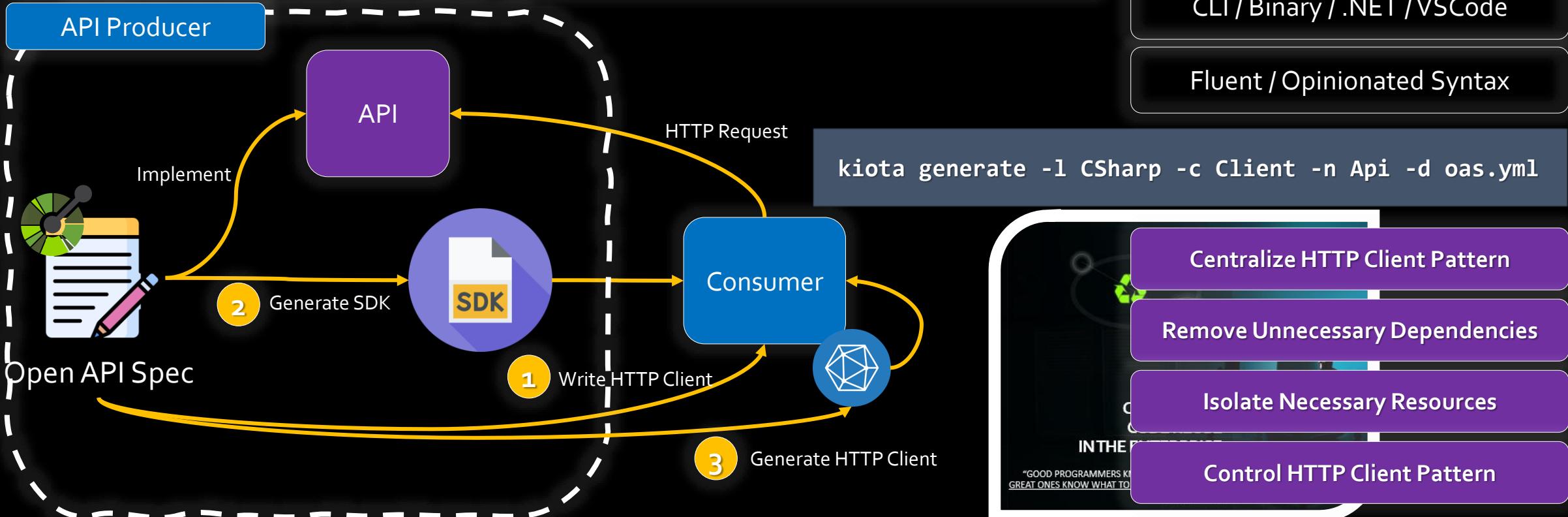
POST /store/order



Kiota shows a ton of potential to be the standardized enterprise approach to HTTP Client usage and adoption, whether internal or external, but is still in its infancy.

Registry

Integration / Update



API ESSENTIALS TOOLKIT

Capabilities

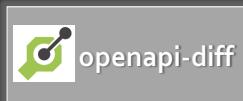
Linting



Conformance



Breaking Change Detection



Documentation



API Catalog



THE GREAT UNBUNDLING

Cable TV Cord Cutting

The internet promised to be the great unbundler. The evisceration of distribution costs would free consumers from paying for cable television channels they never watched, allowing them to bypass cable monopolies and pay only for the shows they wanted.

July 2022 – More Streaming Viewers than Cable TV

81% of 65 and above US adults and 34% of 18 to 29-year-olds watch cable TV

86.72% of US households with a \$150,000 income are cable TV subscribers.



Large Organizations Continue with Full Service Providers?

Affects Who is Producing Next Generation Practices?

Will API Management Come Full Circle?

Future of Full Life Cycle API Management?



API ESSENTIALS TOOLKIT

Inventor of the World Wide Web:

The right tool can make something impossible possible.

Tim Berners-Lee



Spectral

openapi-diff



Stoplight

42crunch

Optic CI

PRISM
Brought to you by [Stoplight](#).

Redocly

SwaggerHub

Restler

Kiota

10 MUST HAVE TOOLS FOR API DESIGN & MANAGEMENT

PRAIRIE DEV CON
CLOUD | MOBILE | WEB | DEV



TRAVIS GOSSELIN

travisgosselin.com

linkedin.com/in/travisgosselin

@travisjgosselin

