



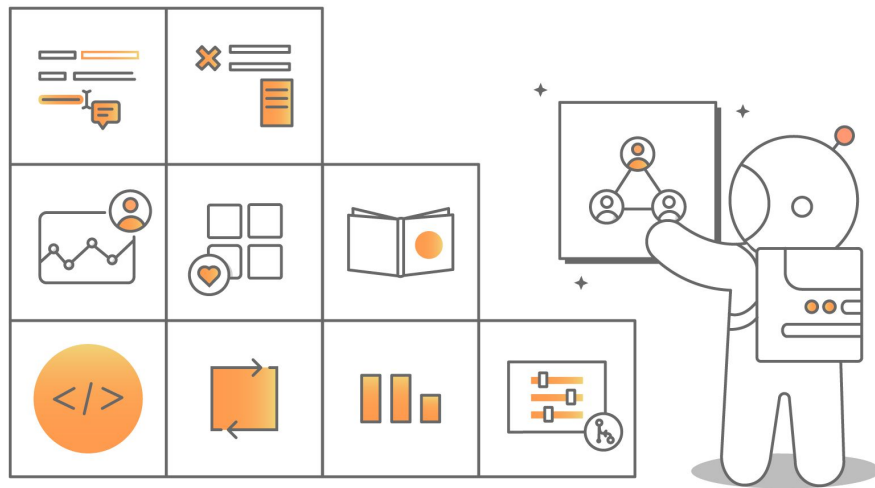
POSTMAN

# 10 Postman Features Everyone Should Know



**Arlemi Turpault**

Developer Advocacy Lead





POSTMAN





15 Answers

Active

Oldest

Votes

2075

I have been making a Chrome app called [Postman](#) for this type of stuff. All the other extensions seemed a bit dated so made my own. It also has a bunch of other features which have been helpful for documenting our own API here.



Postman now also has [native apps](#) (i.e. standalone) for Windows, Mac and Linux! It is more preferable now to use native apps, read more [here](#).



Share Improve this answer Follow

edited Jul 4 '17 at 6:39



Rakete1111

42.2k ● 11 ● 107 ● 141

answered Mar 15 '12 at 8:32



Abhinav

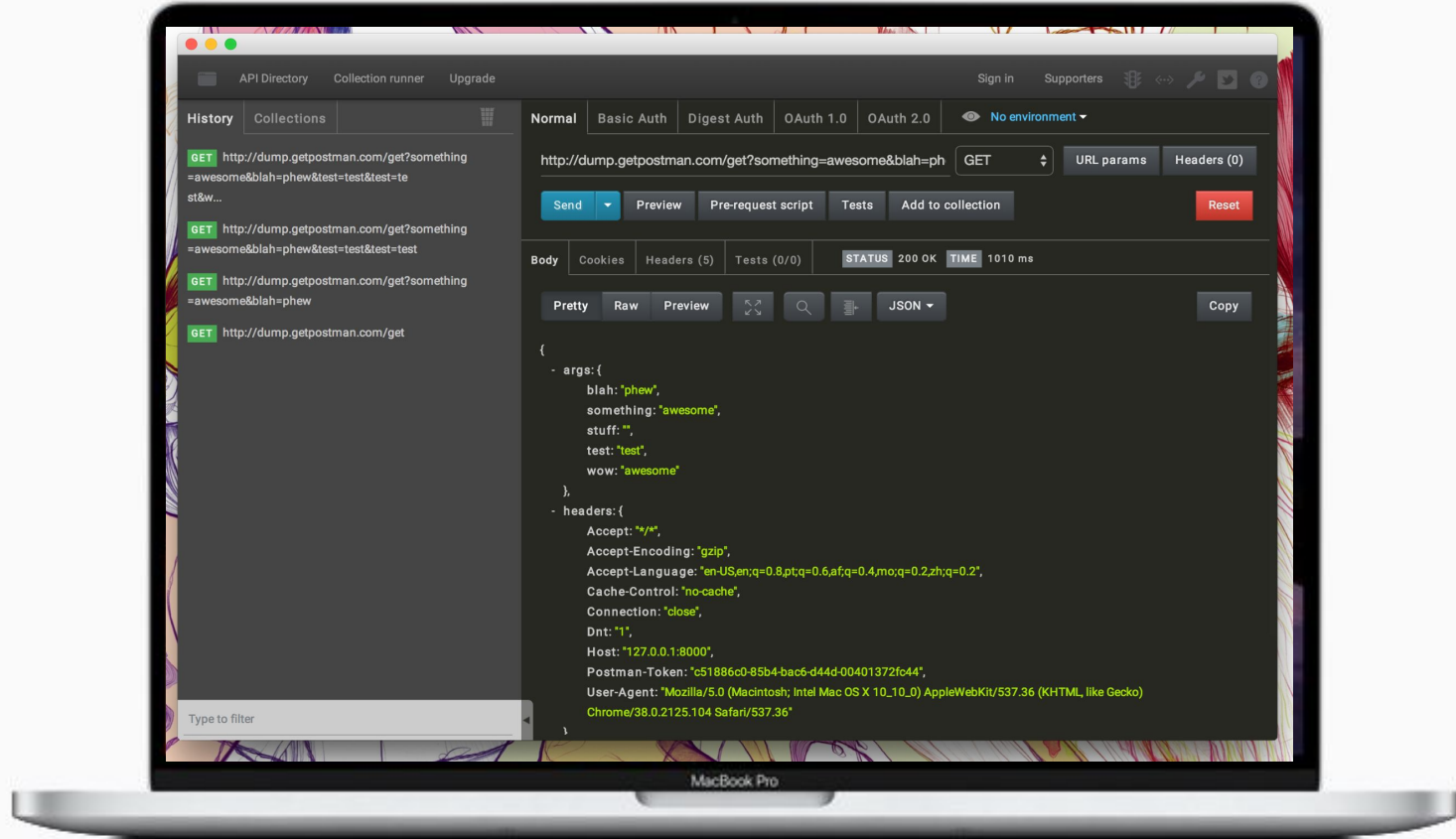
32.5k ● 9 ● 39 ● 48



@getpostman

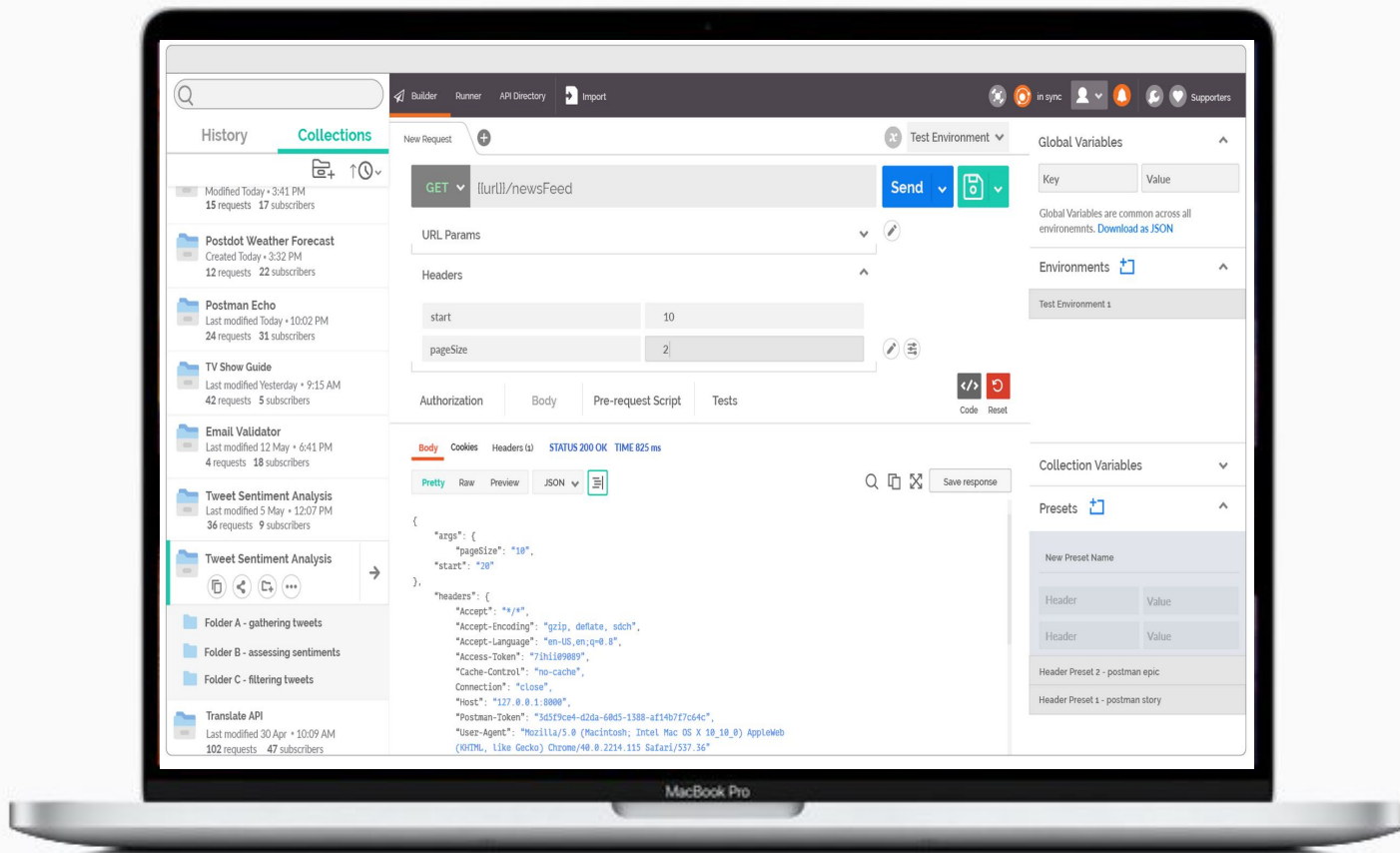


# API Client



@arlemi

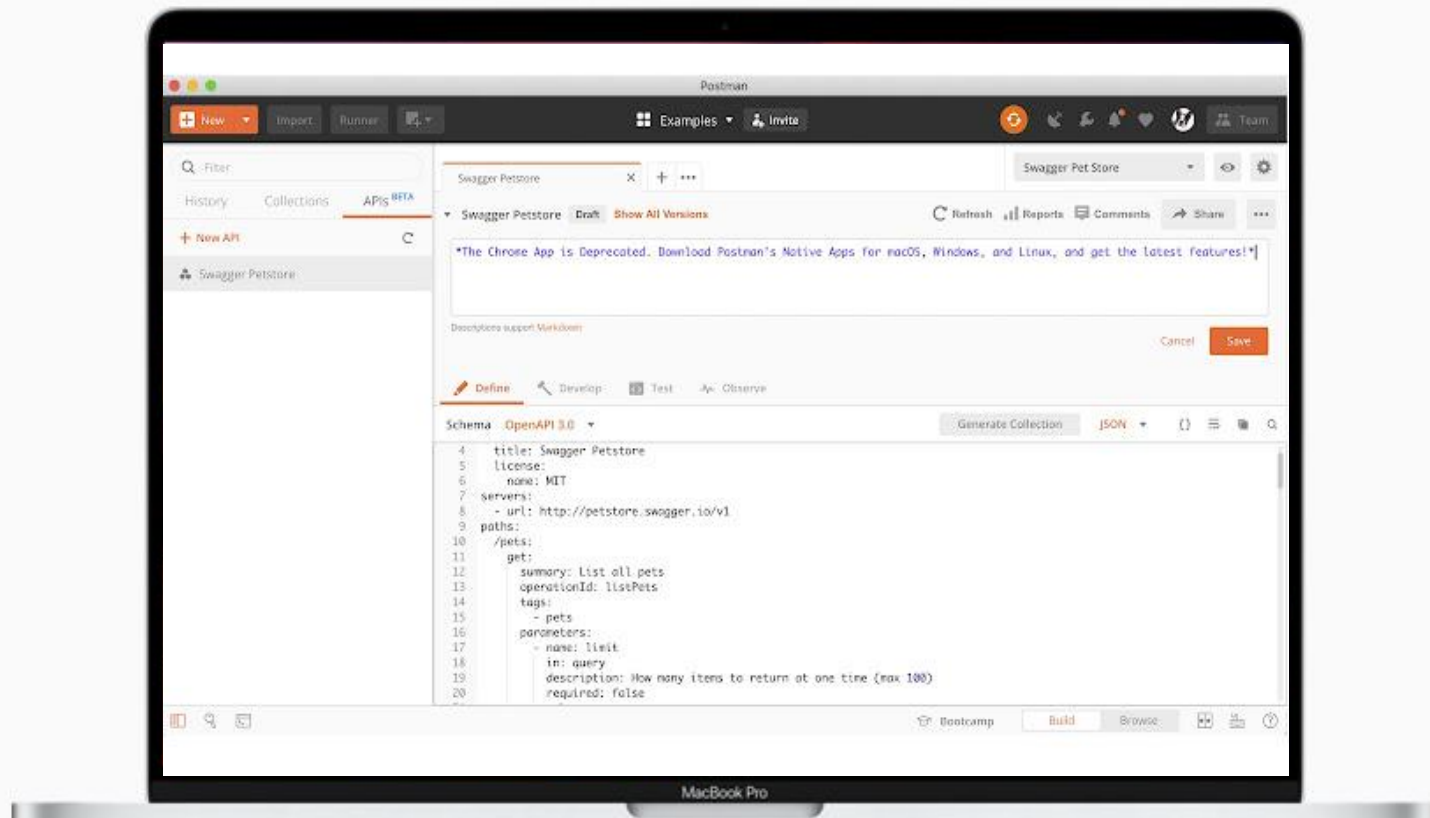
# API Client



@arlemi

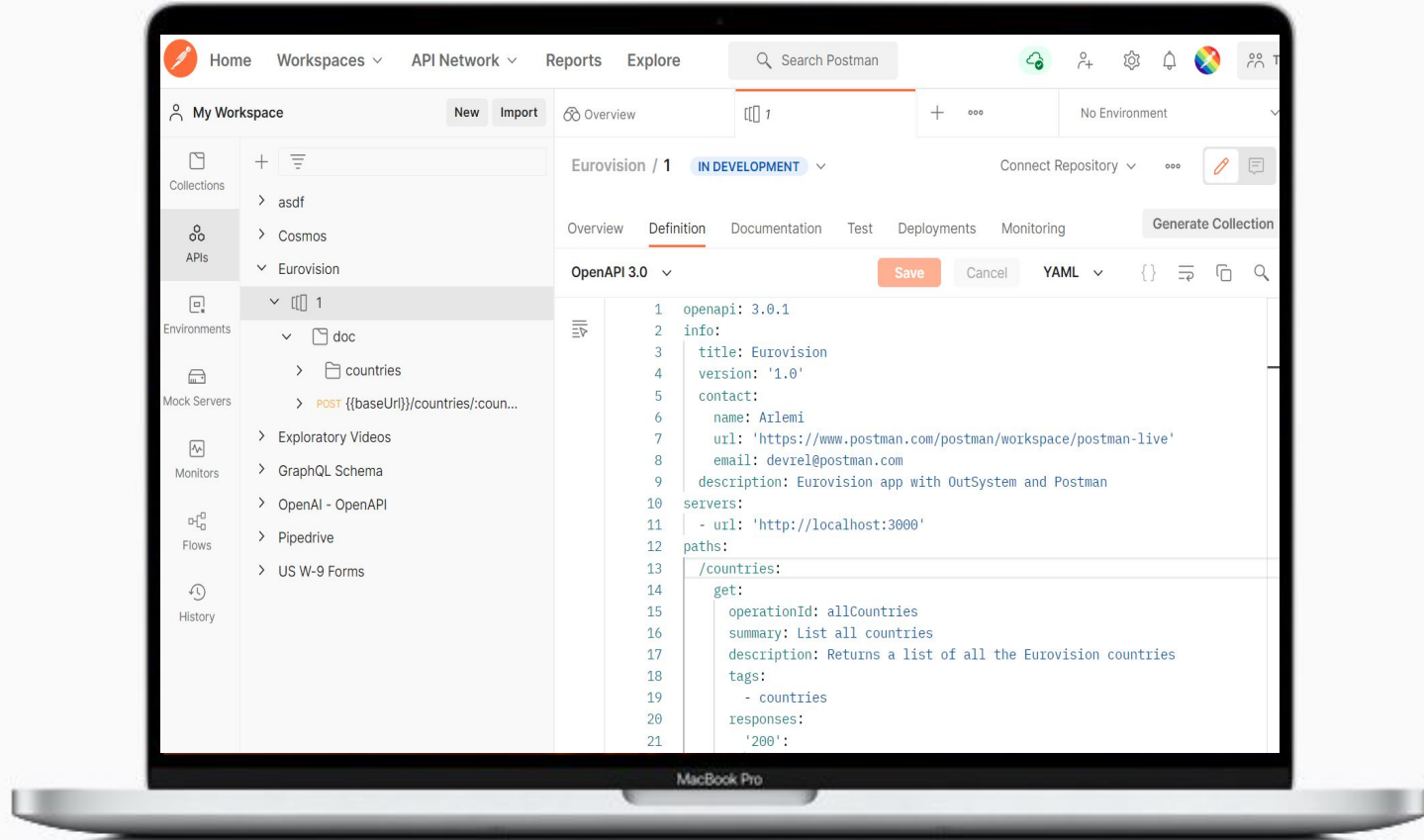


# API Collaboration Platform



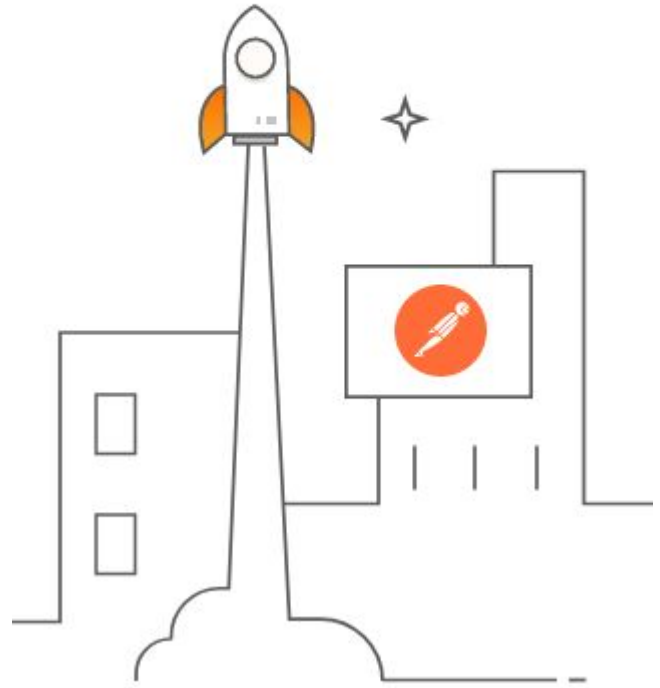


# API Platform



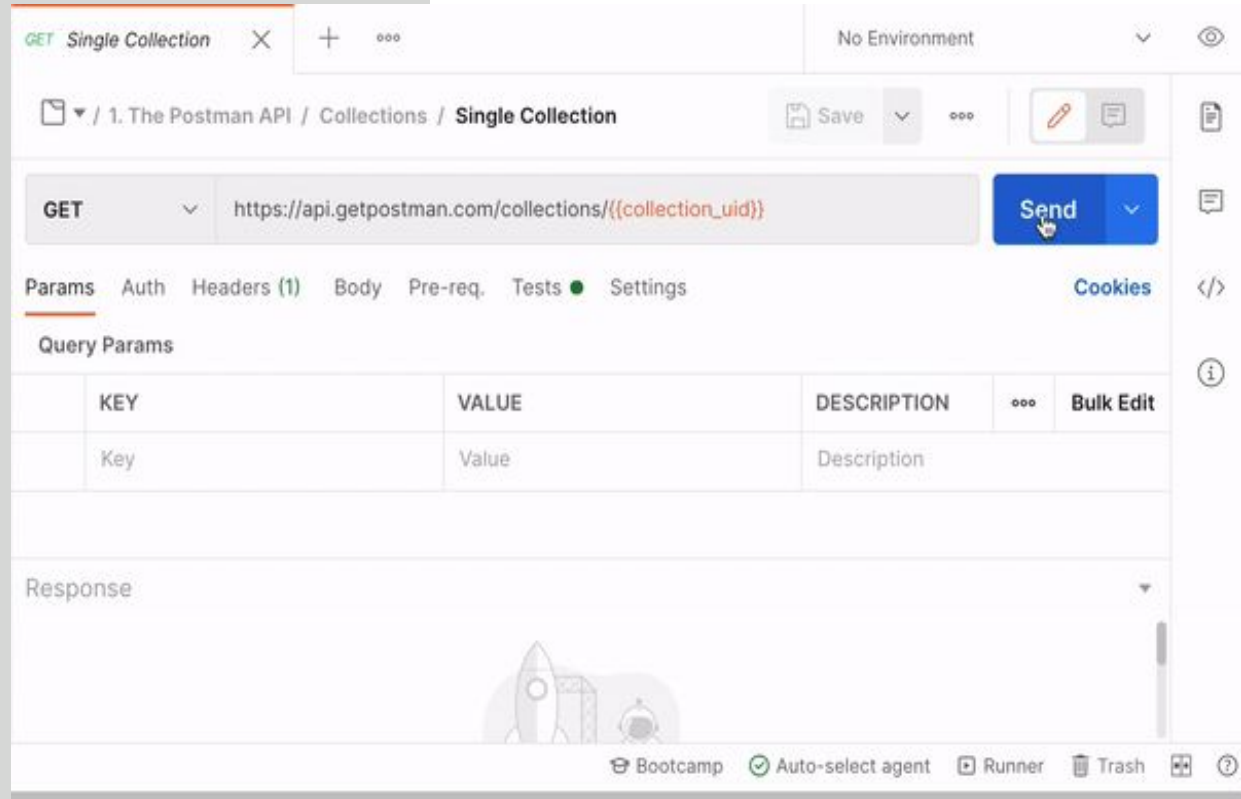
@arlemi

# 10 things about Postman Everyone Should Know



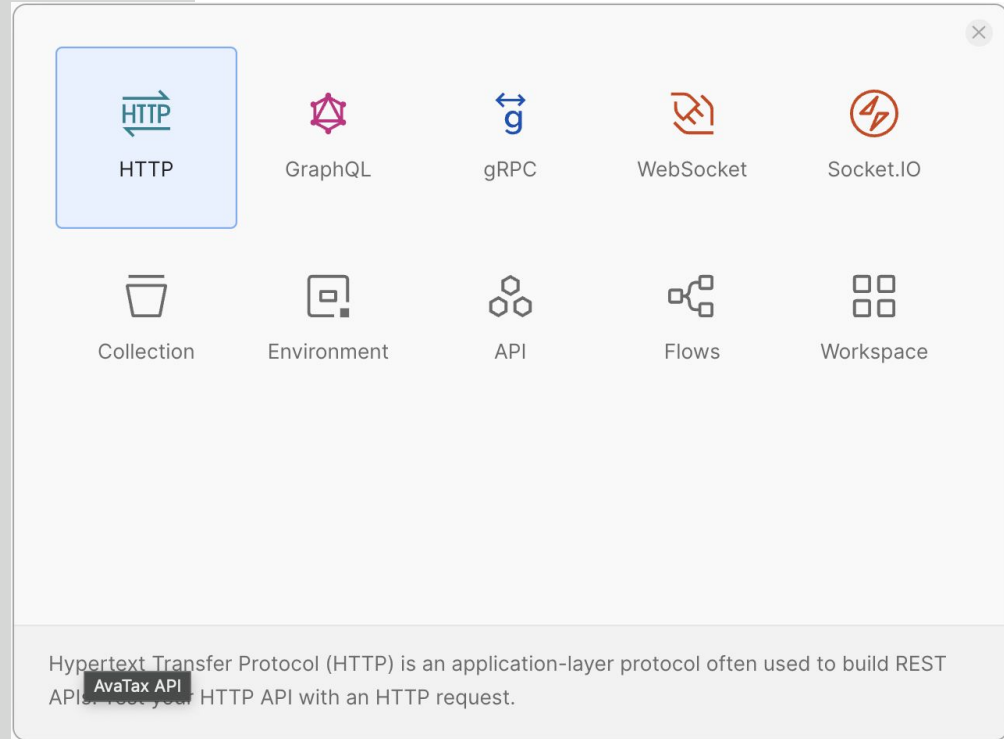


# The Postman API

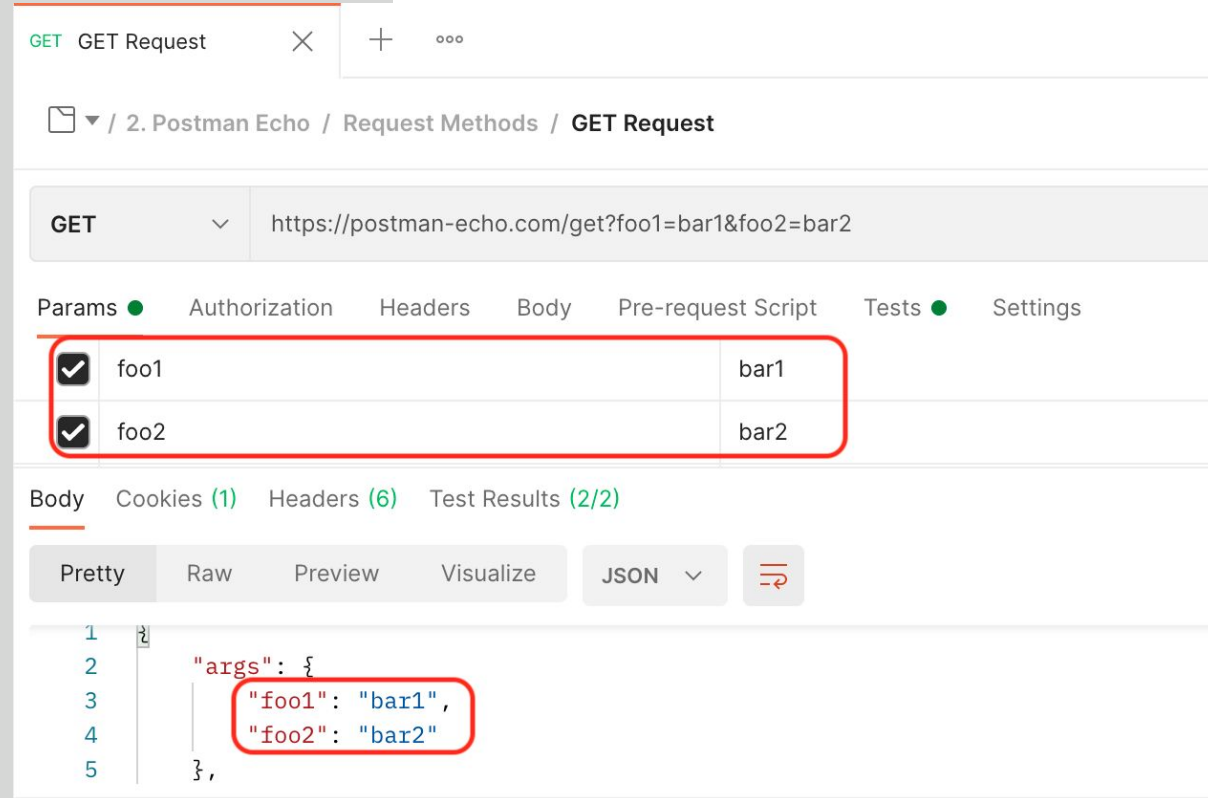


# Multi-Protocol

- gRPC
- WebSocket
- Socket.io
- GraphQL



# Postman Echo(s)



GET Request

2. Postman Echo / Request Methods / GET Request

GET <https://postman-echo.com/get?foo1=bar1&foo2=bar2>

Params ● Authorization Headers Body Pre-request Script Tests ● Settings

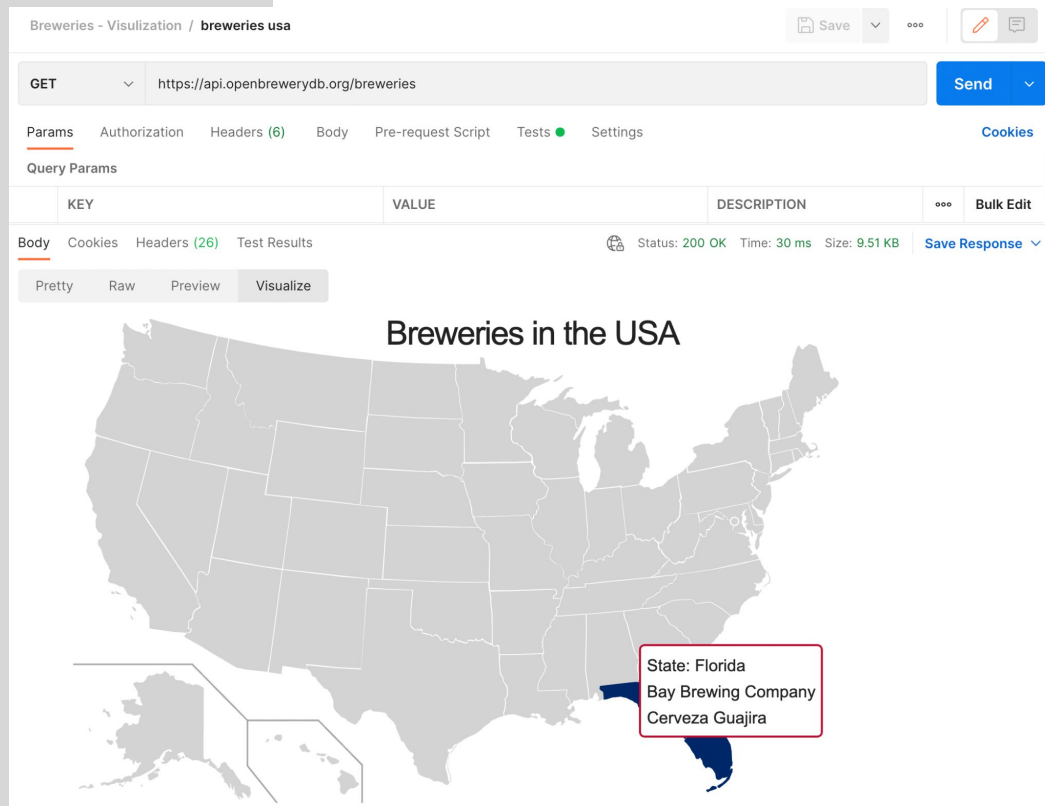
<input checked="" type="checkbox"/>	foo1	bar1
<input checked="" type="checkbox"/>	foo2	bar2

Body Cookies (1) Headers (6) Test Results (2/2)

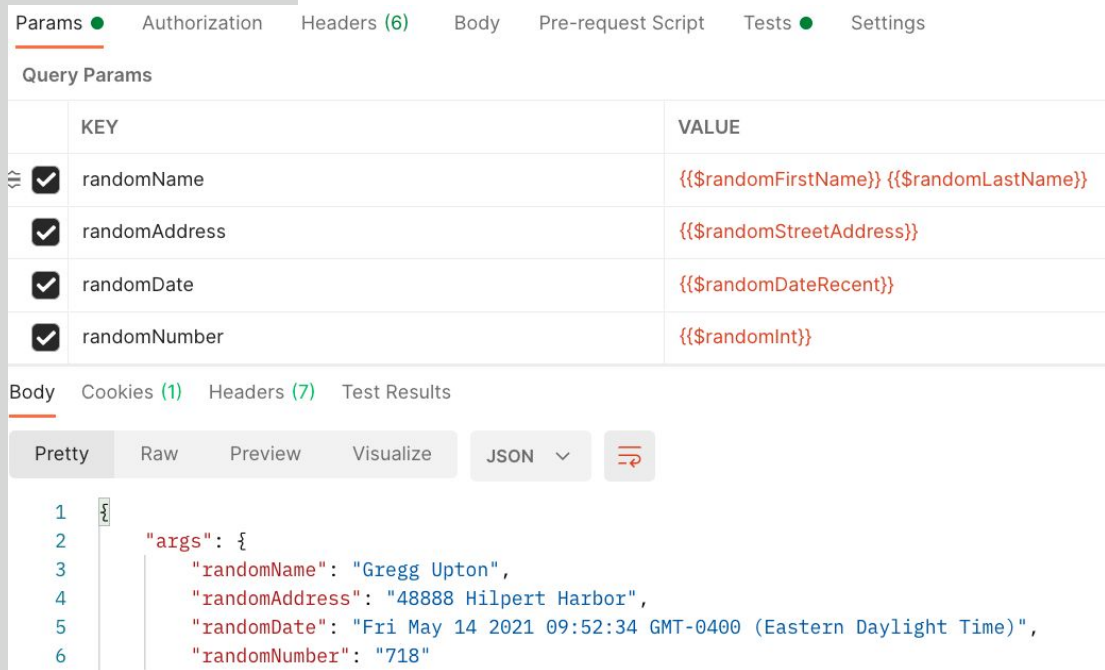
Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "args": {
3     "foo1": "bar1",
4     "foo2": "bar2"
5   },
6 }
```

# Postman Visualizers



# Built-in Libraries




The image shows a screenshot of the Postman REST client interface. The top navigation bar includes tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. The 'Params' tab is selected, displaying a table of query parameters. Below the table, the 'Body' tab is selected, showing a JSON response in the 'Pretty' view.

**Query Params**

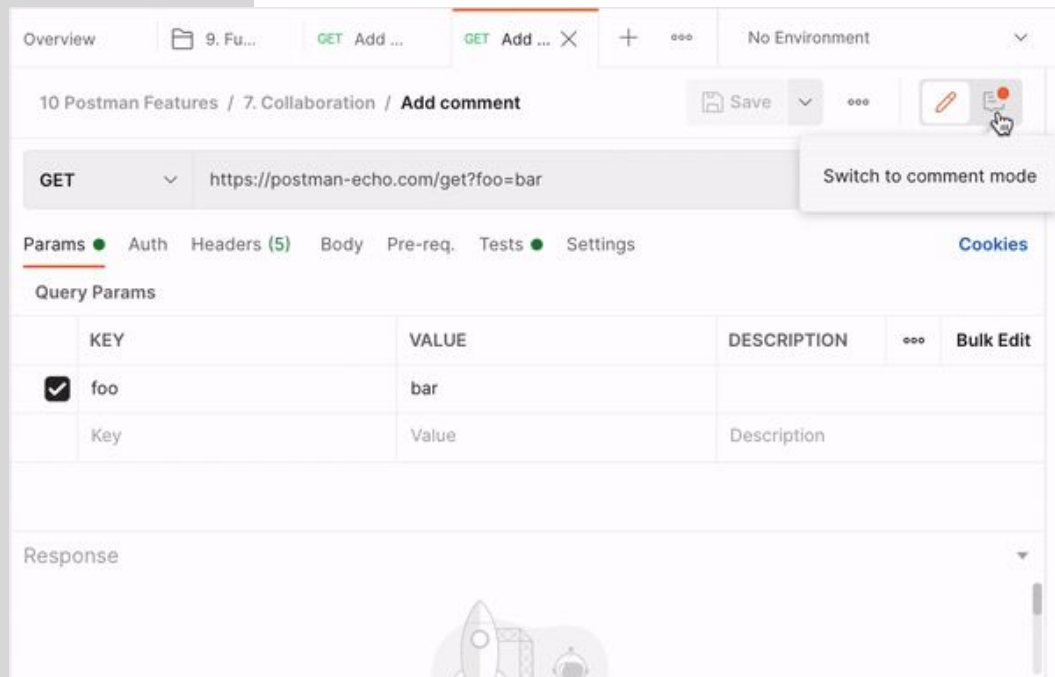
	KEY	VALUE
<input checked="" type="checkbox"/>	randomName	{{ \$randomFirstName }} {{ \$randomLastName }}
<input checked="" type="checkbox"/>	randomAddress	{{ \$randomStreetAddress }}
<input checked="" type="checkbox"/>	randomDate	{{ \$randomDateRecent }}
<input checked="" type="checkbox"/>	randomNumber	{{ \$randomInt }}

**Body** Cookies (1) Headers (7) Test Results

Pretty Raw Preview Visualize JSON 

```
1  {
2    "args": {
3      "randomName": "Gregg Upton",
4      "randomAddress": "48888 Hilpert Harbor",
5      "randomDate": "Fri May 14 2021 09:52:34 GMT-0400 (Eastern Daylight Time)",
6      "randomNumber": "718"
```

# Collaboration & Commenting



The image shows a screenshot of the Postman application interface. At the top, there's a navigation bar with tabs for 'Overview', '9. Fu...', 'GET Add ...', and 'GET Add ... X'. To the right, it says 'No Environment'. Below this, a breadcrumb trail reads '10 Postman Features / 7. Collaboration / Add comment'. A 'Save' button and a menu icon are also visible. The main area shows a REST client request with the method 'GET' and the URL 'https://postman-echo.com/get?foo=bar'. A dropdown menu is open, showing 'Switch to comment mode'. Below the URL bar, there are tabs for 'Params', 'Auth', 'Headers (5)', 'Body', 'Pre-req.', 'Tests', and 'Settings'. The 'Params' tab is selected, showing a table of query parameters. The table has columns for 'KEY', 'VALUE', and 'DESCRIPTION'. One parameter is listed: 'foo' with value 'bar'. Below the table, there's a 'Response' section. In the bottom right corner, there's a rocket ship icon.

Overview 9. Fu... GET Add ... GET Add ... X + ... No Environment

10 Postman Features / 7. Collaboration / Add comment Save ...

GET https://postman-echo.com/get?foo=bar Switch to comment mode

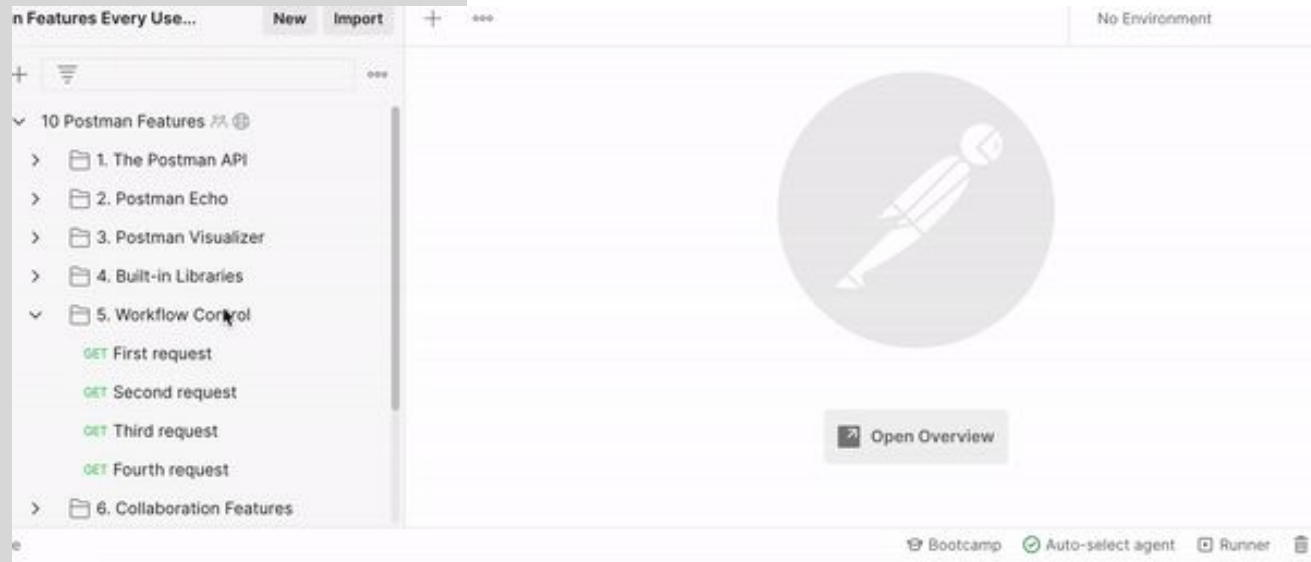
Params Auth Headers (5) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	foo	bar			
	Key	Value	Description		

Response

# Workflow Controls





# Folder-level Scripts & Variables

The screenshot displays the Postman interface with a folder named "7. Collection-Level Information" selected in the left sidebar. The main panel shows the "Tests" tab for this folder, containing two test scripts. The first test checks for a status code of 200, and the second test checks if the response body includes the string "Sean".

Left Sidebar (Folder List):

- > 2. Postman Echo
- > 3. Postman Visualizer
- > 4. Built-in Libraries
- > 5. Workflow Control
- > 6. Collaboration Features
- ✓ 7. Collection-Level Information
  - GET Get echo name 1
  - GET Get echo name 2
  - GET Get echo name 3
  - GET Get echo name 4
- > 8. Monitors for Regression Te...

Top Bar: 7. Collection-Level... | New | Import | No Environment

10 Postman Features / 7. Collection-Level Information | Run | Save

Authorization | Pre-request Script | **Tests**

These tests will execute after every request in this folder. [Learn more about Postman's execution order.](#)

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Body matches string", function () {
6   pm.expect(pm.response.text()).to.include("Sean");
7 });
```



# Test Automation



Plant Watering Monitor **HEALTHY**

Run Refresh

View: Run summary Individual requests

Filter By:

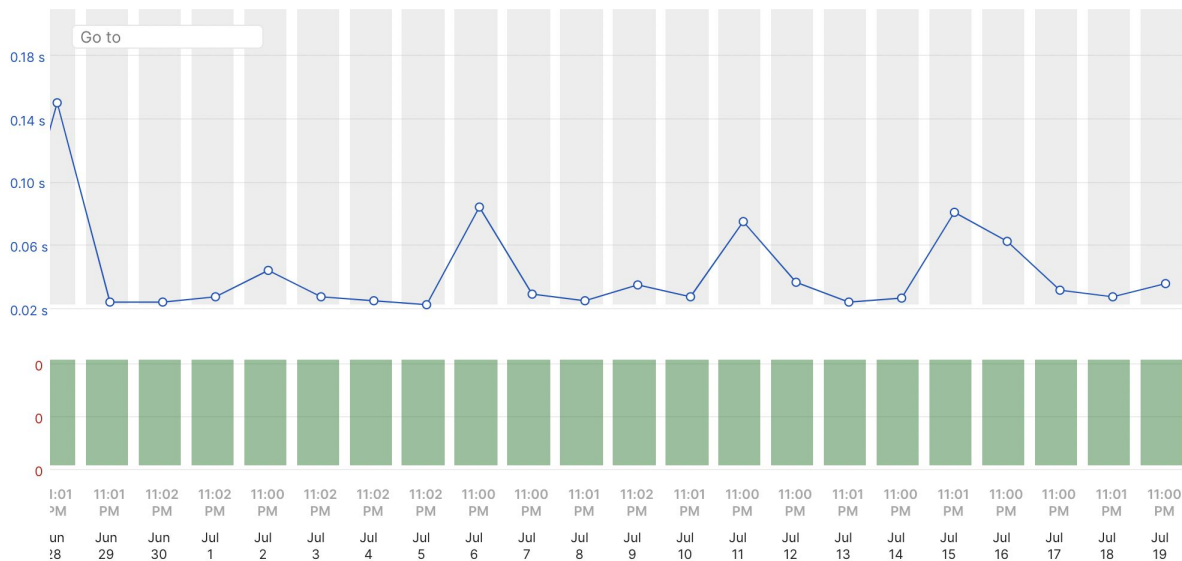
All Requests

Type: All

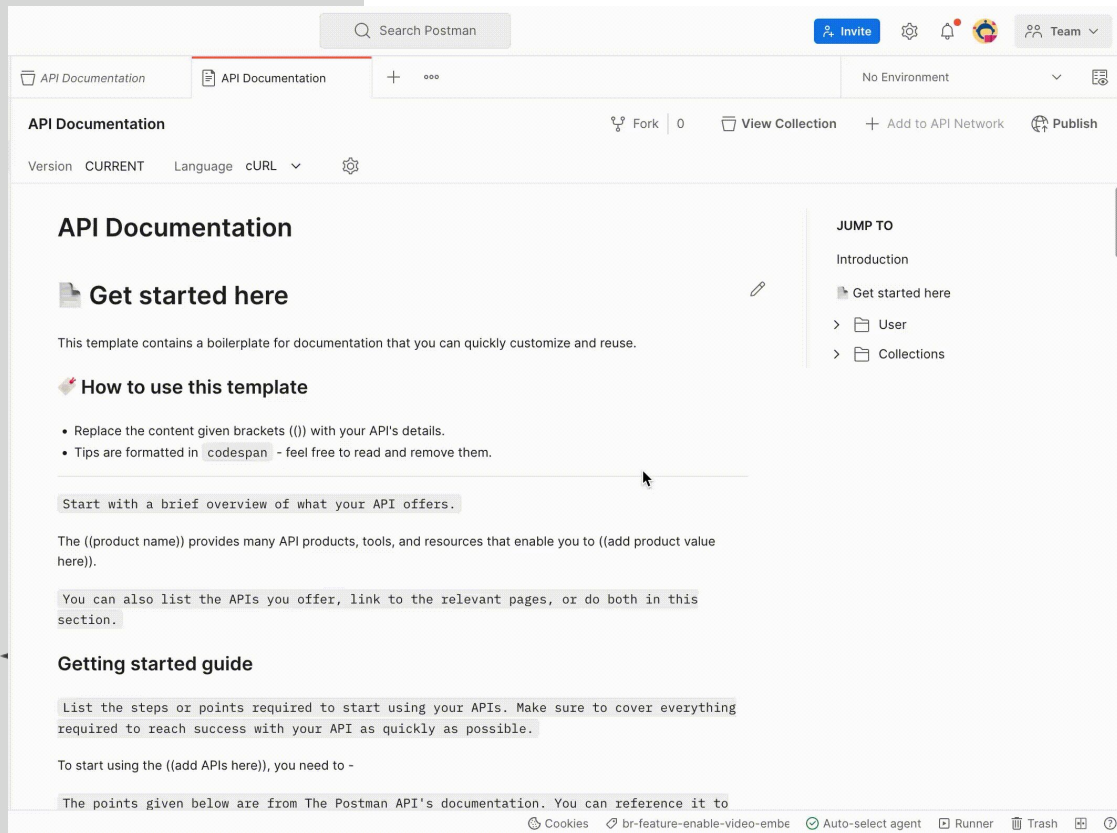
Run result: All

US (East)

Clear Filters



# Documentation



The screenshot shows the Postman API Documentation editor. At the top, there's a search bar labeled "Search Postman" and a blue "Invite" button. Below this, a tab labeled "API Documentation" is active. The main content area is titled "API Documentation" and includes a "Fork" button with a count of 0, a "View Collection" button, an "Add to API Network" button, and a "Publish" button. The content is organized into sections: "Get started here" (with a document icon and edit icon), "How to use this template" (with a red flag icon), and "Getting started guide". The "Get started here" section contains a paragraph: "This template contains a boilerplate for documentation that you can quickly customize and reuse." The "How to use this template" section contains two bullet points: "Replace the content given brackets (()) with your API's details." and "Tips are formatted in `codespan` - feel free to read and remove them." The "Getting started guide" section contains two paragraphs: "Start with a brief overview of what your API offers." and "The ((product name)) provides many API products, tools, and resources that enable you to ((add product value here)).". A right sidebar titled "JUMP TO" contains links to "Introduction", "Get started here", "User", and "Collections". The bottom of the interface shows a status bar with icons for Cookies, br-feature-enable-video-embe, Auto-select agent, Runner, Trash, and a help icon.

Search Postman

Invite

API Documentation

API Documentation

No Environment

API Documentation

Fork 0 View Collection Add to API Network Publish

Version CURRENT Language cURL

## API Documentation

### Get started here

This template contains a boilerplate for documentation that you can quickly customize and reuse.

### How to use this template

- Replace the content given brackets (()) with your API's details.
- Tips are formatted in `codespan` - feel free to read and remove them.

Start with a brief overview of what your API offers.

The ((product name)) provides many API products, tools, and resources that enable you to ((add product value here)).

You can also list the APIs you offer, link to the relevant pages, or do both in this section.

### Getting started guide

List the steps or points required to start using your APIs. Make sure to cover everything required to reach success with your API as quickly as possible.

To start using the ((add APIs here)), you need to -

The points given below are from The Postman API's documentation. You can reference it to

JUMP TO

- Introduction
- Get started here
- User
- Collections

Cookies br-feature-enable-video-embe Auto-select agent Runner Trash

# Bonus Feature: Flows

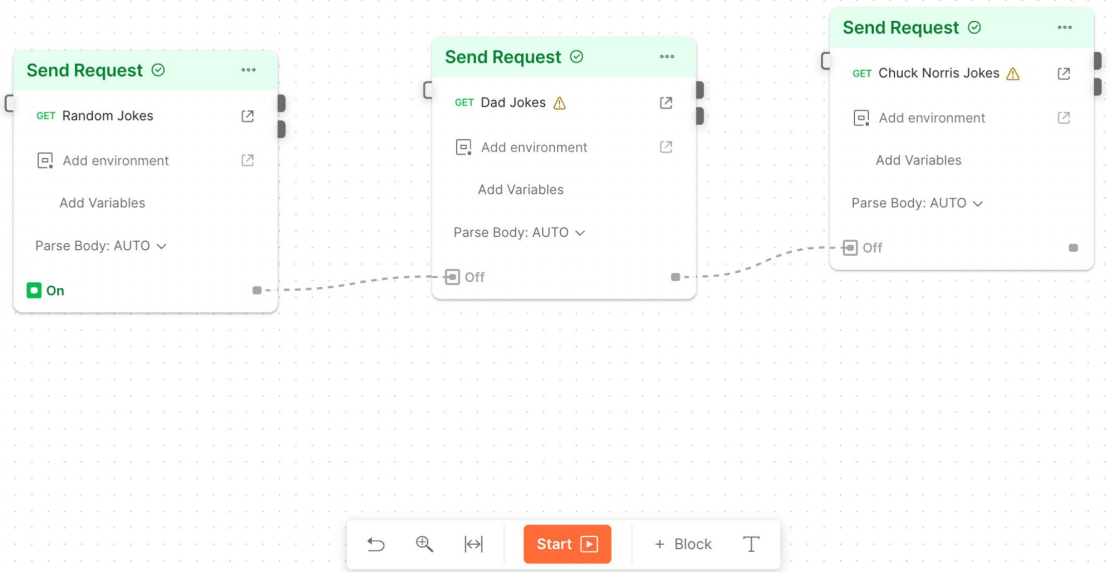
## 2 - Chaining requests chaining requests

 Fork | 0  Feedback  Docs

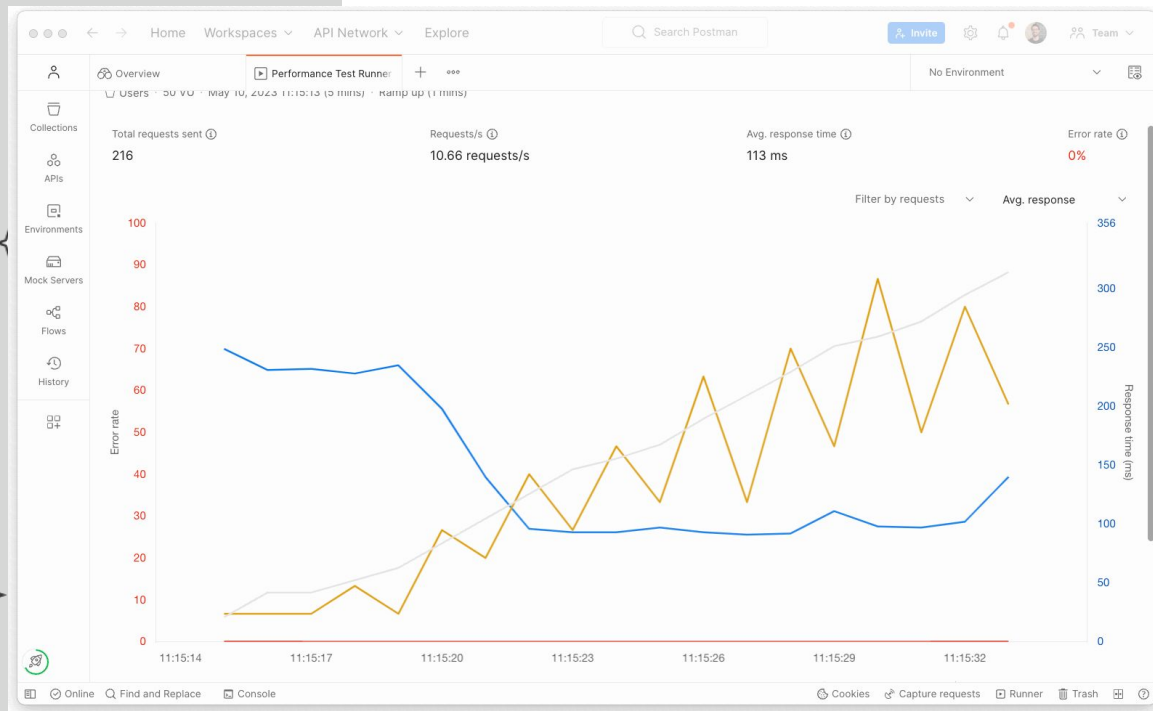
### Chaining Requests

Requests can be chained to run sequentially by linking blocks together. This is done by clicking and holding the bottom right grey icon to the bottom left "on" icon in the next block.

When everything is chained together, clicking start will run the blocks sequentially.



# Bonus Feature: Performance Testing



# Additional Resources



## Postman Webinars

*[postman.com/events/intergalactic](https://postman.com/events/intergalactic)*



## Postman Workshops

*[quickstarts.postman.com](https://quickstarts.postman.com)*



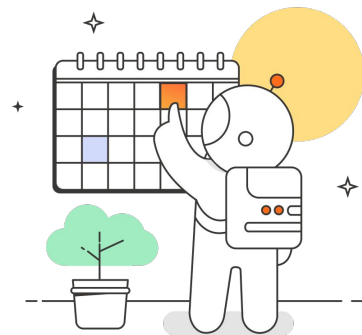
## Community Forum

*[community.postman.com](https://community.postman.com)*



## Postman Learning Center

*[learning.postman.com](https://learning.postman.com)*



@arlemi



# Thank You

