POSTMAN

Design, Develop & Mock APIs with Postman

PRESENTED BY
Kaustav Das Modak
Developer Advocate, Postman

@kaustavdm

POSTMAN

What we will cover

Concepts
- Real-life API collaboration
- API Design: What we need
- Postman fundamentals

Role: Producer
- Creating collections and Mocks
- Using “API” and schema

Role: Consumer
- Using Mocks
- Collaborating with producer

POSTMAN

Part 1: Concepts

POSTMAN

Let’s talk about APIs

POSTMAN

Applications today are built of multiple interacting components

POSTMAN

When systems talk to each other, we should carefully design how they interact

POSTMAN

Design your APIs before you implement them

POSTMAN

Agree on domain boundaries and data models before you implement them

POSTMAN

Agree on domain boundaries and data models before you implement them

$\Uparrow$

Collaborate

Business use case
Abstraction
Encapsulation

Agree on <u>domain boundaries</u> and data models before you implement them

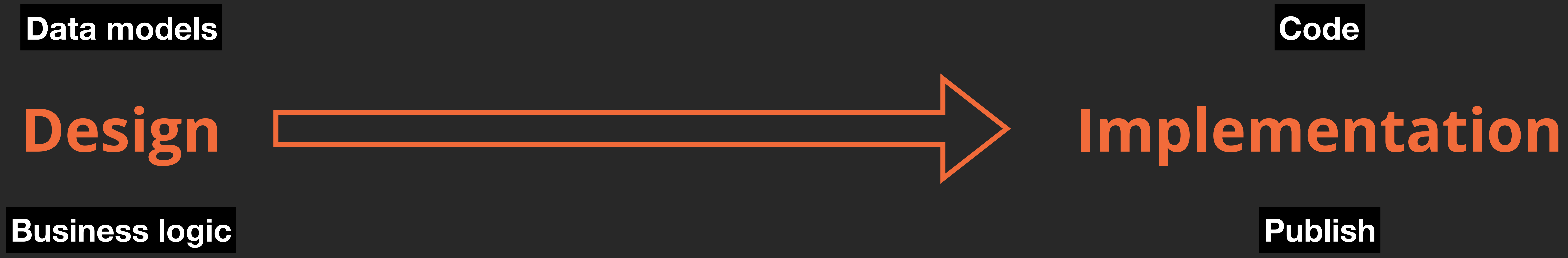
Collaborate

Business use case
Abstraction
Encapsulation

Entities
Resources
Data structures

Agree on domain boundaries and data models before you implement them

Collaborate


Data models
Design
Business logic
Code
Implementation
Publish

POSTMAN

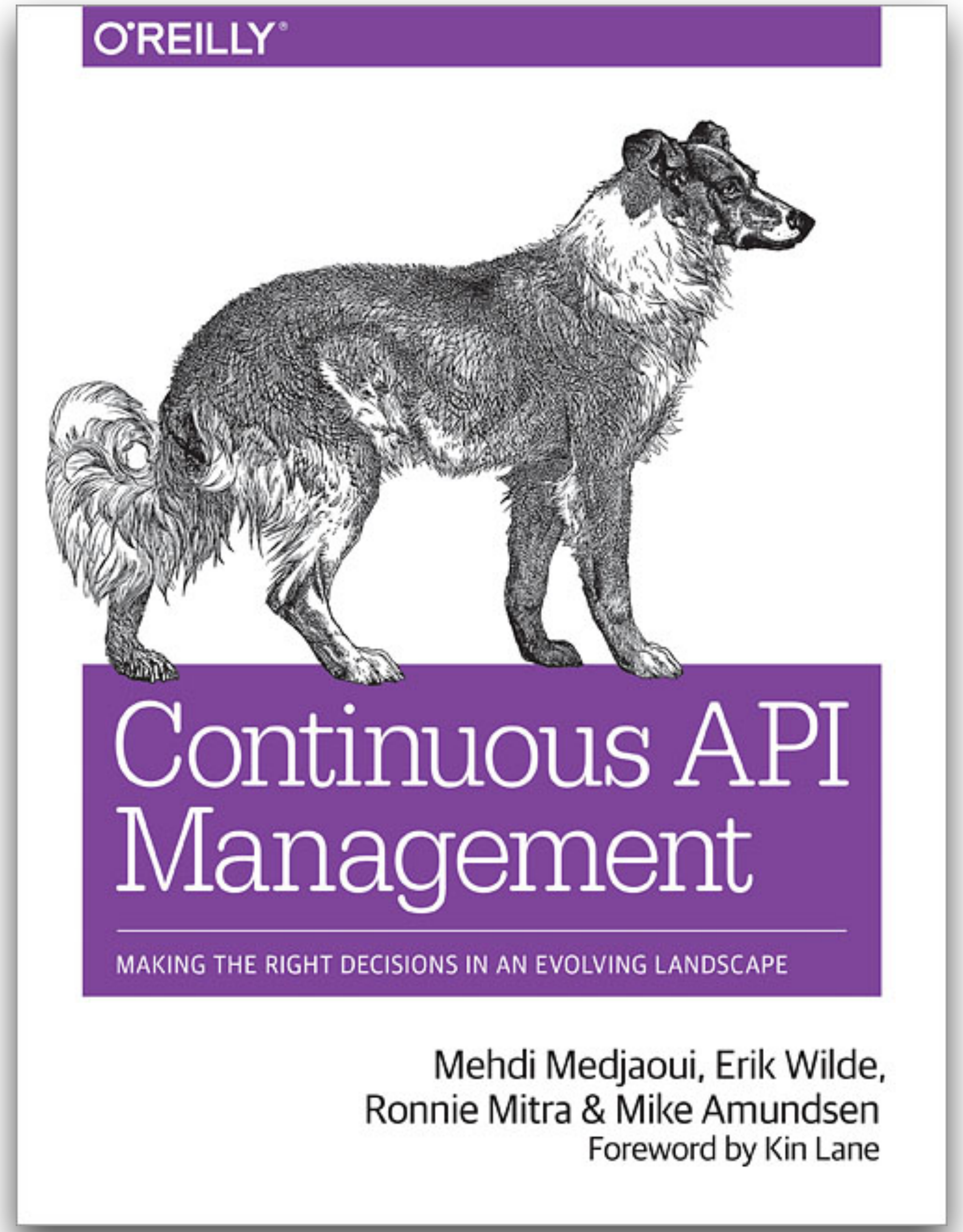
A good design should adapt to changes

POSTMAN

Key concepts

POSTMAN

If you work with APIs...



O'REILLY

Continuous API Management

MAKING THE RIGHT DECISIONS IN AN EVOLVING LANDSCAPE

Mehdi Medjaoui, Erik Wilde, Ronnie Mitra & Mike Amundsen

Foreword by Kin Lane

What is an API?

POSTMAN

What is an API?

Interface

Schema
OpenAPI
RAML
GraphQL
Postman

POSTMAN

What is an API?

Interface	**Implementation**
Schema	**Code**
OpenAPI	Databases
RAML	Repositories
GraphQL	Microservices
Postman	...

POSTMAN

What is an API?

Interface	Implementation	Instance
Schema	**Code**	**Deployed**
OpenAPI	Databases	Interface +
RAML	Repositories	Implementation +
GraphQL	Microservices	Servers
Postman	...	...

What is an API?

Interface	Implementation	Instance
Schema	**Code**	**Deployed**
OpenAPI	Databases	Interface +
RAML	Repositories	Implementation +
GraphQL	Microservices	Servers
Postman	...	...

POSTMAN

What is an API?

Interface

Schema
OpenAPI
RAML
GraphQL
Postman

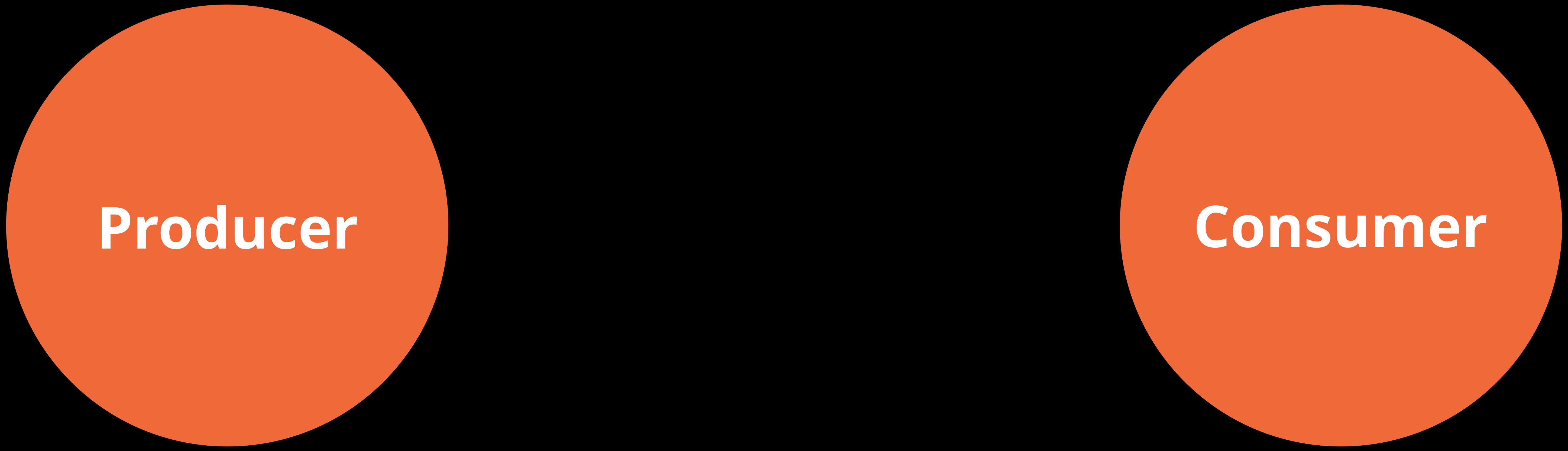
Implementation

Code
Databases
Repositories
Microservices
...

Instance

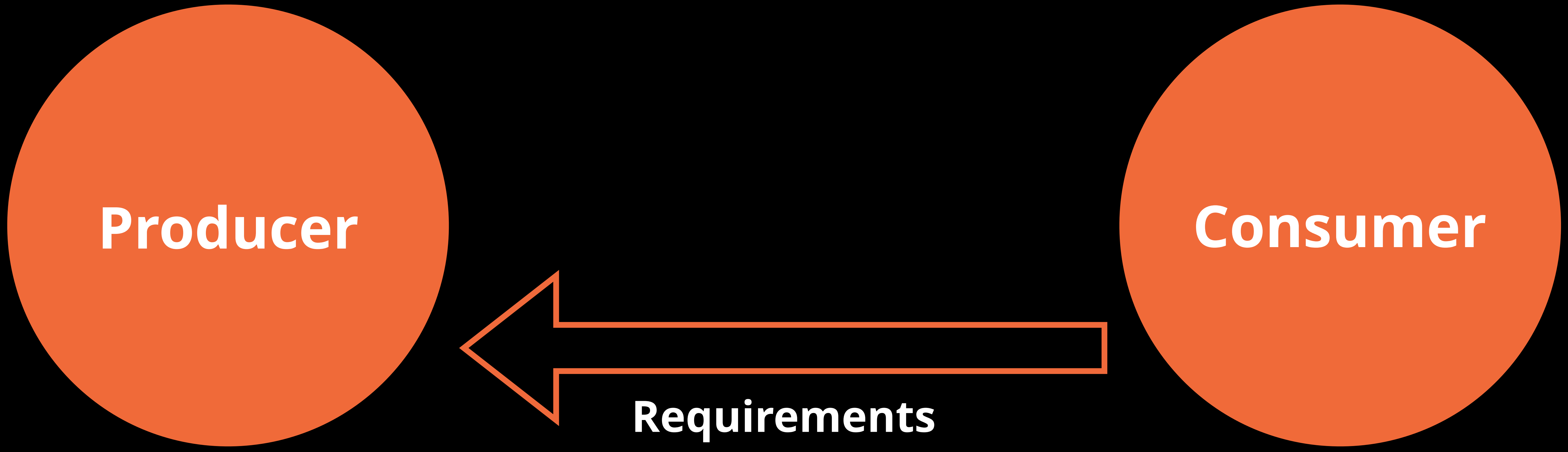
Deployed
Interface +
Implementation +
Servers
...

POSTMAN



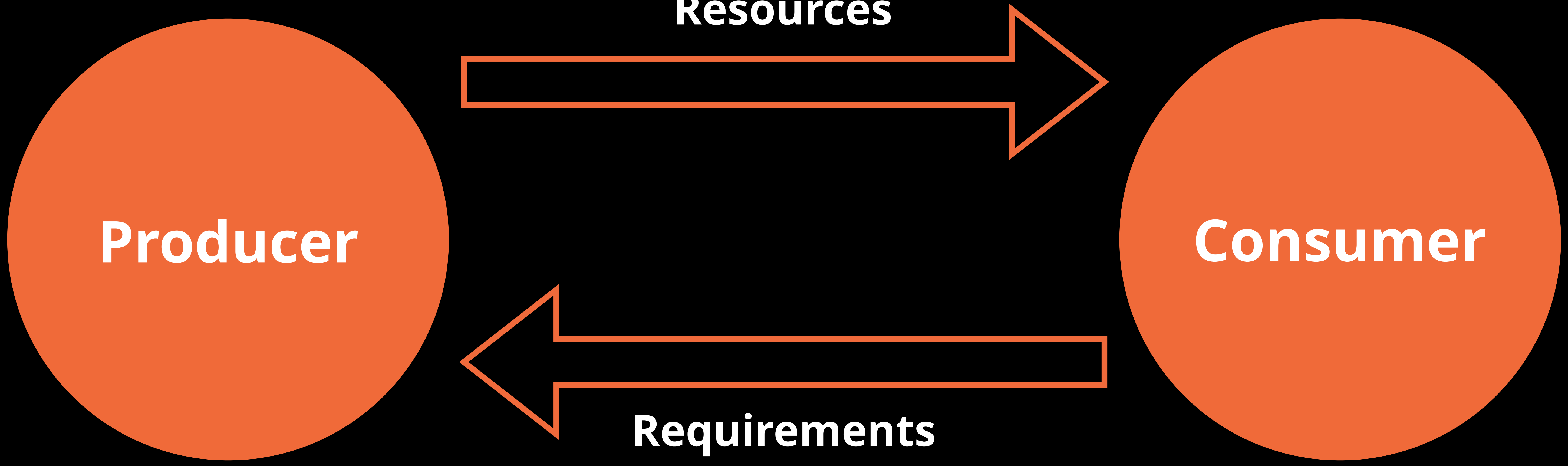
Roles

POSTMAN



Roles

POSTMAN



Roles

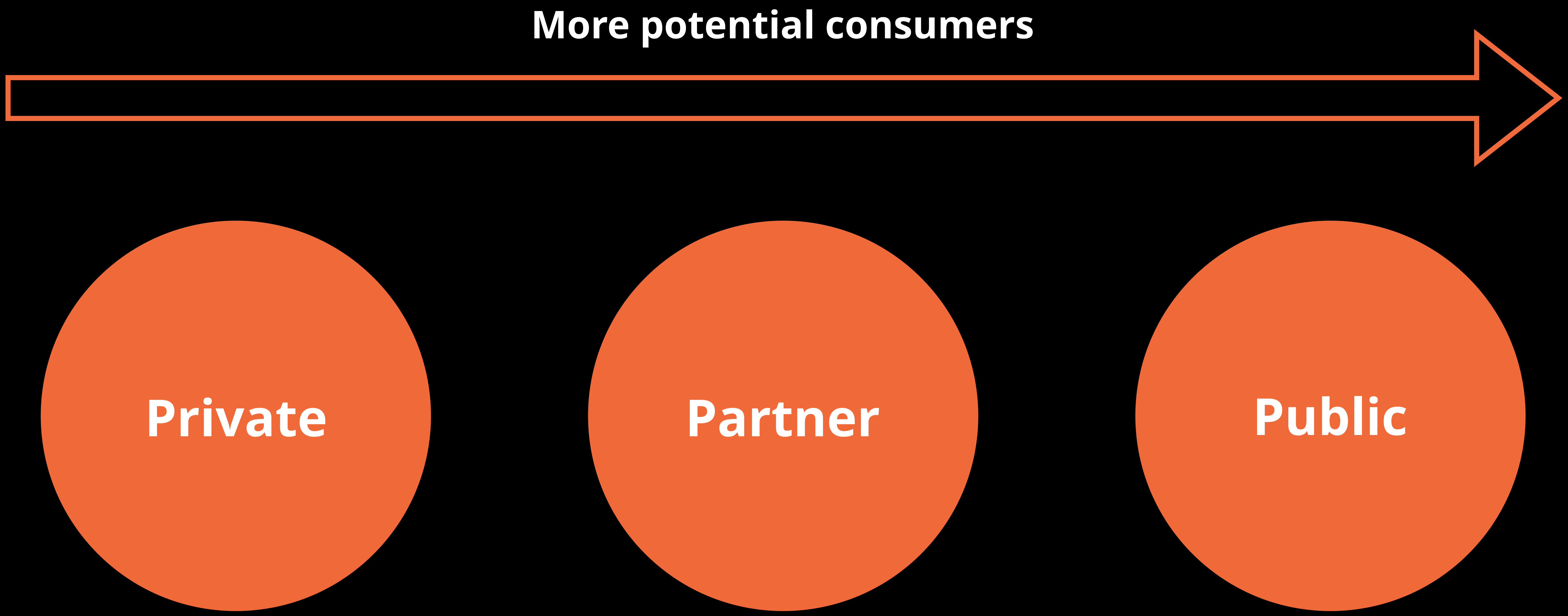
POSTMAN

Private

Partner

Public

Scopes



Scopes

POSTMAN

Schema

POSTMAN

OpenAPI
2.0 | 3.0

GraphQL

RAML

Postman
Collections

WSDL

API Blueprint

Schema

POSTMAN

Questions to ask

Producers

How do we gather requirements?

How do we share API docs?

How do we collect feedback?

Consumers

How do we give our inputs?

How do we consume docs?

How do we test our requirements?

POSTMAN

What we need for effective API design

Design the interface
Write API schema
Build a testable, executable spec
Collaborate on decisions

Document the interface
Resource & usage descriptions
Request/Response examples

Collaborate on implementation
API Mocks
API Contracts

POSTMAN

What is the single source of truth?

POSTMAN

Part 2: Setting things up

POSTMAN

```
GET postman.com
```

POSTMAN

What we will build today

Design API for a hypothetical service to manage list of cats.

As both producer and consumer.

POSTMAN

v0.1

Routes:

```
- GET /cats
  Returns list of all cats
- POST /cats
  Add a new cat
```

Cat schema:

```
- id: Integer
- name: String
- breed: String
- age: Integer
```

POSTMAN

v0.2

Routes (new):

```
– GET /cats/{{catId}}
  Find a specific cat
```

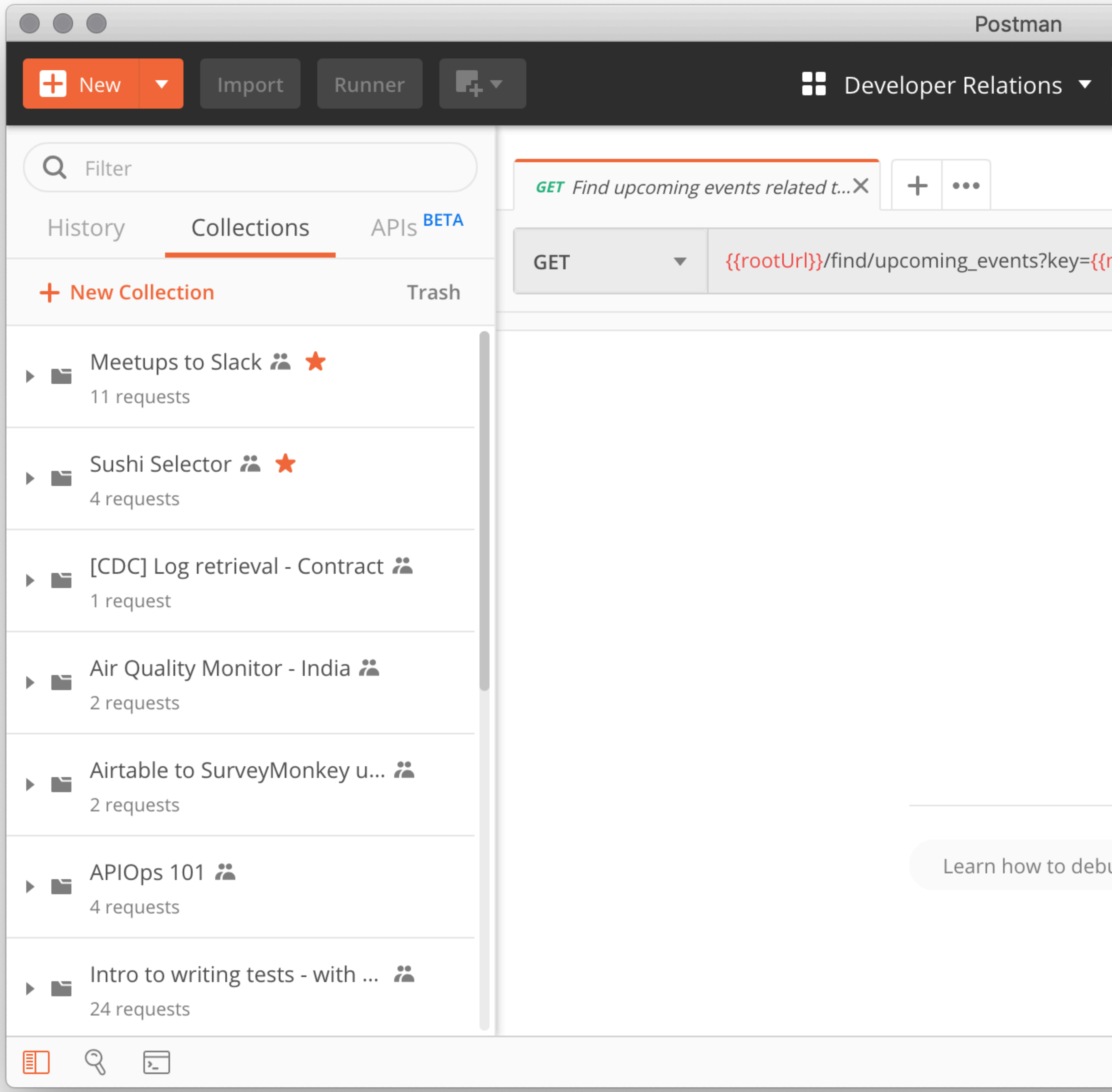
POSTMAN

Part 3: Postman fundamentals

POSTMAN

Collections

Group and organize your requests into meaningful collections.



Meetups to Slack
11 requests

Sushi Selector
4 requests

[CDC] Log retrieval - Contract
1 request

Air Quality Monitor - India
2 requests

Airtable to SurveyMonkey u...
2 requests

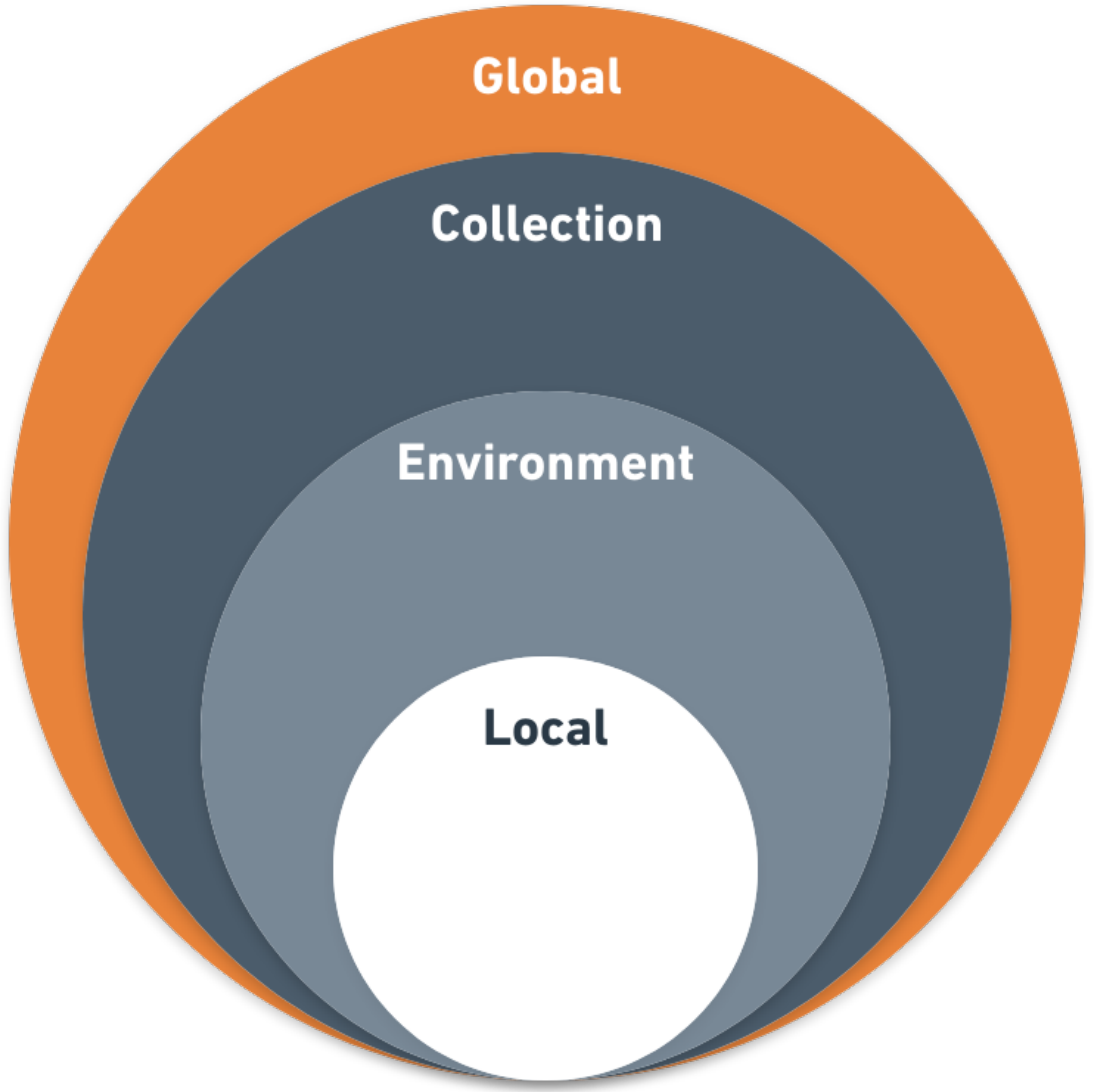
APIOps 101
4 requests

Intro to writing tests - with ...
24 requests

POSTMAN

Variables

Foundation of dynamic values for requests. Can be manipulated programmatically.



POSTMAN

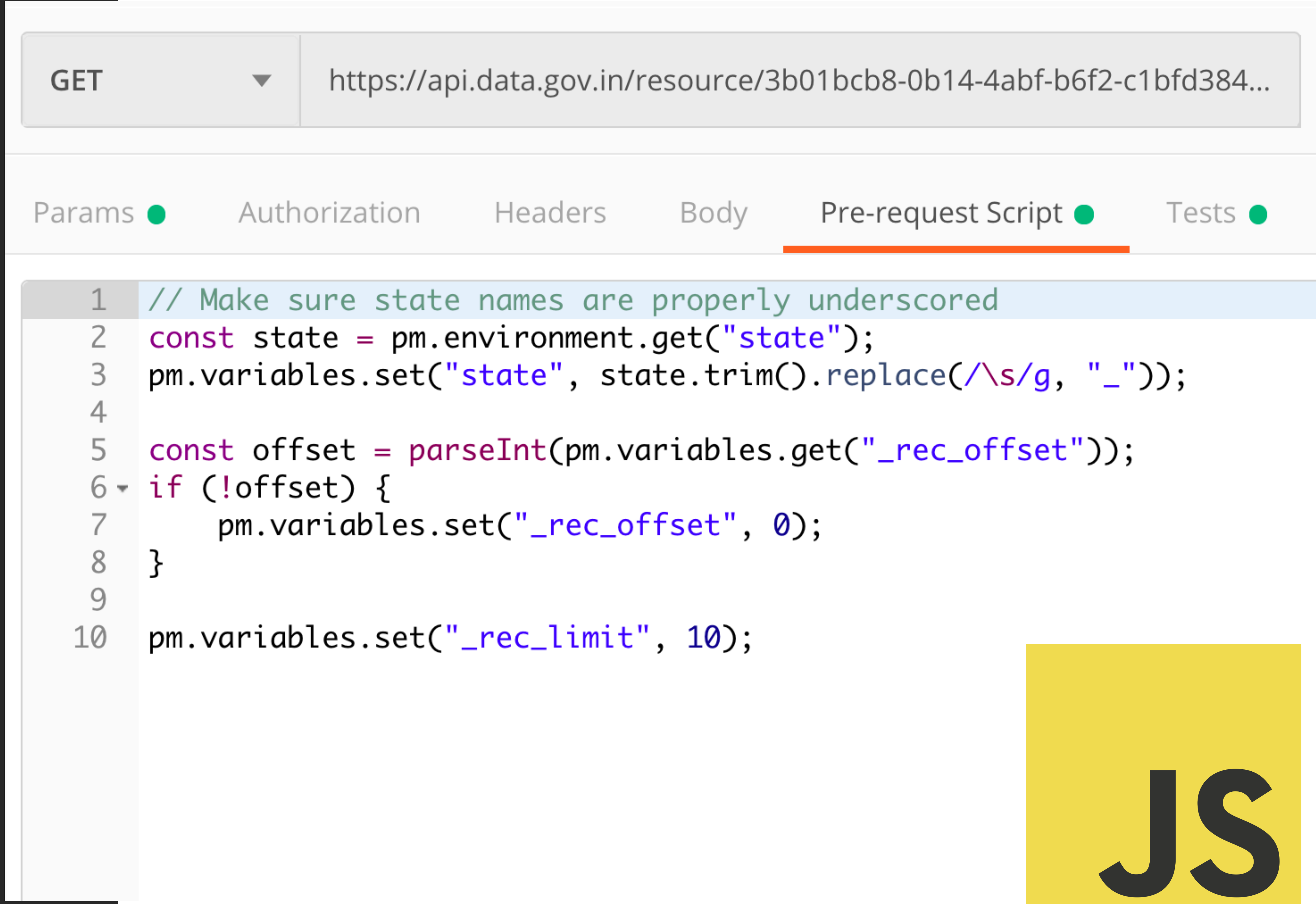
Workspaces

- Organised by service and function
- Service producers and consumers share their collections in them

POSTMAN

Pre-request Script

- Written in JavaScript
- Executed in a sandboxed NodeJS environment
- **Executed *before* request is sent**
- **Modify request through variables**



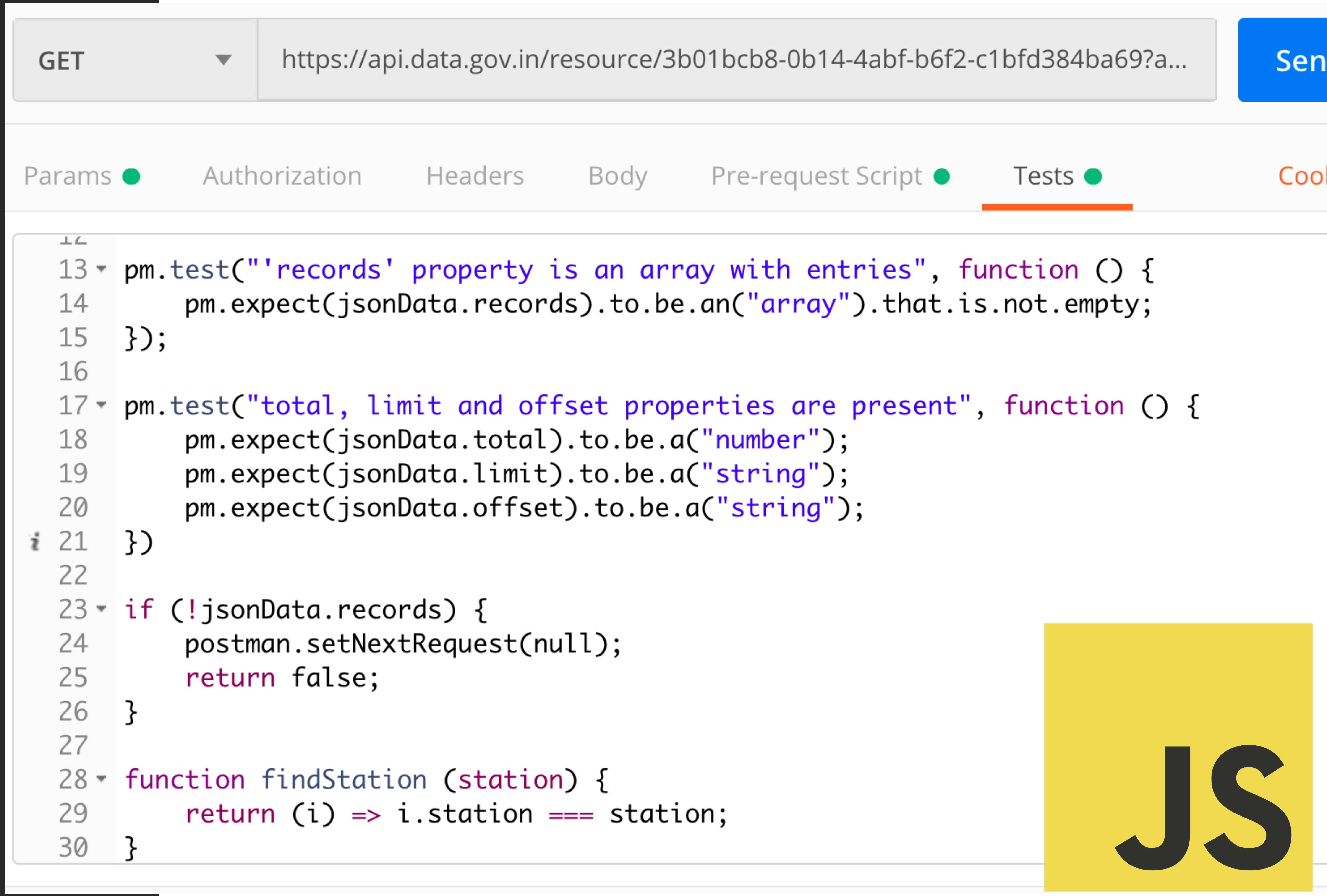
GET https://api.data.gov.in/resource/3b01bcb8-0b14-4abf-b6f2-c1bfd384...

```
// Make sure state names are properly underscored
const state = pm.environment.get("state");
pm.variables.set("state", state.trim().replace(/\s/g, "_"));

const offset = parseInt(pm.variables.get("_rec_offset"));
if (!offset) {
    pm.variables.set("_rec_offset", 0);
}

pm.variables.set("_rec_limit", 10);
```

POSTMAN

Tests

- Written in JavaScript
- Executed in a sandboxed NodeJS environment
- **Executed *after* response is received**
- **Can have assertions**
- Quick-start snippets


GET
https://api.data.gov.in/resource/3b01bcb8-0b14-4abf-b6f2-c1bfd384ba69?a...
Sen
Params
Authorization
Headers
Body
Pre-request Script
Tests
Coo
JS

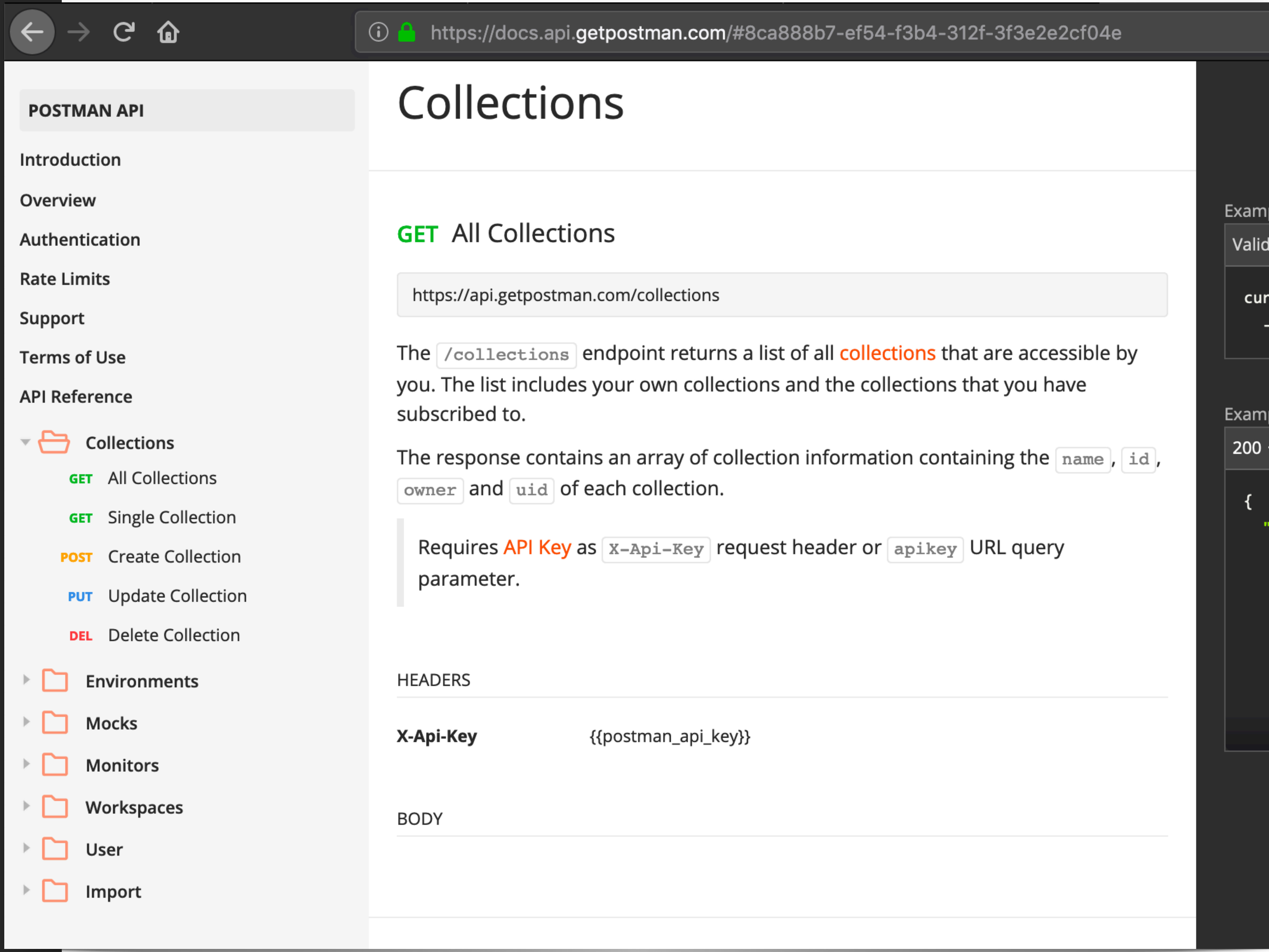
```
pm.test("'records' property is an array with entries", function () {
    pm.expect(jsonData.records).to.be.an("array").that.is.not.empty;
});

pm.test("total, limit and offset properties are present", function () {
    pm.expect(jsonData.total).to.be.a("number");
    pm.expect(jsonData.limit).to.be.a("string");
    pm.expect(jsonData.offset).to.be.a("string");
})

if (!jsonData.records) {
    postman.setNextRequest(null);
    return false;
}

function findStation (station) {
    return (i) => i.station === station;
}
```

POSTMAN

Postman API

Programmatically interact with elements in the Postman ecosystem

`api.getpostman.com`



https://docs.api.getpostman.com/#8ca888b7-ef54-f3b4-312f-3f3e2e2cf04e

POSTMAN API

- Introduction
- Overview
- Authentication
- Rate Limits
- Support
- Terms of Use
- API Reference
- Collections
  - GET All Collections
  - GET Single Collection
  - POST Create Collection
  - PUT Update Collection
  - DEL Delete Collection
- Environments
- Mocks
- Monitors
- Workspaces
- User
- Import

Collections

GET All Collections

https://api.getpostman.com/collections

The `/collections` endpoint returns a list of all collections that are accessible by you. The list includes your own collections and the collections that you have subscribed to.

The response contains an array of collection information containing the `name`, `id`, `owner` and `uid` of each collection.

Requires API Key as `X-Api-Key` request header or `apikey` URL query parameter.

HEADERS

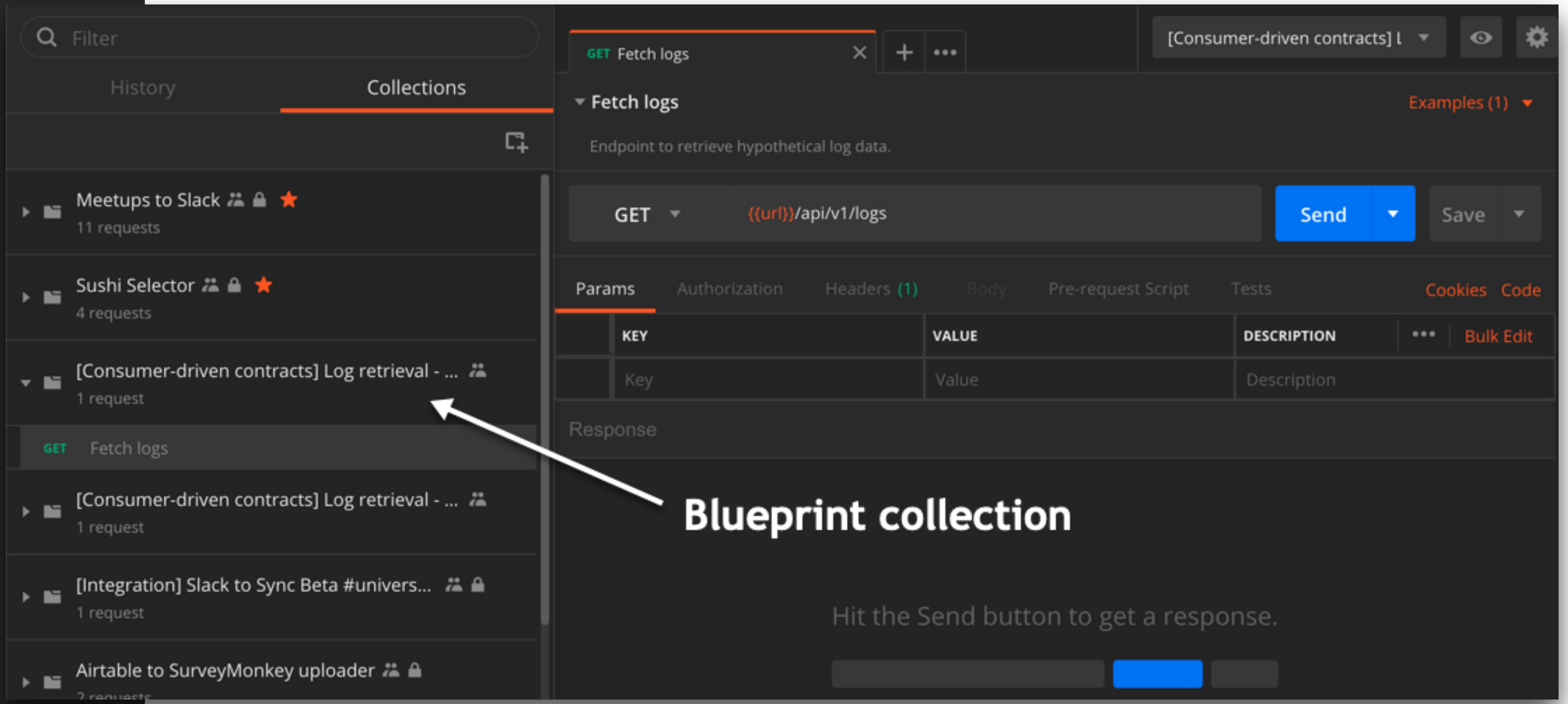
X-Api-Key	{{postman_api_key}}

BODY

POSTMAN

Blueprints

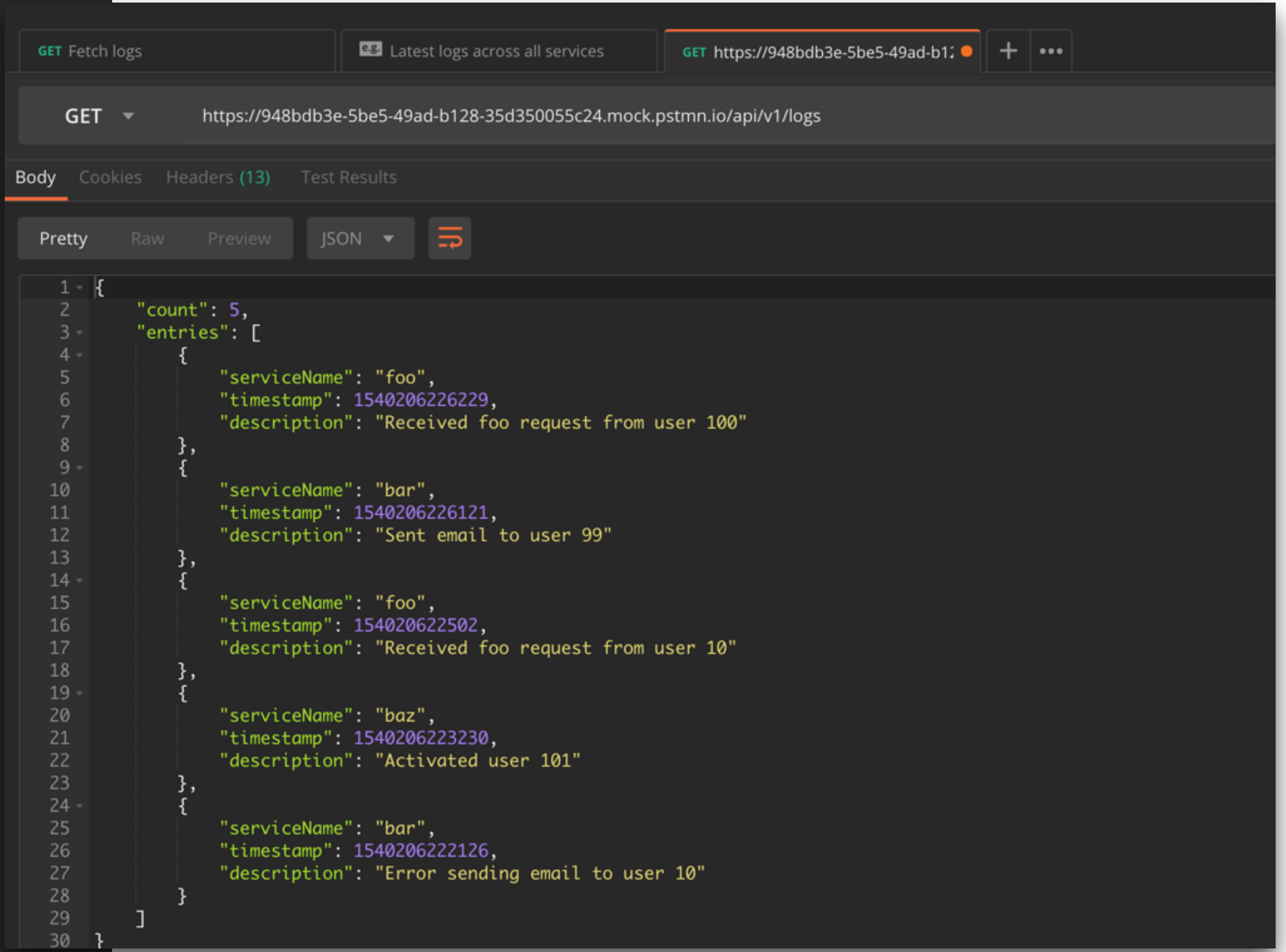
- **Collections** created by service producers to describe an API
- Includes **examples** of each request to document responses



POSTMAN

Mock servers

- Created by service producers from blueprint collections
- Used by service consumers to test API contracts



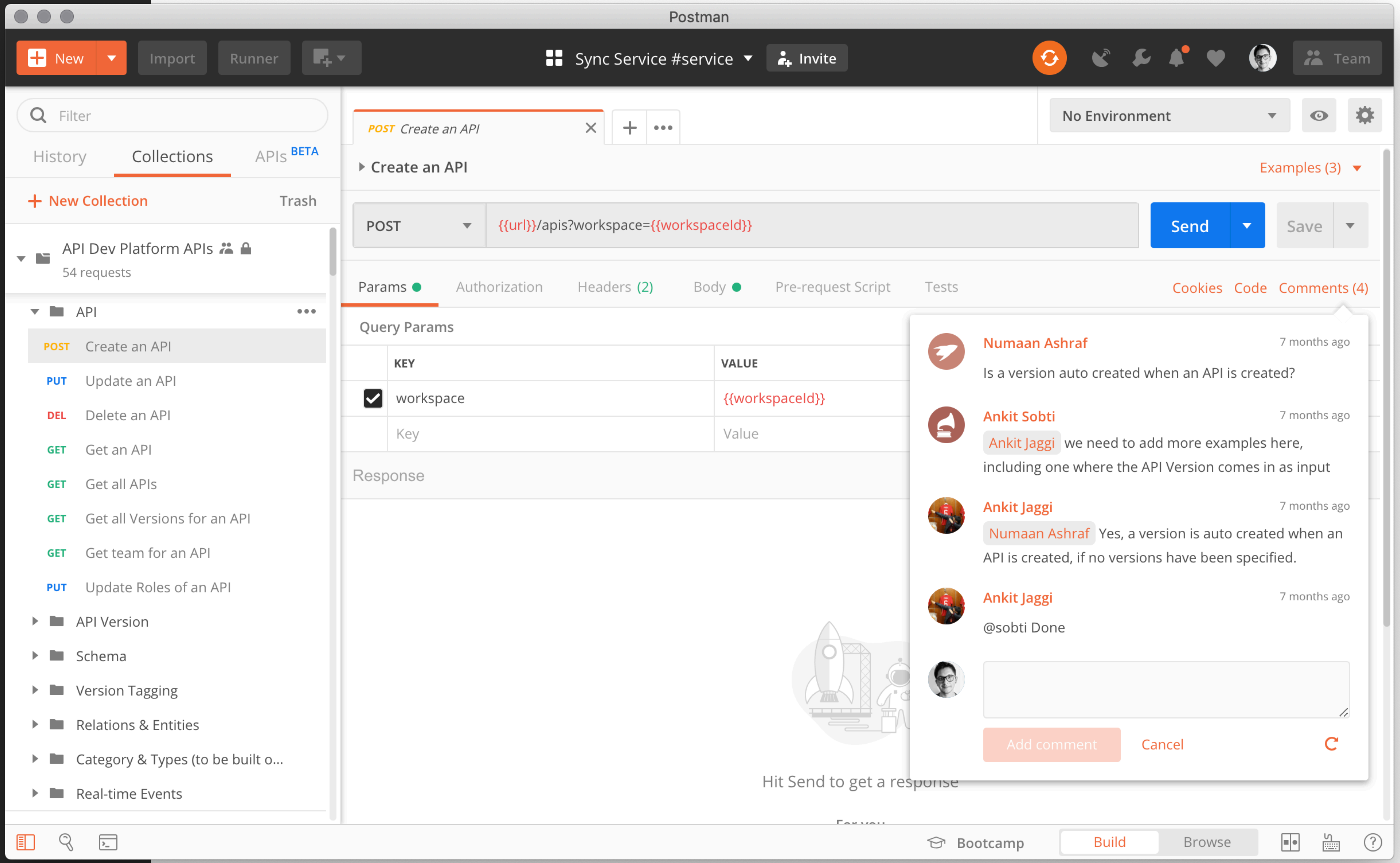
https://948bdb3e-5be5-49ad-b128-35d350055c24.mock.pstmn.io/api/v1/logs

```
{
    "count": 5,
    "entries": [
        {
            "serviceName": "foo",
            "timestamp": 1540206226229,
            "description": "Received foo request from user 100"
        },
        {
            "serviceName": "bar",
            "timestamp": 1540206226121,
            "description": "Sent email to user 99"
        },
        {
            "serviceName": "foo",
            "timestamp": 154020622502,
            "description": "Received foo request from user 10"
        },
        {
            "serviceName": "baz",
            "timestamp": 1540206223230,
            "description": "Activated user 101"
        },
        {
            "serviceName": "bar",
            "timestamp": 1540206222126,
            "description": "Error sending email to user 10"
        }
    ]
}
```

POSTMAN

Comments

- To make **contextual** comments and tag other team members in collections, in the app, and in the browser
- Used to **negotiate** API design among stakeholders



API Dev Platform APIs
54 requests

API
- POST Create an API
- PUT Update an API
- DEL Delete an API
- GET Get an API
- GET Get all APIs
- GET Get all Versions for an API
- GET Get team for an API
- PUT Update Roles of an API

API Version
Schema
Version Tagging
Relations & Entities
Category & Types (to be built o...
Real-time Events

POST Create an API

Create an API

POST {{url}}/apis?workspace={{workspaceId}}

Query Params

	KEY	VALUE
✓	workspace	{{workspaceId}}
	Key	Value

Numaan Ashraf 7 months ago
Is a version auto created when an API is created?

Ankit Sobti 7 months ago
Ankit Jaggi we need to add more examples here, including one where the API Version comes in as input

Ankit Jaggi 7 months ago
Numaan Ashraf Yes, a version is auto created when an API is created, if no versions have been specified.

Ankit Jaggi 7 months ago
@sobti Done

POSTMAN

Questions?

(and a quick break)

POSTMAN

Part 4: Role - API Producer

POSTMAN

Create blueprint collection

- Build a blueprint collection
  - Add requests
  - Add documentation
  - Add examples
    - Success cases
    - Failure cases
- Share collection in Workspace
- Comments on requests

POSTMAN

Create Mock Server

- What is a Mock Server? (Recap)
- Create a mock server from Blueprint collection
- Execute Blueprint collection against Mock

POSTMAN

Create “API” and schema

- Create a new “API”
- Add OpenAPI 3.0 YAML schema
  - From: bit.ly/postman-api-yaml
  - File: api-v0.1.yaml

POSTMAN

Use versioning

- Edit API version tag to v0.1
- Add blueprint collection and mock to API v0.1

POSTMAN

Questions?

(and another quick break)

POSTMAN

Part 5: Role - API Consumer

POSTMAN

Using Mocks as a consumer

- Build contract collection based on blueprint collection
- Send requests to Mock endpoint
- Save and document requests
- Add tests to assert on response
- Switch base URL using environments

POSTMAN

Part 6: Work with versions

POSTMAN

Create v0.2

- Create v0.2 for the API
- Update OpenAPI schema
  - From: bit.ly/postman-api-yaml
  - File: api-v0.2.yaml
- Update blueprint
- Update contract
- Tag collections with new version

POSTMAN

Questions?

(and we’re done!)

POSTMAN

Thank you!

@postmanclient
@kaustavdm
betterpractices.dev